



New features to turbocharge pipeline development



Bob Welshmer
Senior Sales Engineer
Prophecy

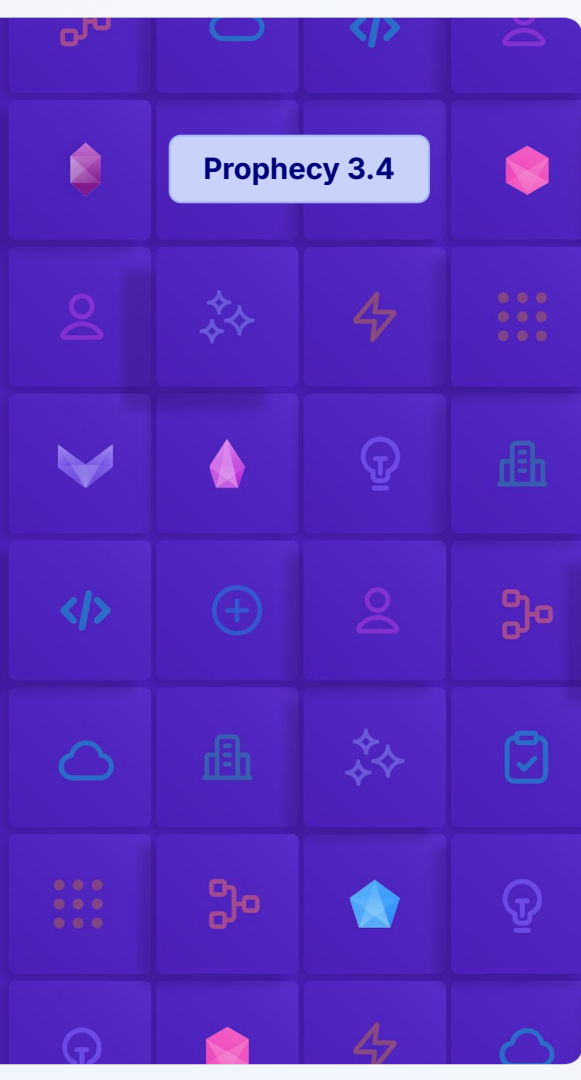


Anya Bida
Technical Evangelist
Prophecy



Kuldeep Singh
AI Architect
Prophecy

Prophecy 3.4



**Productivity wins!
Swag wins!**





**Welcome
Data Engineers,
Data Analysts,
Data Leaders!**





New features to turbocharge pipeline development



Bob Welshmer
Senior Sales Engineer
Prophecy

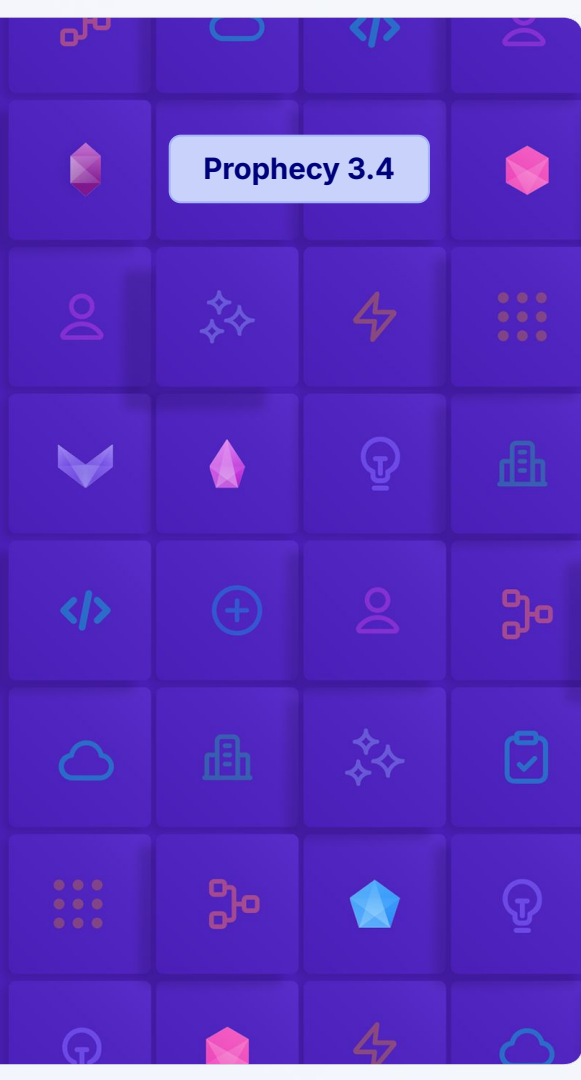


Anya Bida
Technical Evangelist
Prophecy



Kuldeep Singh
AI Architect
Prophecy

Prophecy 3.4



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



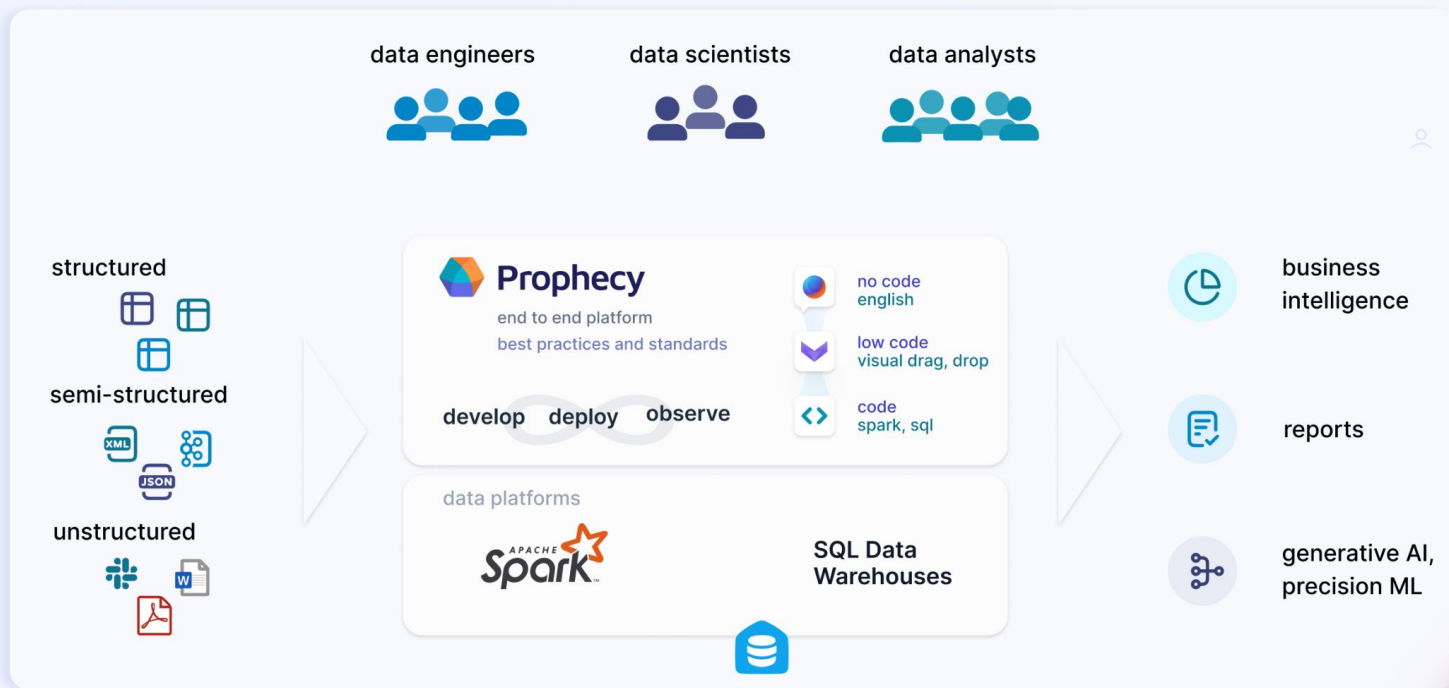
Generative AI
Precision ML



Reports



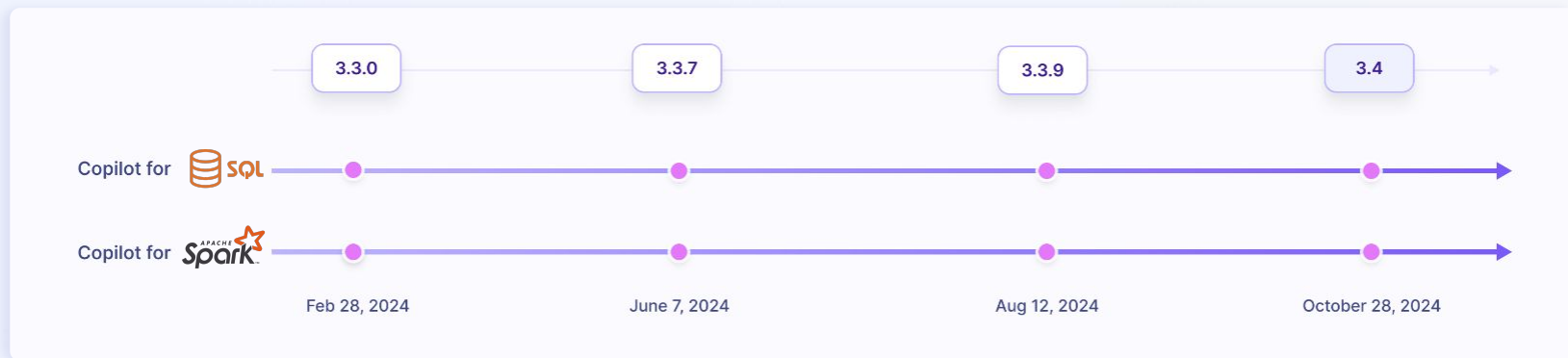
Prophecy Data Transformation Copilot





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

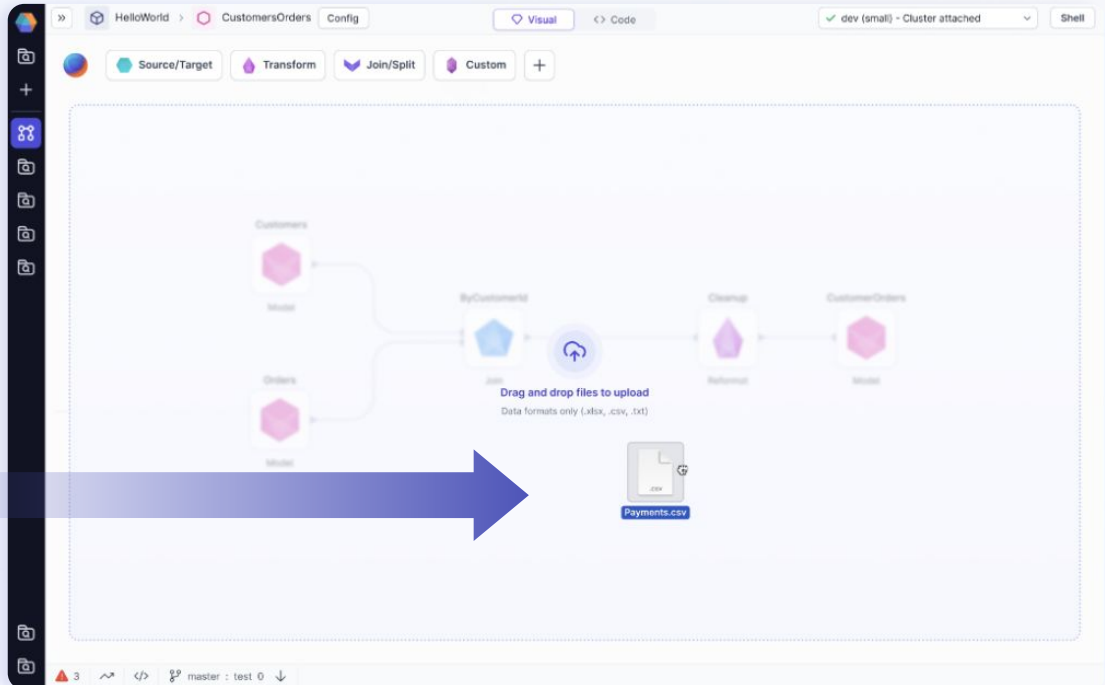
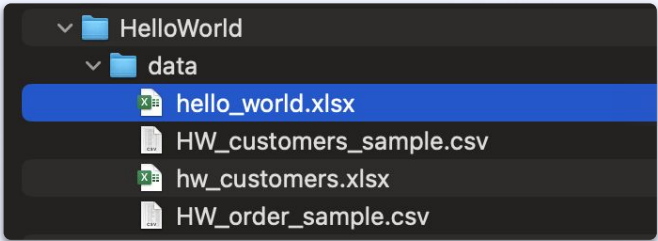
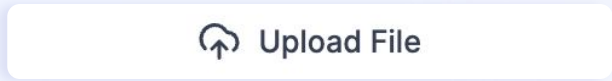
Problem

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

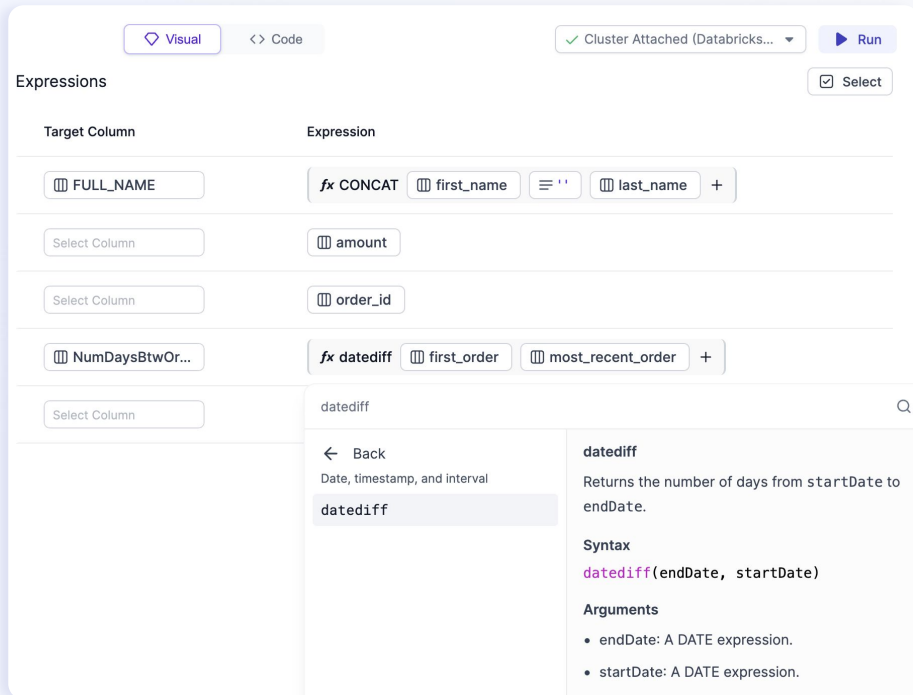
SQL or Python skills are usually required to create complex business expressions

👉 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 displays the Visual Expression Builder interface. At the top, there are tabs for 'Visual' and 'Code', and a 'Run' button. Below this is a table of expressions with columns for 'Target Column' and 'Expression'. The table contains several rows, including one for 'FULL_NAME' which uses the 'CONCAT' function to combine 'first_name' and 'last_name'. A 'Select' checkbox is visible in the top right corner of the table.

Below the table, a detailed view of the 'datediff' function is shown. It includes a 'Back' button, a description of the function as 'Date, timestamp, and interval', and a search bar. The function name 'datediff' is highlighted in the search results. To the right, the function's description is provided: 'Returns the number of days from startDate to endDate.' The syntax is shown as `datediff(endDate, startDate)`. The arguments section lists: 'endDate: A DATE expression.' and 'startDate: A DATE expression.'

Enhanced in 3.4

Synthetic Data Generator

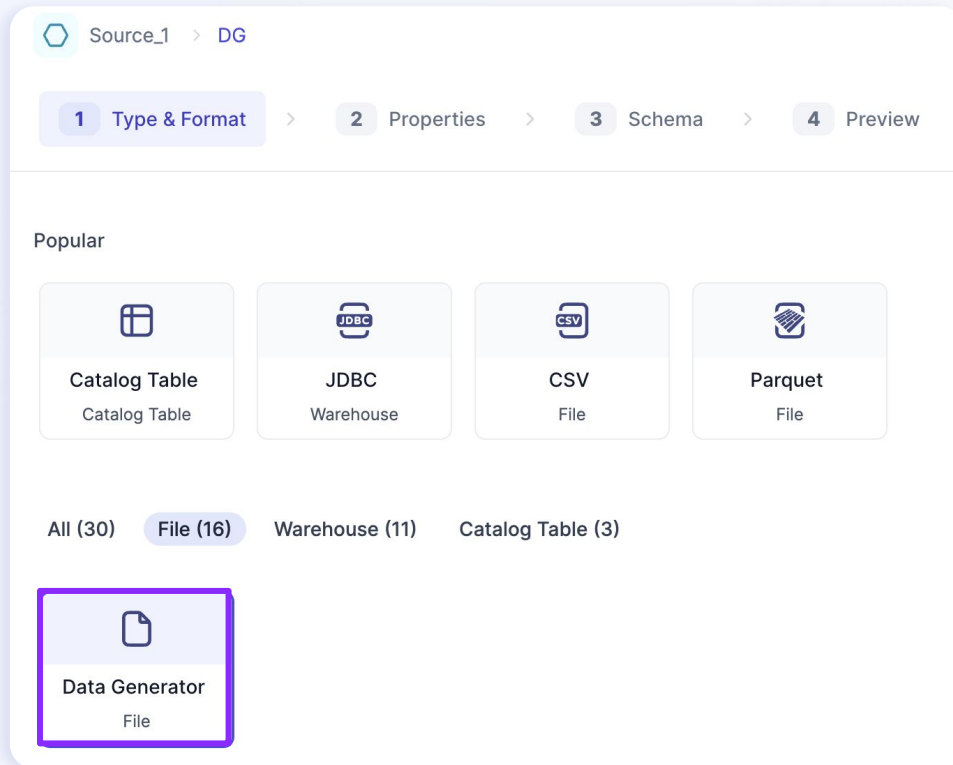
Problem

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



New in 3.4

Cleanup Messy Data

Problem

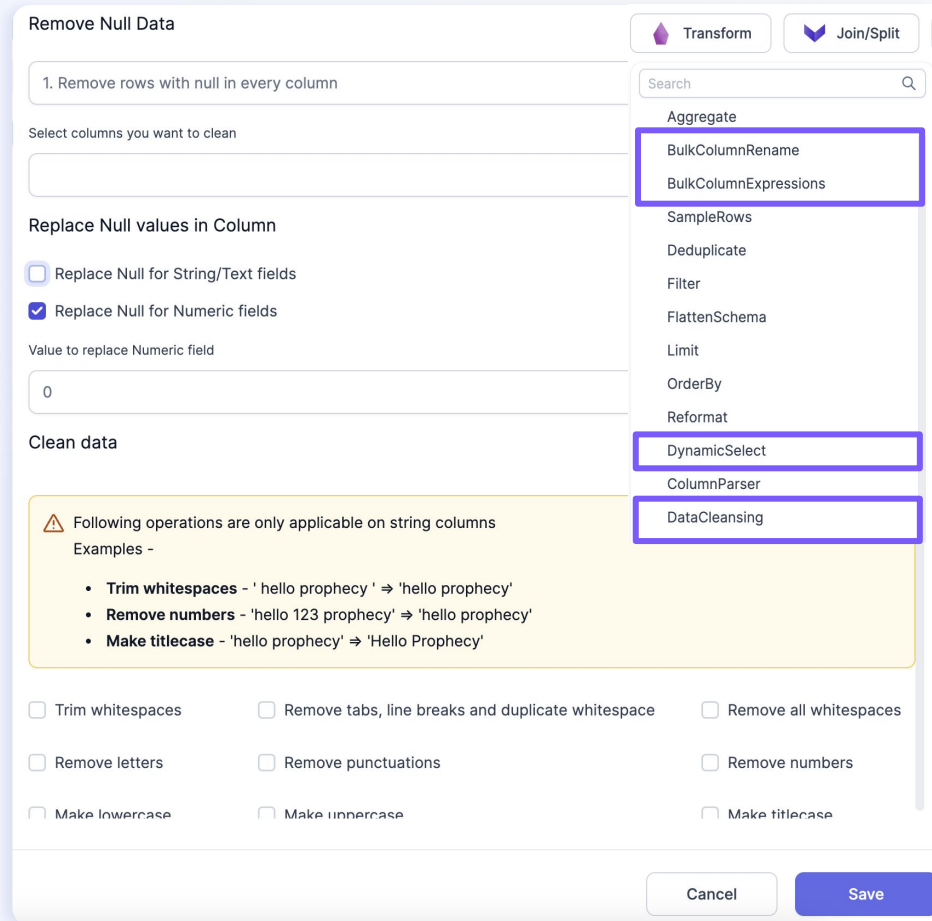
Got dozens(or hundreds!) of columns to tidy?
Managing these 1 by 1 took lots of time
consuming steps.

👉 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



Remove Null Data

1. Remove rows with null in every column

Select columns you want to clean

Replace Null values in Column

Replace Null for String/Text fields

Replace Null for Numeric fields

Value to replace Numeric field

0

Clean data

Following operations are only applicable on string columns

Examples -

- **Trim whitespaces** - ' hello prophecy ' ⇒ 'hello prophecy'
- **Remove numbers** - 'hello 123 prophecy' ⇒ 'hello prophecy'
- **Make titlecase** - 'hello prophecy' ⇒ 'Hello Prophecy'

Trim whitespaces

Remove tabs, line breaks and duplicate whitespace

Remove all whitespaces

Remove letters

Remove punctuations

Remove numbers

Make lowercase

Make uppercase

Make titlecase

Transform

Join/Split

Search

Aggregate

BulkColumnRename

BulkColumnExpressions

SampleRows

Deduplicate

Filter

FlattenSchema

Limit

OrderBy

Reformat

DynamicSelect

ColumnParser

DataCleansing

Cancel

Save

Data Delivery & Access Simplified

Problem

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

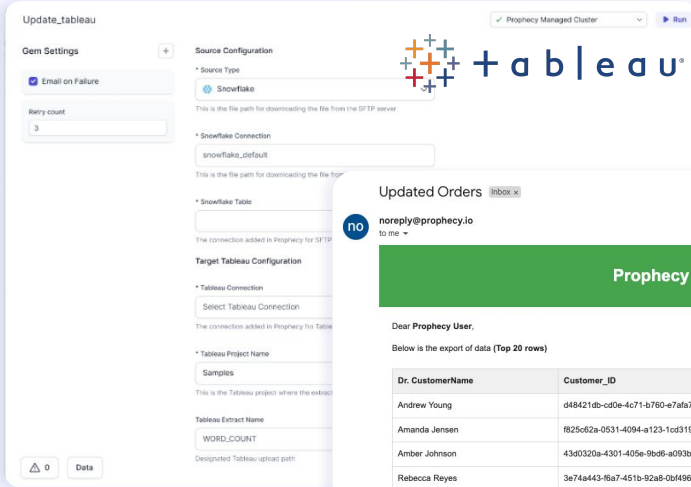
👉 Solution

New Source Gems

- Sharepoint
- SFTP

New Target Gems

- Tableau
- Email



The screenshot shows the Tableau configuration interface for a new source and target. The 'Source Configuration' section is set to 'Snowflake' with a file path for downloading files from the SFTP server. The 'Target Configuration' section is set to 'Email' with a project name 'Updated Orders' and a designated Tableau upload path.

The email export shows the following data:

Dr. CustomerName	Customer_ID	ShippingType
Andrew Young	d48421db-c00e-4c71-b760-e7afa7a36362	None
Amanda Jensen	f825c62a-0531-4094-a123-1cd3195604be	3
Amber Johnson	43d0320a-4301-4015e-9bd6-a093b56fa2f0	4
Rebecca Reyes	3e74a443-6ba7-451b-92a8-0b4496e09c69	2
Laura Watson	None	3
Kevin Simon	d8978a2a-466b-4a22-9b62-37525b6f5981	2
Laura Wright	e5cfaa65-8b84-4d77-8bc6-7330e3869bc9	3
Frederick Williams	5376ce67-7bd6-425f-830f-e55b02a5a819	2
Kathy Butler	2f840881-6225-4d31-8174-8784891fb3f7	4
Kevin Bums	38bbe949-692c-471c-b611-1d62d199e0f7	3

New in 3.4

NOTE: Admins to control who can send data

Sophisticated write options

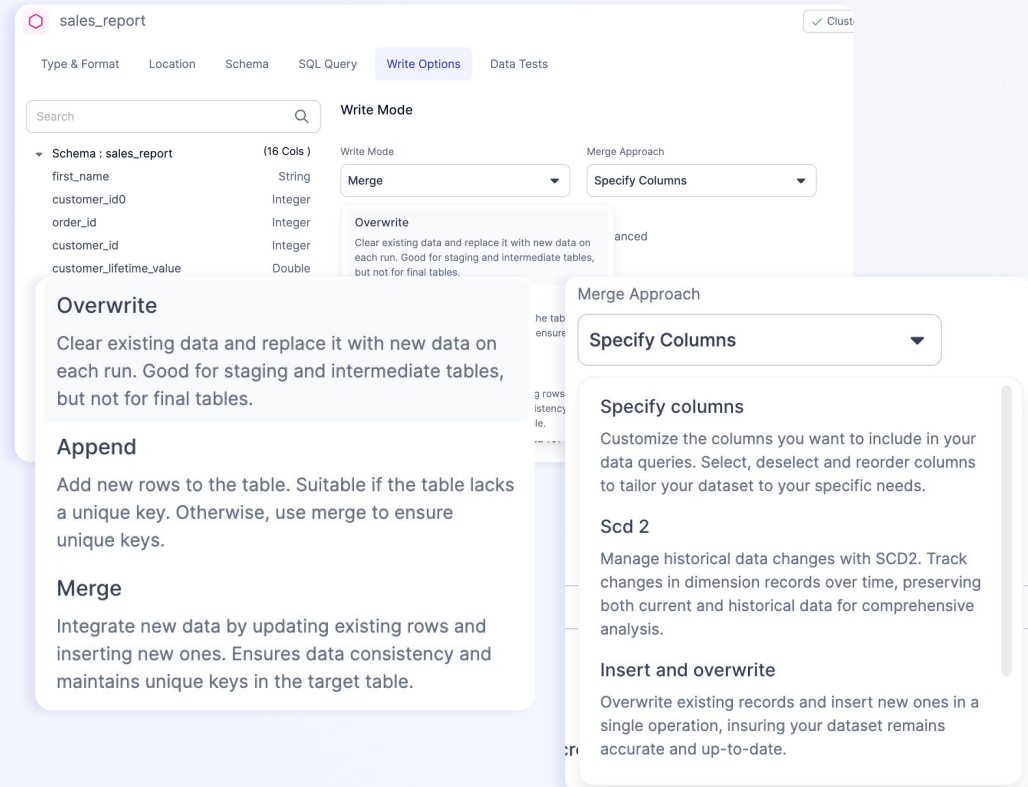
Problem

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 the 'Write Options' tab in the Databricks UI for a table named 'sales_report'. The interface includes a search bar, a table of column details, and configuration options for 'Write Mode' and 'Merge Approach'. The 'Write Mode' dropdown is set to 'Merge', and the 'Merge Approach' dropdown is set to 'Specify Columns'. Three callout boxes provide detailed descriptions for 'Overwrite', 'Append', and 'Merge' write modes, and another callout box provides details for the 'Specify Columns' merge approach.

Schema : sales_report	(16 Cols)
first_name	String
customer_id0	Integer
order_id	Integer
customer_id	Integer
customer_lifetime_value	Double

Write Mode

Write Mode: Merge

Merge Approach: Specify Columns

Overwrite
Clear existing data and replace it with new data on each run. Good for staging and intermediate tables, but not for final tables.

Append
Add new rows to the table. Suitable if the table lacks a unique key. Otherwise, use merge to ensure unique keys.

Merge
Integrate new data by updating existing rows and inserting new ones. Ensures data consistency and maintains unique keys in the target table.

Merge Approach: Specify Columns
Specify columns
Customize the columns you want to include in your data queries. Select, deselect and reorder columns to tailor your dataset to your specific needs.

Scd 2
Manage historical data changes with SCD2. Track changes in dimension records over time, preserving both current and historical data for comprehensive analysis.

Insert and overwrite
Overwrite existing records and insert new ones in a single operation, insuring your dataset remains accurate and up-to-date.

Enhanced in 3.4

Custom Gems

Problem

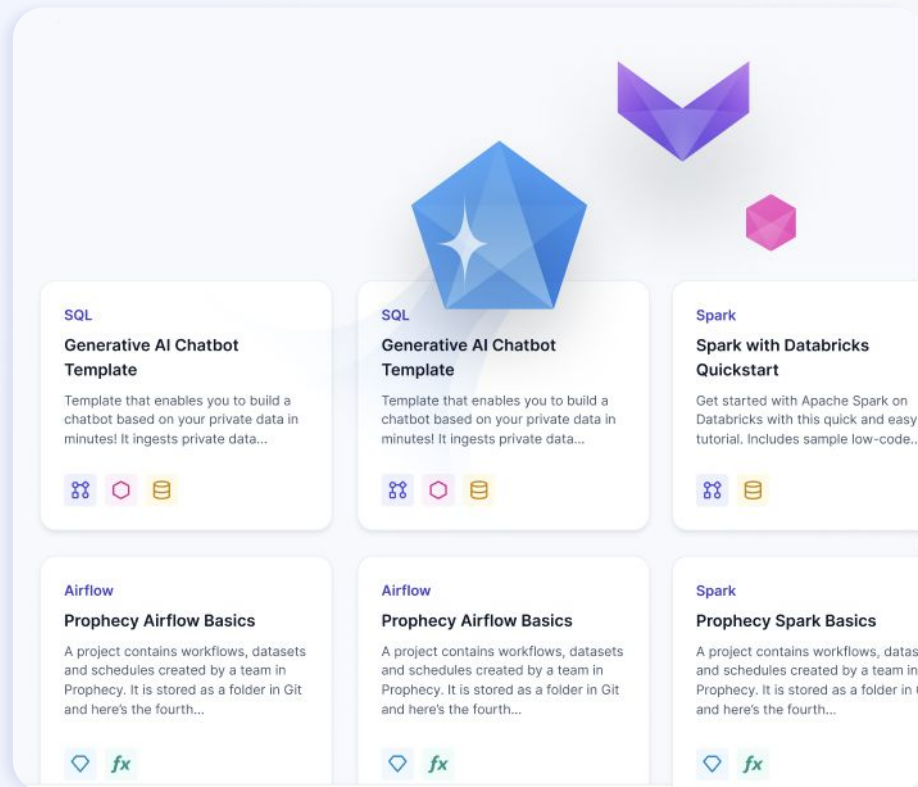
Every company has logic they'd like to standardize, version, and share. SQL users have been left out, until now.

👉 Solution

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

[doc](#)

Enhanced in 3.4





Pipeline observability



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



Lineage Run

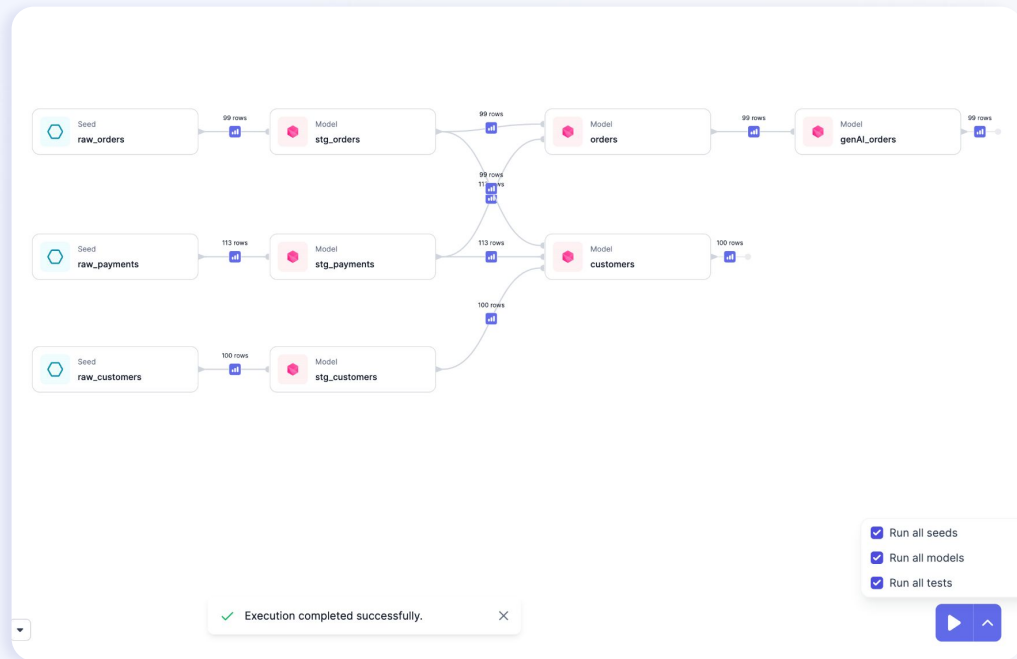
Problem

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](#)



Pipeline monitoring and debugging

Problem

Debugging pipelines is hard when you lack context.

Solution

We've enhanced the pipeline UI for better visibility.

See full run history & identify failures

Understand which gems have failed & why

Data Copilot: Automatically suggests fixes for commonly occurring issues

Enhanced in 3.4

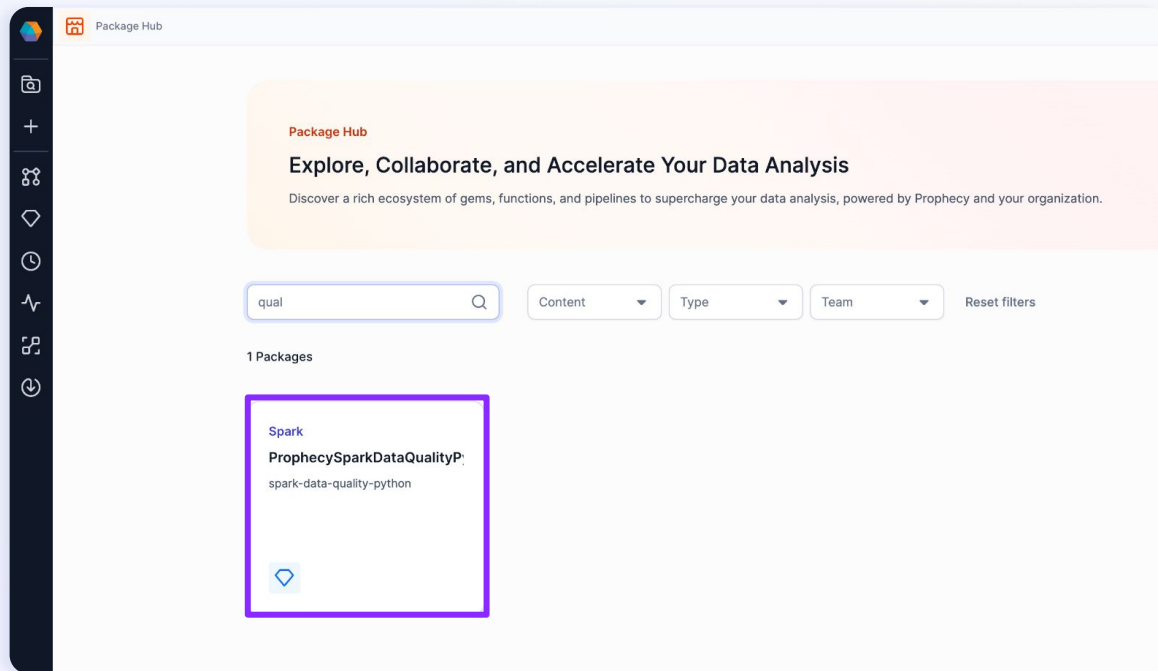
Data Quality Tests

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

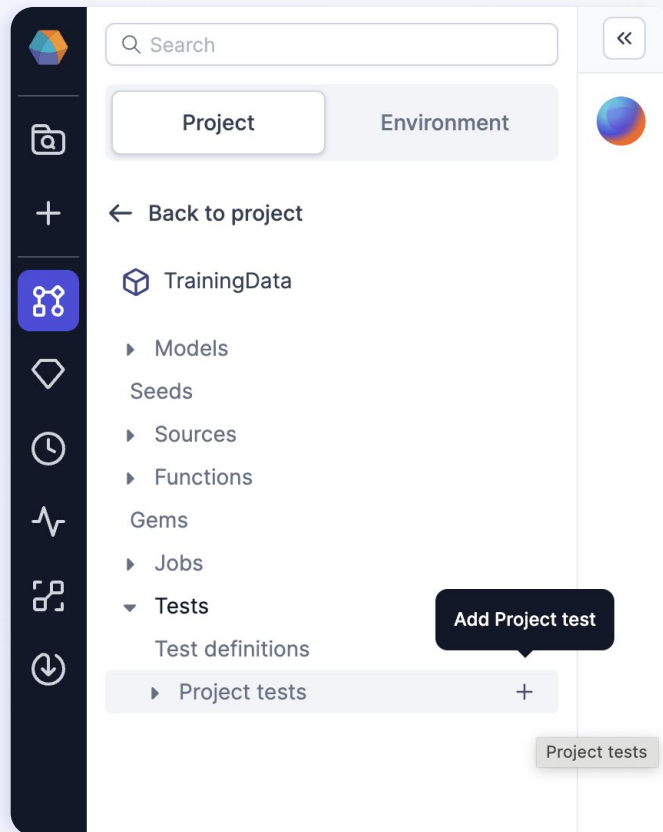
Project tests: single-use tests for models

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

The screenshot displays the Prophecy Copilot AI interface. On the left is a dark sidebar with navigation icons. The main area is divided into two panes. The left pane shows a project explorer for 'webinar_sql_co_pilot_db' with a tree view containing Models (customers_orders, stg_customers, stg_orders, stg_payments), Seeds (raw_clients, raw_orders, raw_payments), Sources (Ungrouped Tables), Functions, Gems, and Jobs. The right pane shows a search bar with the query 'top 50 customers by spent'. Below the search bar, a message indicates 'Showing Copilot suggestion in read-only mode. Copilot results may be incorrect or misleading.' The main workspace displays a data pipeline diagram with nodes for 'stg_customers', 'stg_orders', 'stg_payments', 'customer_ord...', 'total_spent_b...', and 'by_total_spe'. The pipeline starts with three model nodes on the left, which feed into a 'Join' node, followed by an 'Aggregate' node, and finally an 'OrderBy' node.

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

The screenshot displays the Prophecy Copilot AI interface, which integrates a chat-based workflow builder. The top navigation bar includes options for 'Visual' and 'Code' views, and a 'Prophecy' status indicator. The main workspace is divided into a chat area on the left and a visual pipeline on the right.

Chat Interface:

- Prompt 1:** "Build a report that presents top 50 customers by amounts" (with an edit icon).
- Response 1:** "Got it! Here's your report. It joins the orders and customers tables and sums the amounts spent by each customer. The report is saved as a Delta table."
- Prompt 2:** "Cool, now add an additional report that sums the revenue for each European country" (with an edit icon).
- Response 2:** "Using the same data, I've added an additional report that presents the revenue split by each European country."
- Prompt 3:** "FilterByCountry x Actually, could you expand the report to all of our geographies?"
- Response 3:** "No problem! I've removed the European focused filter expanding the aggregation to all geographies." (with 'Reject' and 'Continue' buttons).
- Prompt 4:** "Limit each report output to 100" (with a clear icon).

Visual Pipeline:

- Source:** A teal trapezoid icon representing the data source.
- (R) Subgraph:** A blue pentagon icon representing a subgraph or transformation step.
- Aggregation and Filtering:** A yellow box highlights the 'SumByCountry' step, which includes an 'Aggregate' node (purple pentagon) and an 'OrderBy' node (purple pentagon).
- Final Output:** The pipeline concludes with a 'CustomerCon...' node (purple pentagon) and an 'SQLStatement' node (purple pentagon).



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 DataClean interface. At the top, there are tabs for 'DataCleanUp', 'Config', and 'Schedule'. Below these are buttons for 'Source/Target', 'Transform', 'Join/Spilt', 'Custom', 'Machine Learning', and 'Subgraph'. The main area displays a data pipeline with the following steps: Employee... (Source), deduplicat... (Deduplicate), dynamic_s... (DynamicSelect), type-deno... (BulkColumnEx...), rename-d... (BulkColumnRe...), clean_de... (DataCleansing), reorder... (Reformat), and CleanedData (Target). Below the pipeline is a survey question: 'How likely are you to recommend Prophecy to a friend or colleague in the data space?'. The survey has a scale from 0 to 10, with 10 selected. Below the scale is a text input field with the placeholder 'Tell us a bit more about why you chose 10'. A 'Submit' button is at the bottom. The bottom right corner of the interface says '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

