The Synergy of Relational and Transformational Modeling

With some data mesh thrown in

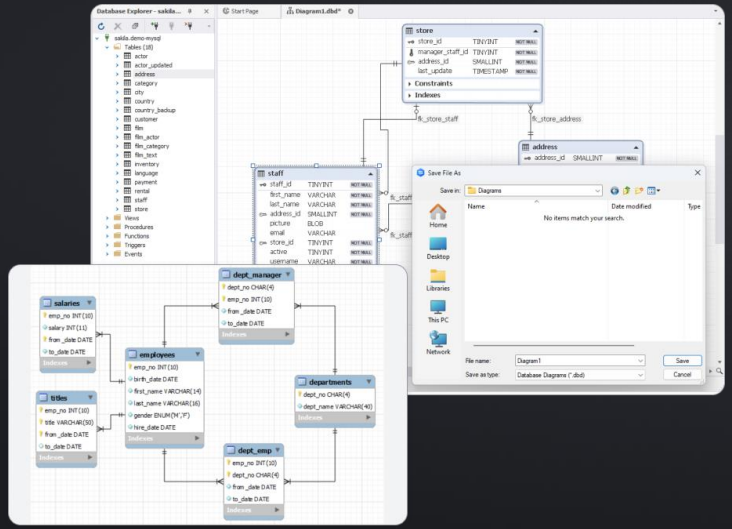
sqldbm.com

Agenda

1. Why data modeling matters
2. Collaborating across the organizations
3. SqlDBM overview
4. Concurrent working and integrations
5. Relational vs transformational (SqlDBM vs dbt)
6. Data governance
7. Q&A

What is data modeling?

What people think it is



What it actually is

sqldbm.com

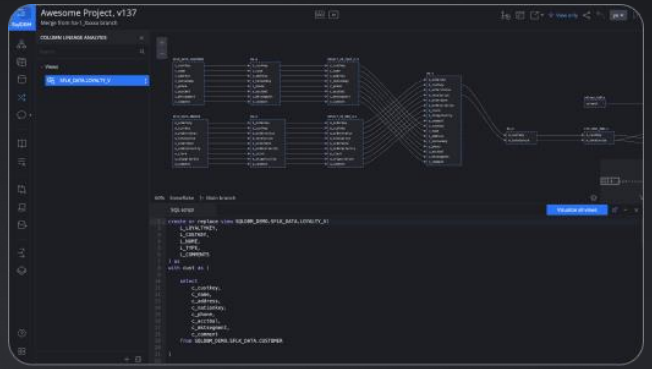
What else it is

SQL script

```
create or replace view SQLDBM_DEMO.SFLK_DATA.LOYALTY_V(
    L_LOYALTYKEY,
    L_CUSTKEY,
    L_NAME,
    L_TYPE,
    L_COMMENTS
) as
with cust as (

    select
        c_custkey,
        c_name,
        c_address,
        c_nationkey,
        c_phone,
        c_acctbal,
        c_mktsegment,
        c_comment
```

What it also is

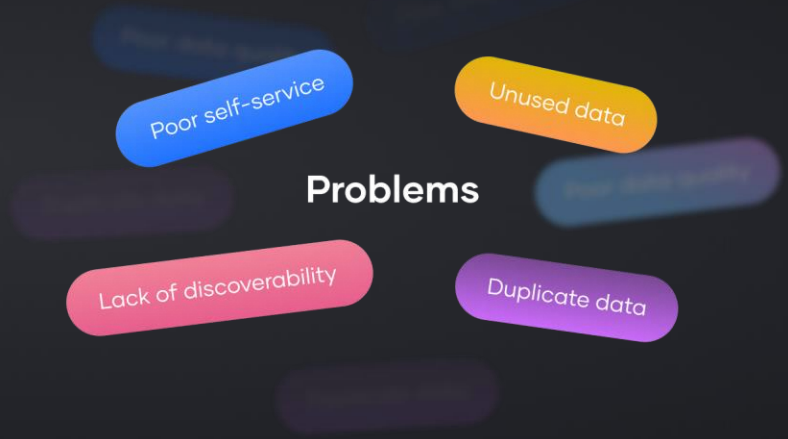


So, what can it do for you?

sqldbm.com

Problems related to lack of data modeling

- Duplicate or unused data
- Multiple names for common objects
- Poor self-service
- Lack of discoverability
- Meetings instead of documentation
- Poor data quality
- Business lack of trust in the data



“A lack of a data model is still a data model, albeit a crappy one.” - Joe Reis

Benefits associated with data modeling

- Consensus and visibility of the business model
- Reusability on subsequent projects and self-service
- Accelerated time to market and analytics
- Effective utilization of data assets unlocks their full value.
- Cross-platform, cross-domain, and universally understood
- Faster onboarding of new hires and consultants

TLDR;
Modeling saves time and time is money.

Our Differentiators

Ongoing Value

Receive monthly feature releases without the need to upgrade versions.

Anytime, anywhere

Access models from any location, and minimize downtime.

Empowering Teams

With our concurrent working decentralized teams can work collaboratively will focused on autonomous priorities. While following consistent standards and patterns.

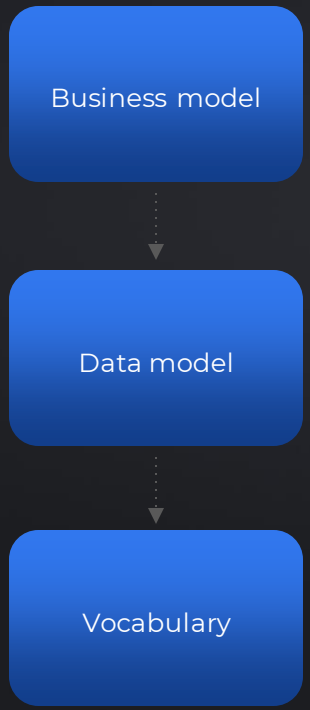
Accelerate Design

Accelerate your time to market leveraging essential features such as naming standards, column templates and table templates

SqlDBM

InnerJoin

What does data modeling consist of?

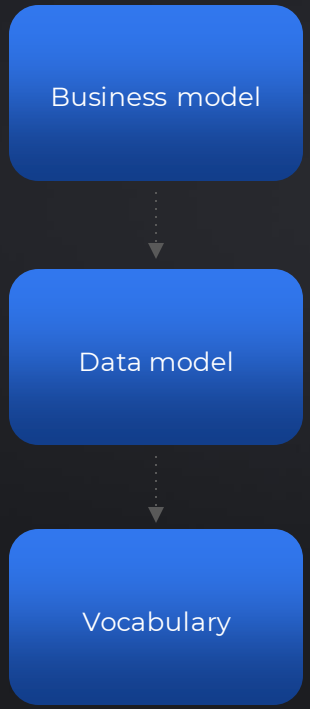


What we do?

What data about what we do, do we care about?

How do we “encode” these things for effective communication?

For example



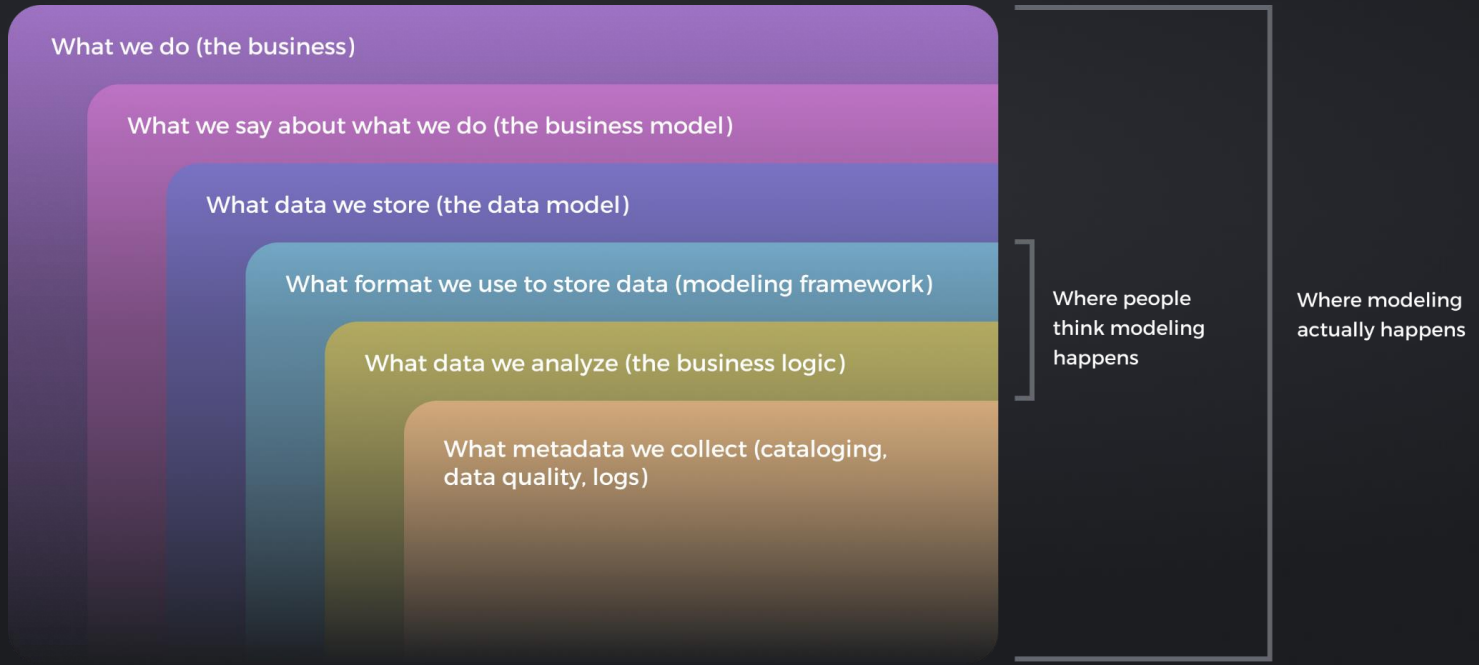
We sell services to subscriptions

Concepts	Collateral *(Ontology)*
Service, service type, subscriber	Entity, Relationship, ERD, Data Dictionary

Active subscriber, type 2 dim, normalization, logical deletion

Modeling is a team sport

Getting buy-in from business teams and product teams



Leveraging and delegating data modeling know-how in the organization

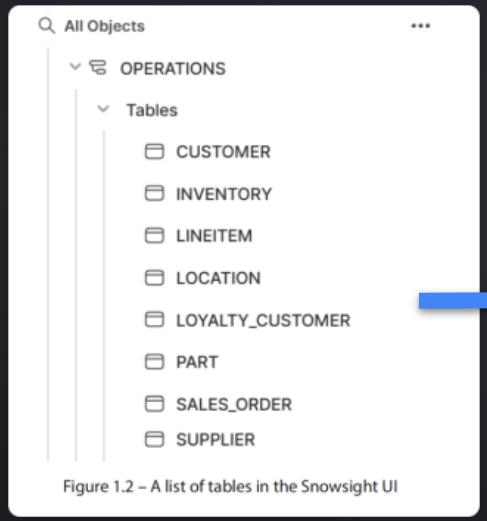


Figure 1.2 – A list of tables in the Snowsight UI

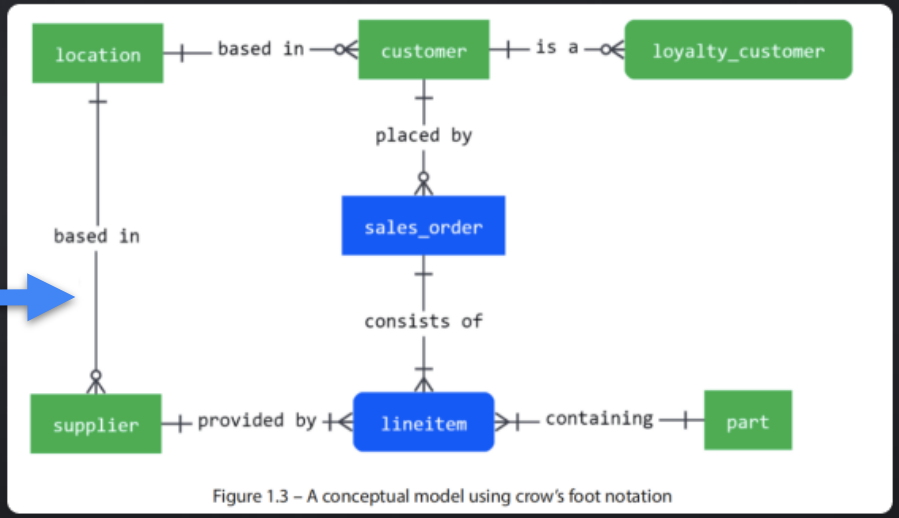
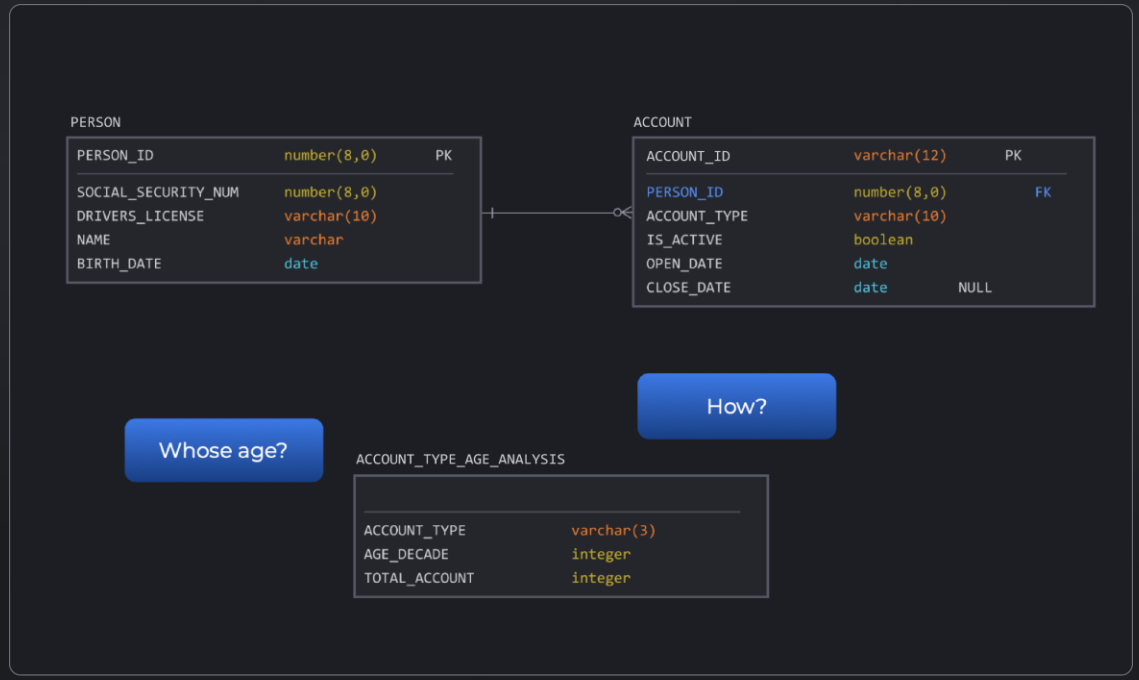


Figure 1.3 – A conceptual model using crow's foot notation

"There needs but one wise man in a company and all are wise, so rapid is the contagion." - Ralph Waldo Emerson

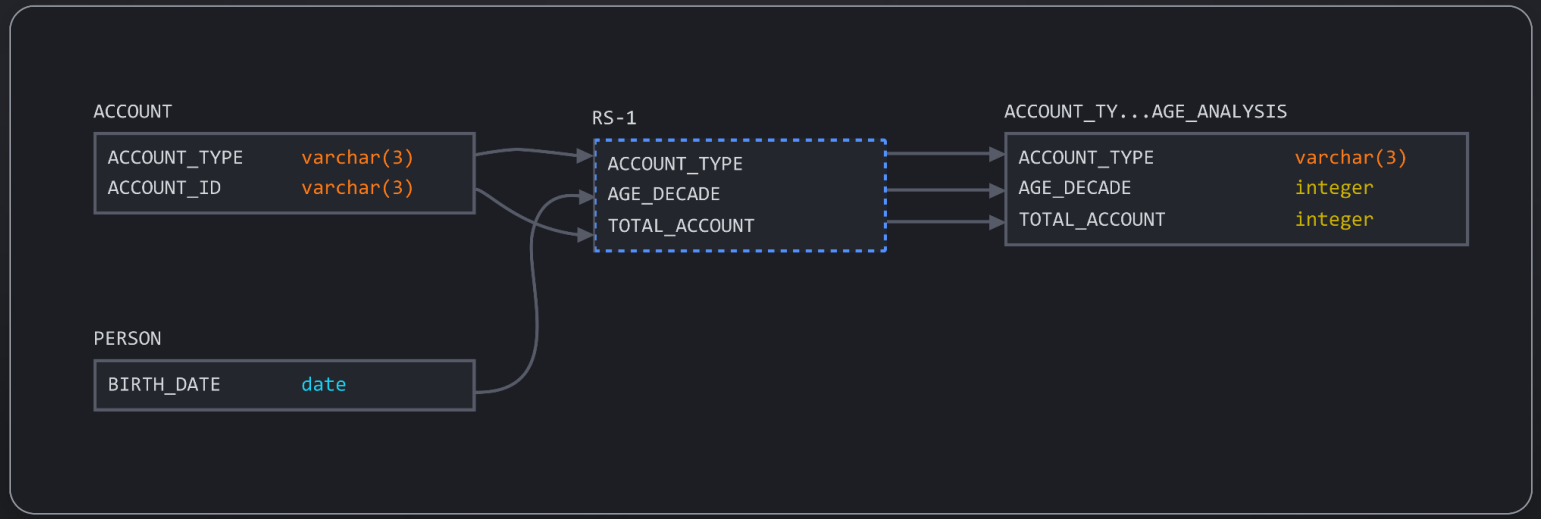
Relational vs transformational modeling (SqlDBM vs dbt)



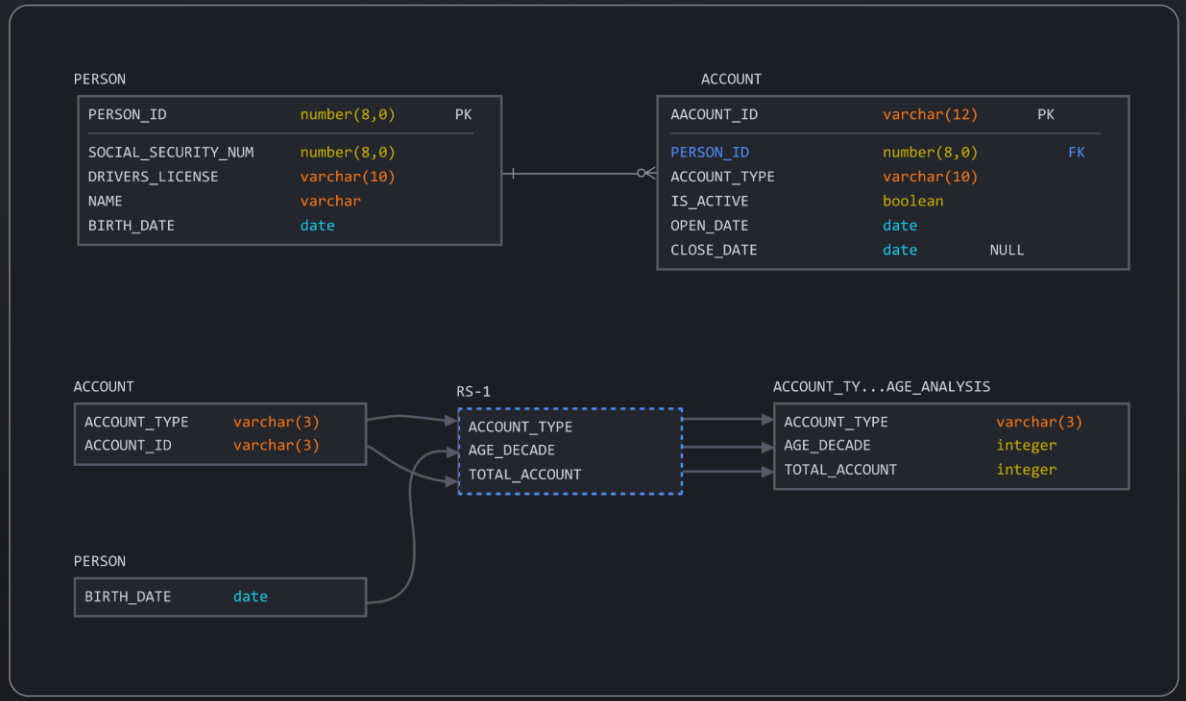
Relational vs transformational modeling (SqlDBM vs dbt)

```
CREATE TABLE account_types_age_analysis AS
SELECT
 a.account_type,
 ROUND(DATEDIFF(years, p.birth_date, CURRENT_DATE()), -1
 ) AS age_decade,
 COUNT(a.account_id) AS total_accounts
 FROM account AS a
 INNER JOIN person AS p
 ON a.person_id = p.person_id
GROUP BY 1, 2;
```

Relational vs transformational modeling (SqlDBM vs dbt)



Relational vs transformational modeling (SqlDBM + dbt)

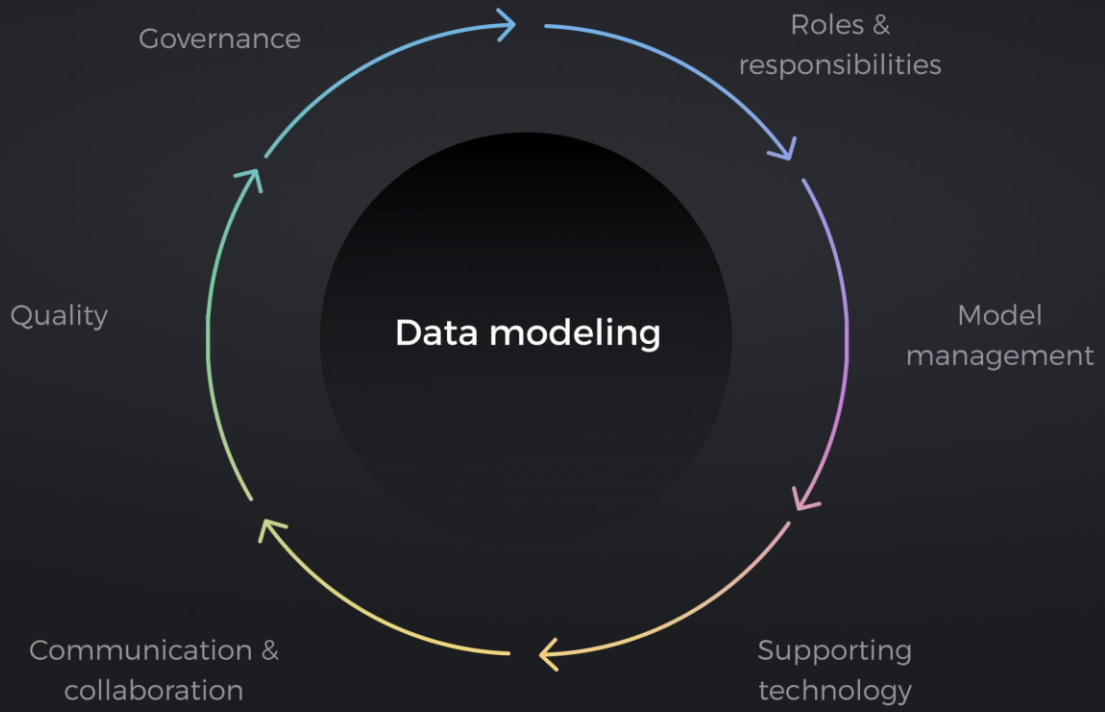


sqldbm.com

Establishing a data modeling framework

- Nominate a champion
- Obtain buy-in
- Create a process
- Demonstrate value early
- Democratize it (make it accessible to everyone)

Data modeling framework



Data modeling framework

Roles & Responsibilities

Define the key roles involved in data modeling within the organization, such as Data Architects, Data Modelers, and Data Stewards, and outlines their specific responsibilities and contributions to the data modeling process, ensuring clear accountability and effective collaboration

Model Management

Establishing a structured process for the lifecycle of data models, from conception to deployment. It encompasses practices for ensuring consistency, compliance with business requirements, and effective version control, thereby enhancing the integrity and usability of data models.

Supporting Technology

Identifying and standardizing the tools and technologies used in data modeling, such as specific modeling software and version control systems. It's about leveraging technology to improve efficiency, ensure consistency, and support the various stages of data modeling.

Data modeling framework

Communication & Collaboration

Emphasize the importance of clear communication and collaborative practices in data modeling. It includes strategies for sharing information, fostering teamwork across different roles, and ensuring that data modeling efforts are aligned with organizational goals and integrated with other business processes.

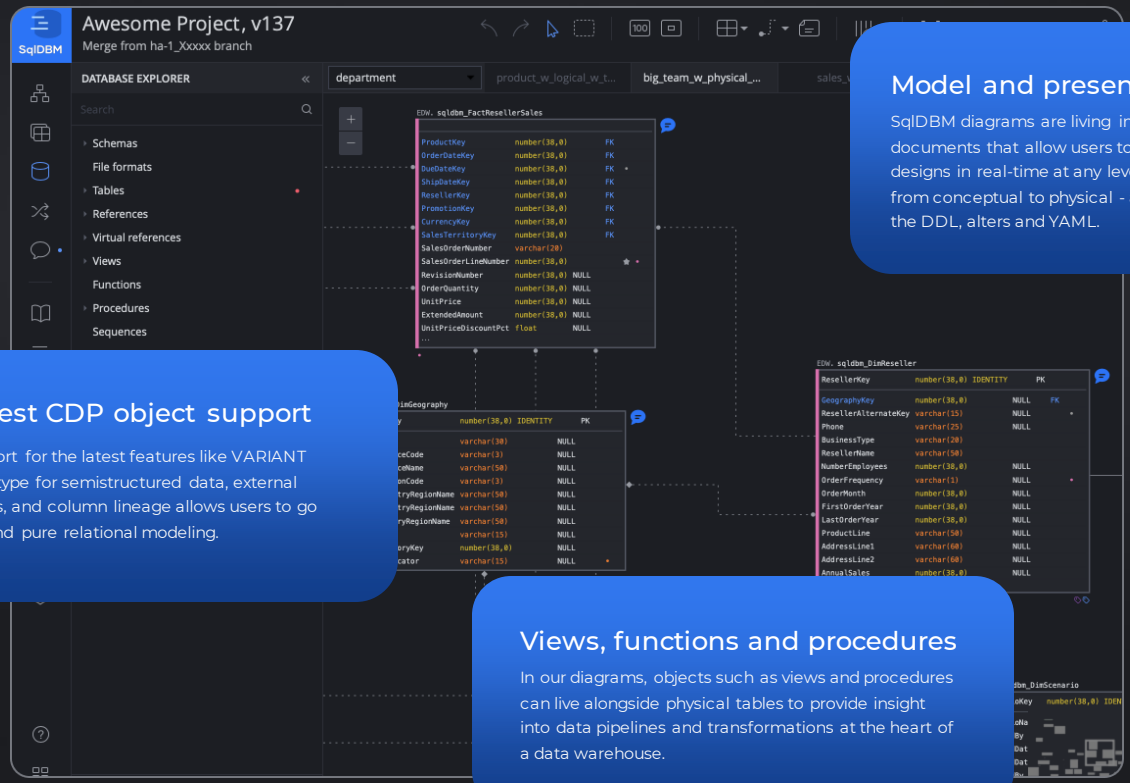
Quality

Maintaining high standards of quality in data models. It involves setting quality benchmarks, regular reviews and audits of data models, and implementing feedback mechanisms to continuously improve the accuracy, efficiency, and effectiveness of the data modeling process.

Governance

Establishment of policies and standards for data naming, metadata management, and model architecture. It ensures that data models adhere to organizational standards, legal compliance, and best practices, thereby safeguarding data integrity and facilitating effective data management.

Let’s talk about the product



Model and present

SqlDBM diagrams are living interactive documents that allow users to present their designs in real-time at any level of detail - from conceptual to physical - and generate the DDL, alters and YAML.

Latest CDP object support

Support for the latest features like VARIANT data type for semistructured data, external tables, and column lineage allows users to go beyond pure relational modeling.

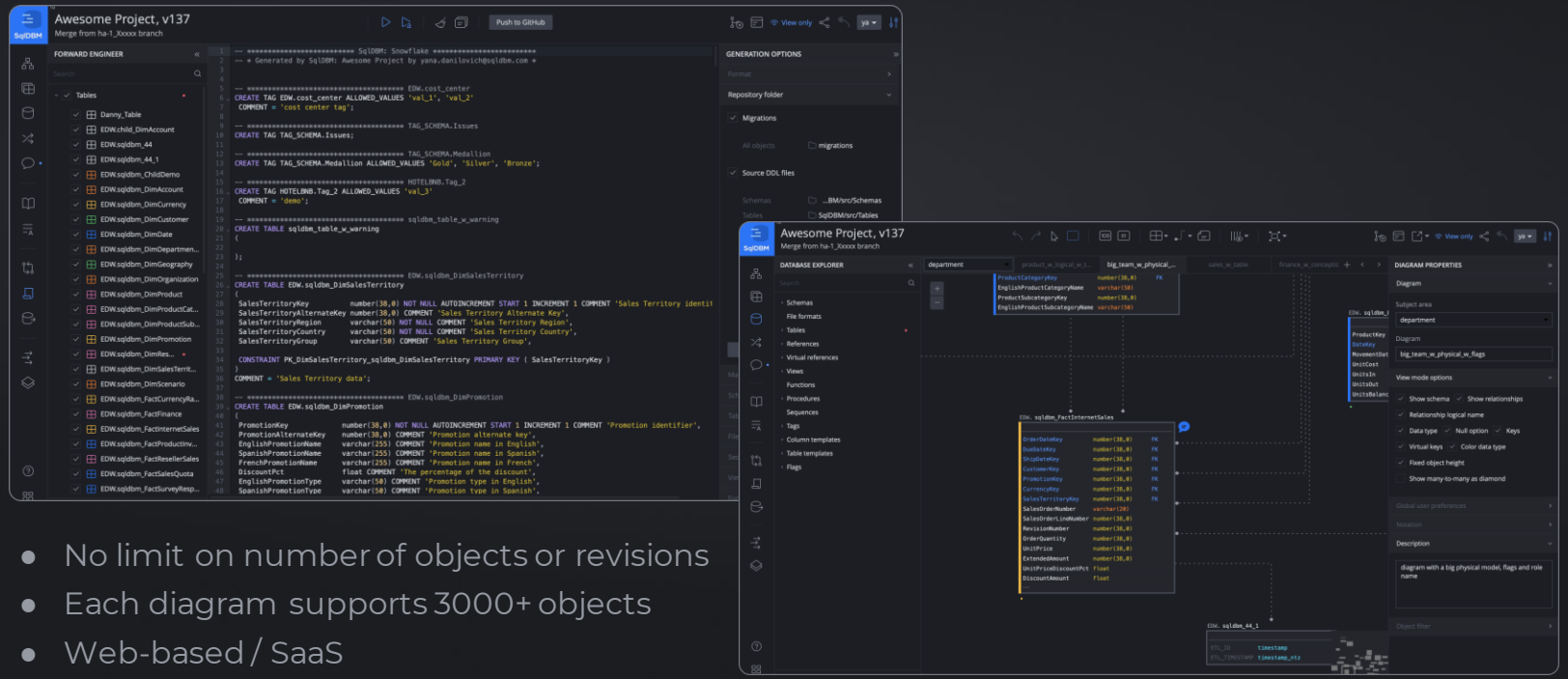
Views, functions and procedures

In our diagrams, objects such as views and procedures can live alongside physical tables to provide insight into data pipelines and transformations at the heart of a data warehouse.

sqldbm.com

SqlDBM basics

Unlimited projects, unlimited change history, unlimited objects



- No limit on number of objects or revisions
- Each diagram supports 3000+ objects
- Web-based / SaaS
- Project-based

sqldbm.com

Let’s talk about the product

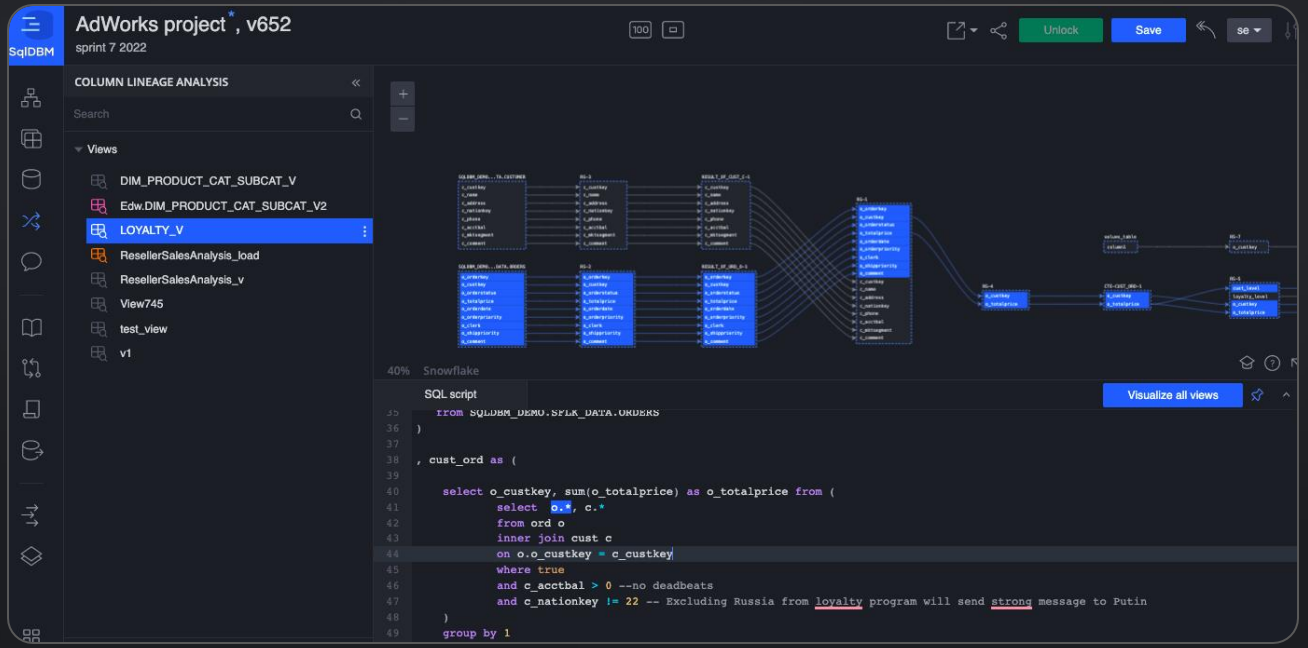
Relational modeling

Model-driven collaboration, governance, and self-service

- ✓ Visualize
- ✓ Relate
- ✓ Search
- ✓ Design
- ✓ Filter
- ✓ Deploy
- ✓ Govern
- ✓ Monitor
- ✓ Discover
- ✓ Integrate

Column view lineage

Break down complex data flows with column-level lineage and substep visualization. Take advantage of syntax highlighting from the diagram as well as code editor to see even real-time changes.



Main Menu

Use Main Menu to switch between other projects (Dashboard), team management (Project-Team), and App theme settings. You also can convert your project to other DB/DW type.

Add Objects

Click to Add Table, Relationship, or a Note. Objects can also be added by right-clicking on the diagram canvas or through keyboard shortcuts.

Export

Export the diagram to PNG or SVG.

Team Collaboration

Invite team members to collaborate on a project. Grant view/edit permissions

Object Explorer

Switch to see all diagrams and subject areas (Diagrams tab), objects of diagram (Diagram explorer tab) or objects of database (Database Explorer tab).

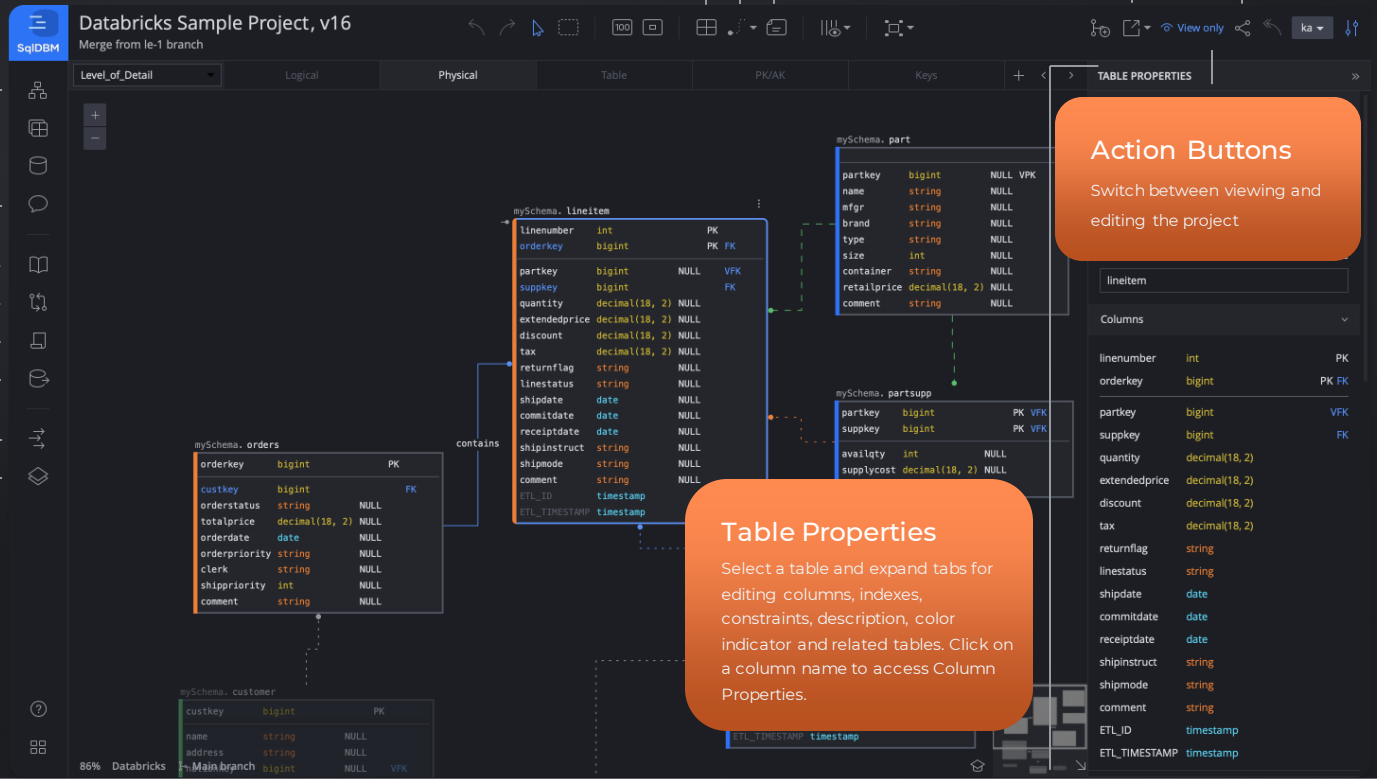
Database Documentation

Compare Revisions

Forward Engineer

Reverse Engineer

DataOps



Action Buttons

Switch between viewing and editing the project

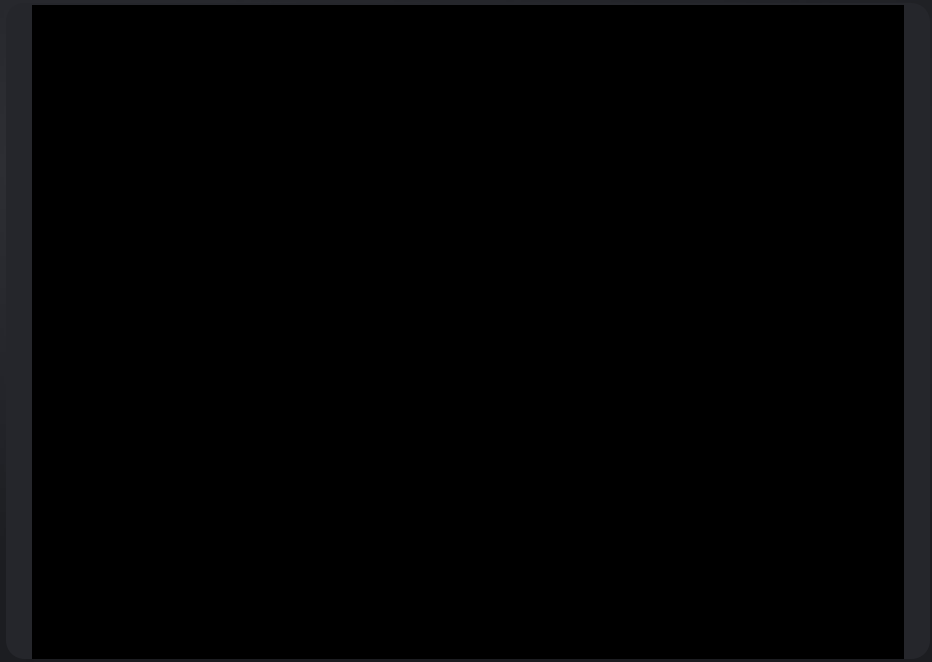
Table Properties

Select a table and expand tabs for editing columns, indexes, constraints, description, color indicator and related tables. Click on a column name to access Column Properties.

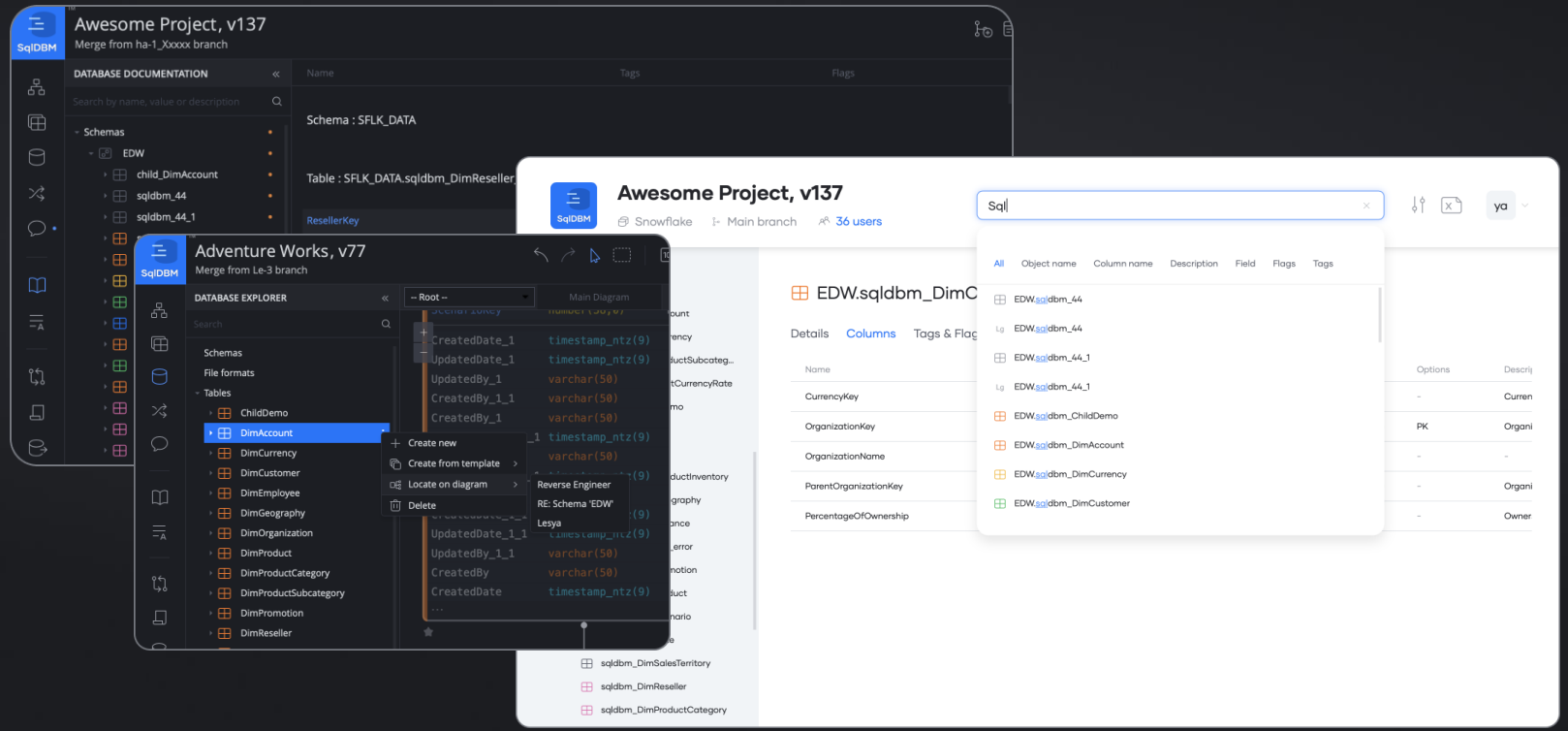
sqldbm.com

How to rely on standards and templates to keep things organized

- Establishing consistency
- SqlDBM conventions
  - Case
  - Object
- Column templates
- Table templates



Discoverability



sqldbm.com

Documentation and governance

Enhancing data documentation features for successful data governance framework

Governance
Project Role

Custom fields

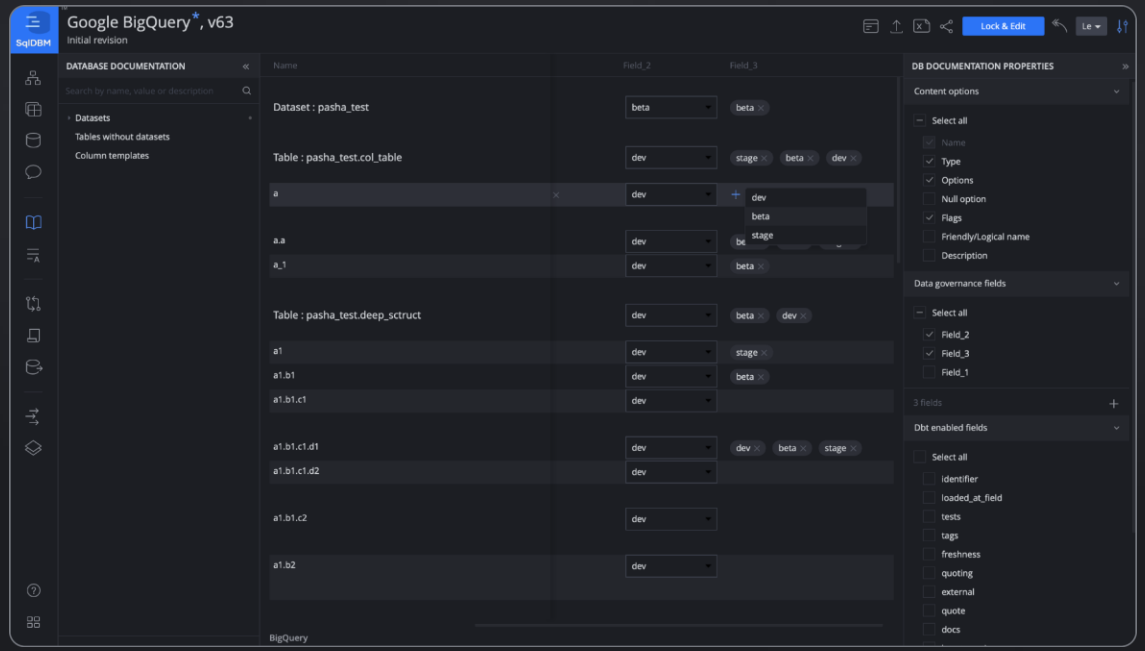
Pages

Flag and metadata
search

DB Documentation
Comments

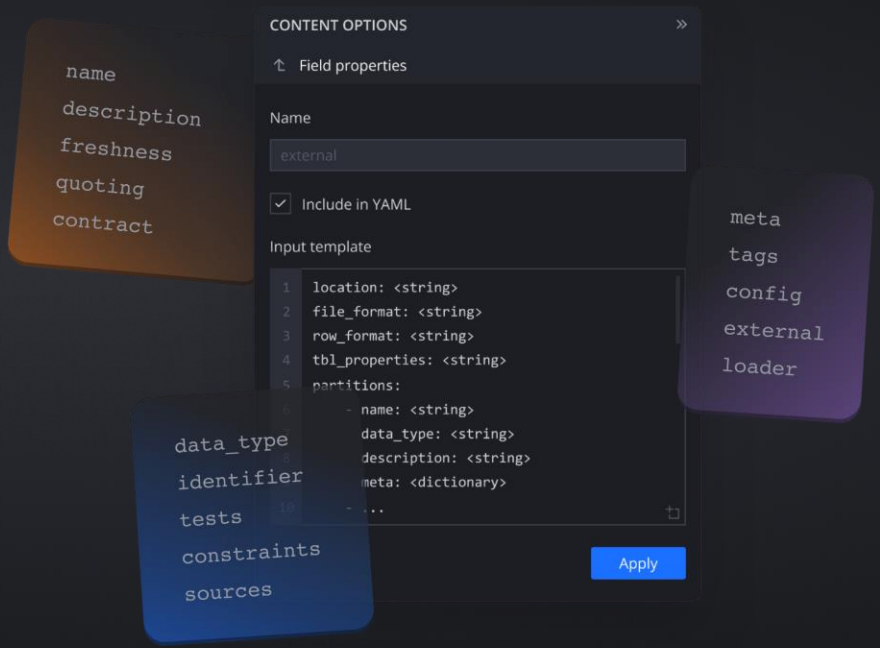
Custom fields

Fields are additional metadata columns that can be created and maintained by users with the Governance role to provide additional attributes (e.g., owner, is validated, source) to objects within the project.



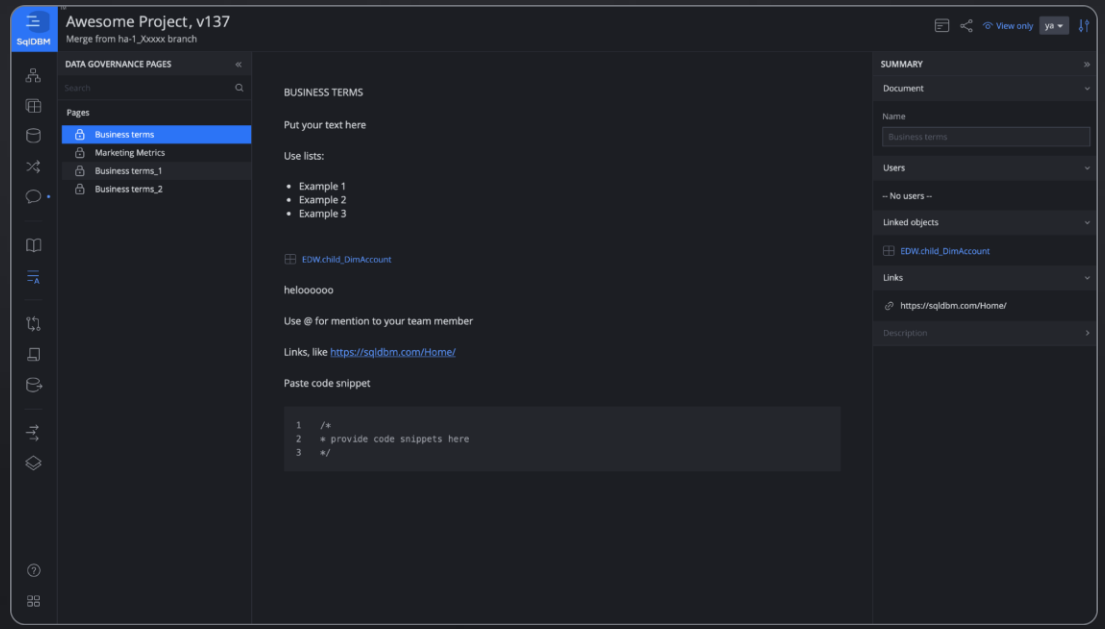
All dbt properties supported

Maintain all the supported dbt properties like tests and freshness at object or column level, all through pre-configured templates and visual interface.



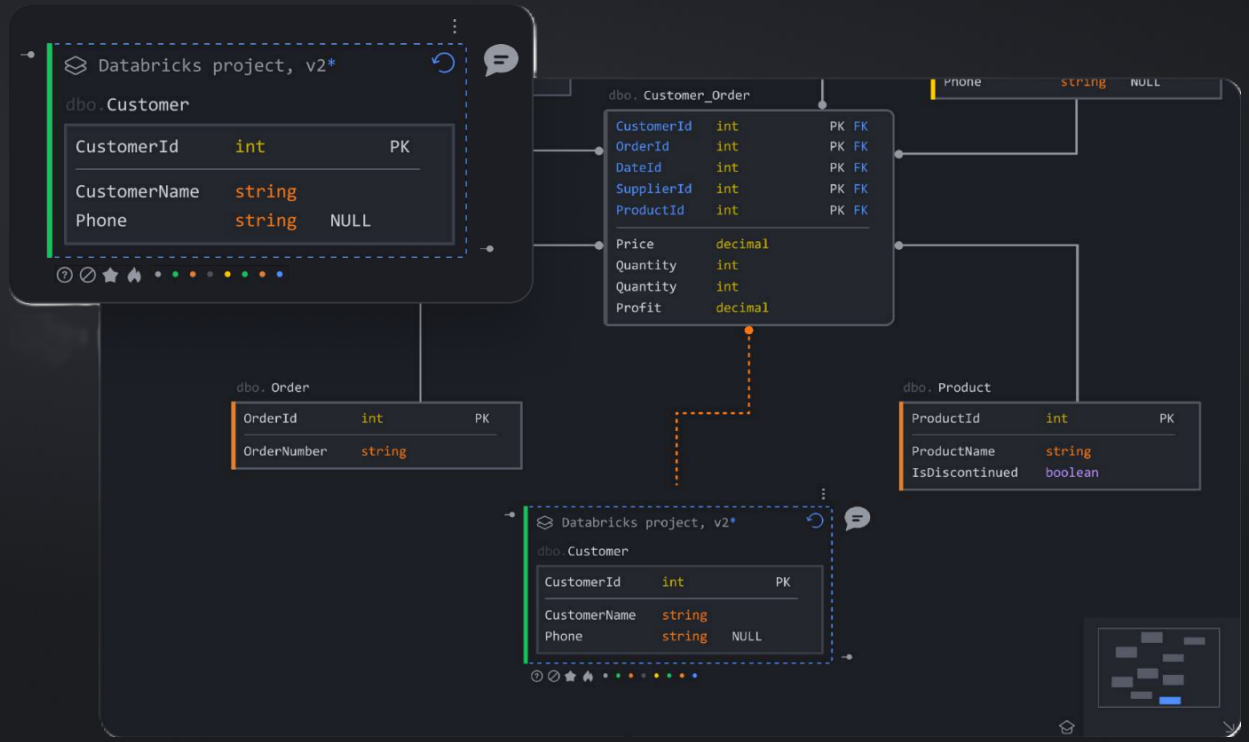
Pages

Extend documentation beyond the object level with rich text full-length content, embedded images, diagrams, styling, and formatting. Pages allow teams to maintain project-level overviews of things like metrics and KPI definitions, business terms, and operational outlines.

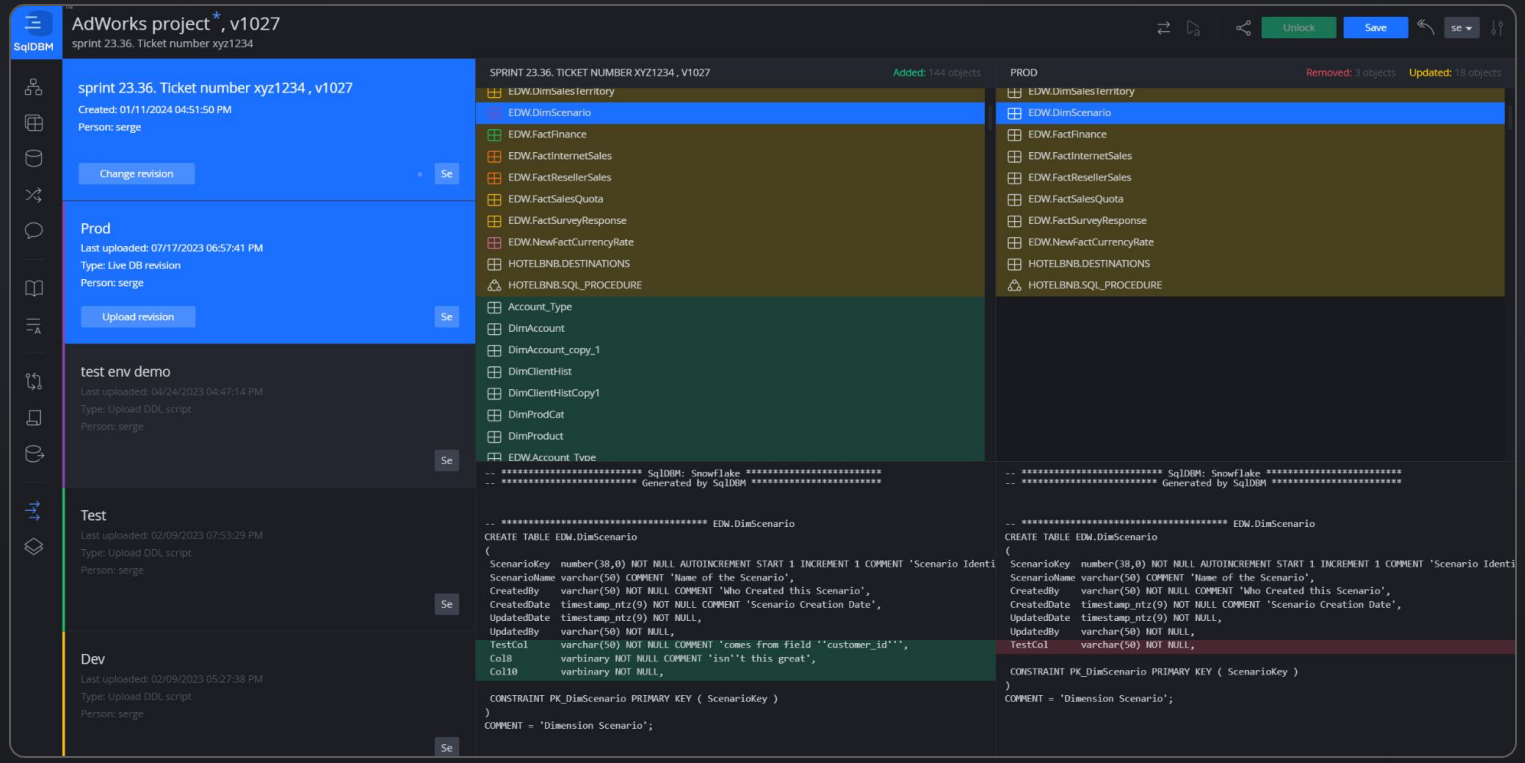


Global Modeling

Work across projects and business domains with object reference and change notification. Global modeling unlocks the potential of Data Mesh through SqlDBM's all-in-one UI for search and discovery.

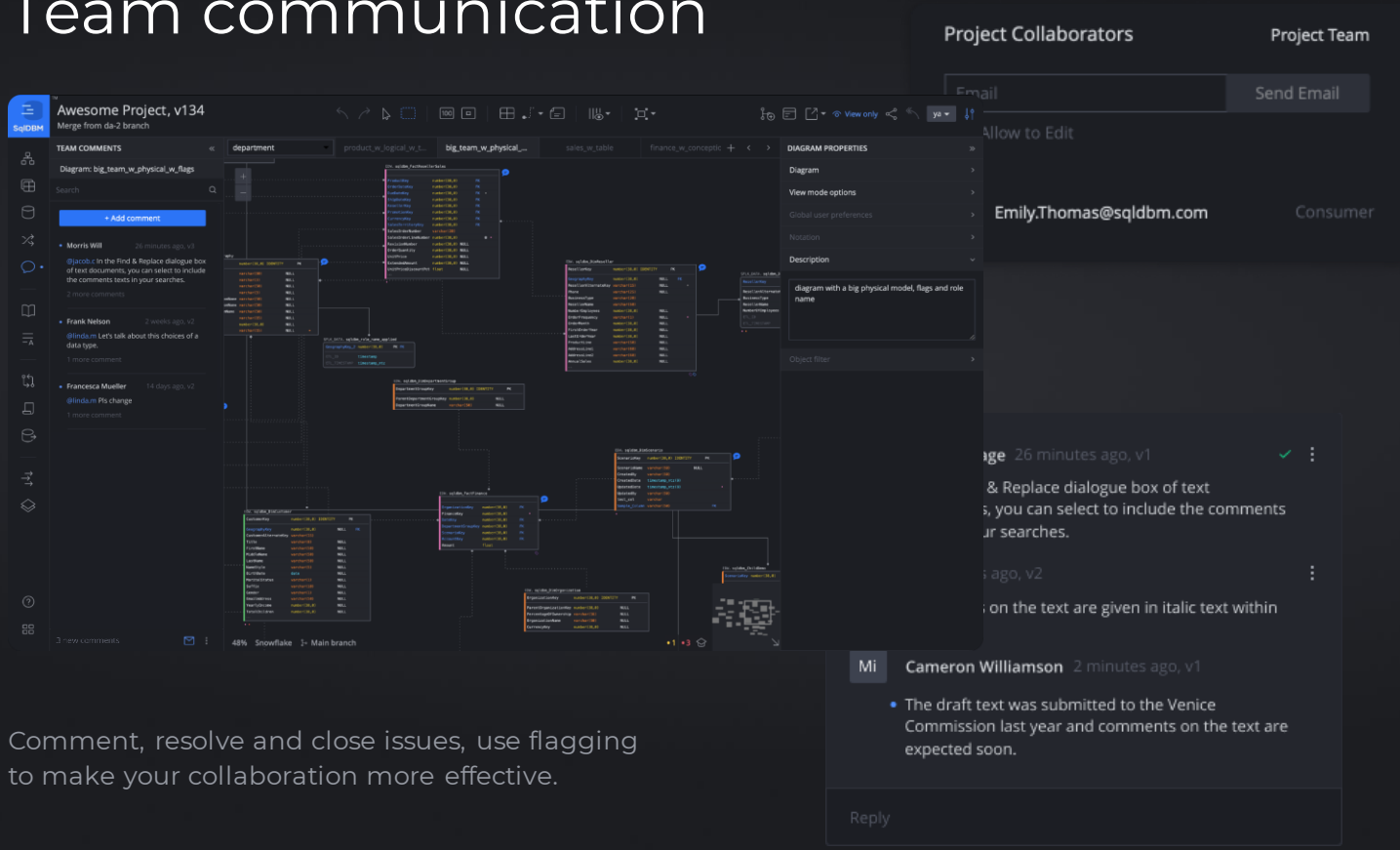


Forward engineering and CICD



sqldbm.com

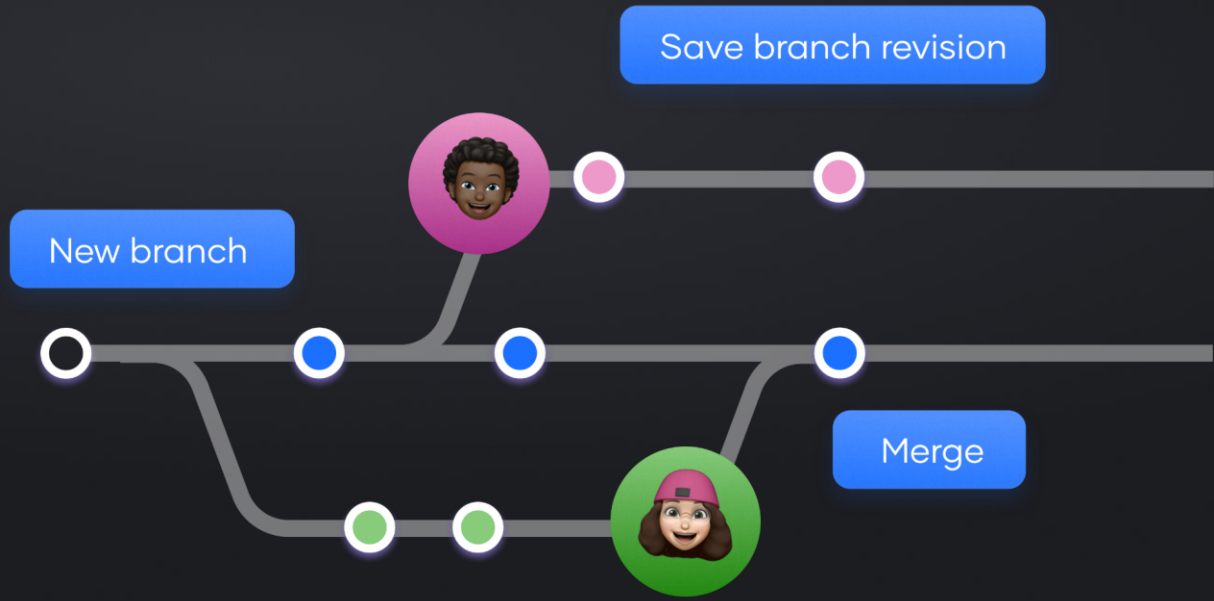
Team communication



Comment, resolve and close issues, use flagging to make your collaboration more effective.

Concurrent modelling

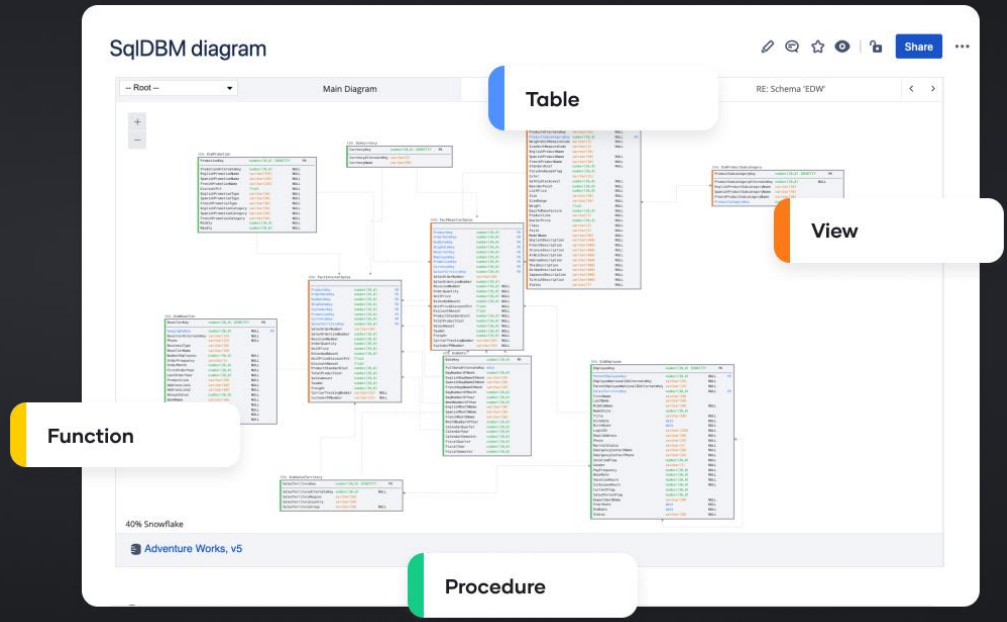
The entire project team can work collaboratively and in parallel without having to lock a project. Changes made in any branch can be compared before merging to main and synchronized with any new updates that happen meanwhile.



Confluence

Share diagrams via Confluence

SqlDBM diagrams are now embeddable directly in Confluence. Share live, dynamic diagrams with your entire organization without needing to access SqlDBM or creating individual viewer accounts



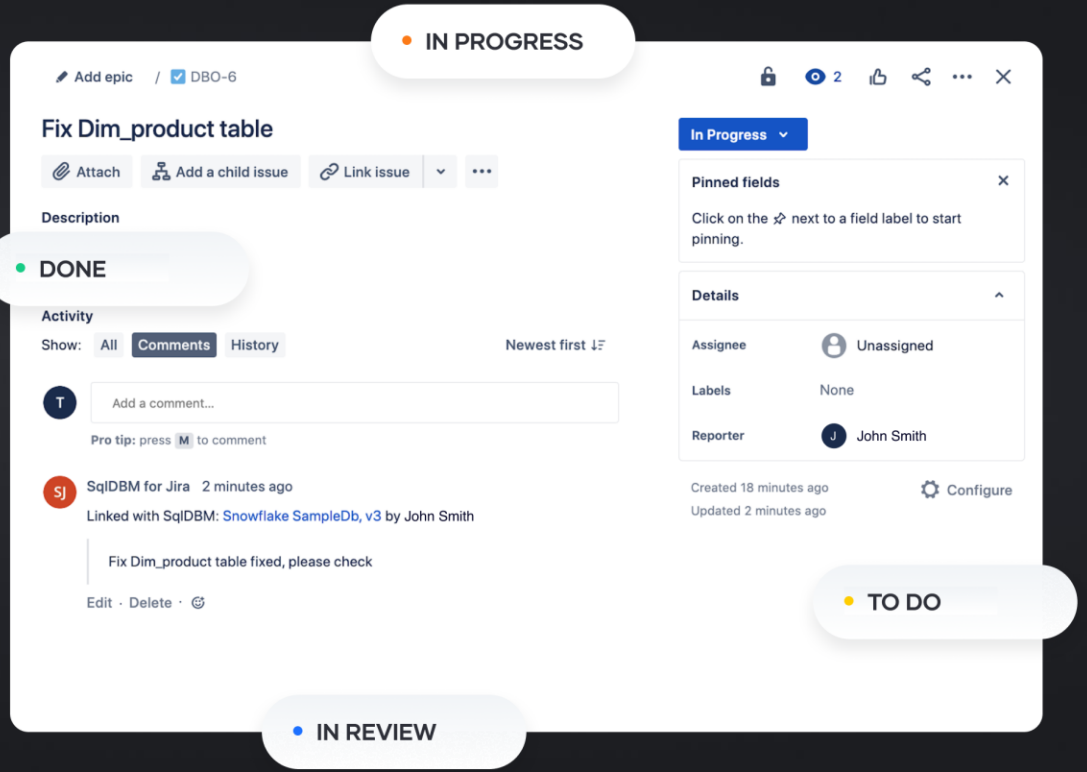
sqldbm.com

Jira

Jira integration

Linking to related Jira tickets and issues ties project changes to the relevant business context and audit trails.

The integration will allow users to jump from a revision straight to a Jira ticket and vice versa - even from diagram comments!

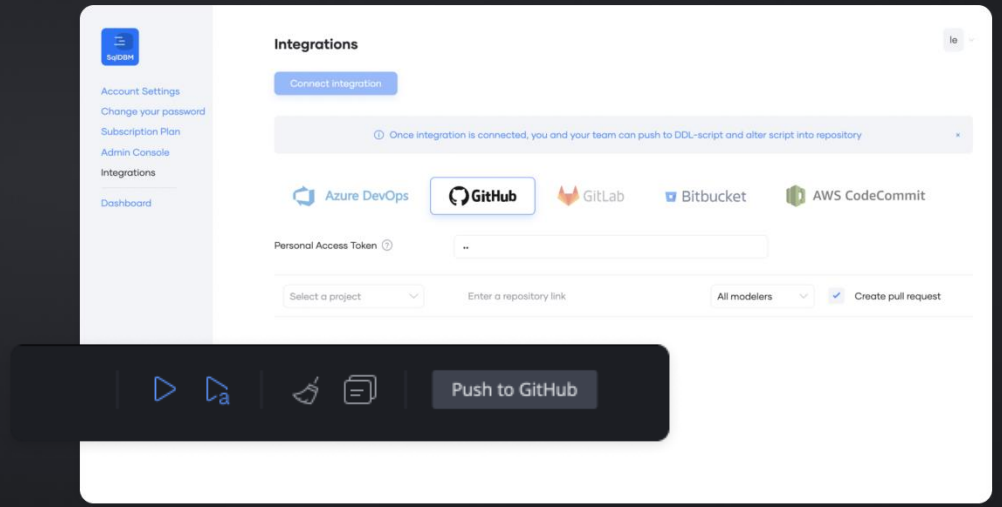


sqldbm.com

GIT

Materialize your changes and push to Git

Compare project revisions to live DB environments and generate ad-hoc alter scripts between them. Pair your project with any online git repository to automatically push alters or individual objects to new branches and pair with CICD tools like Schemachange or Flyway for fully automated deployments.

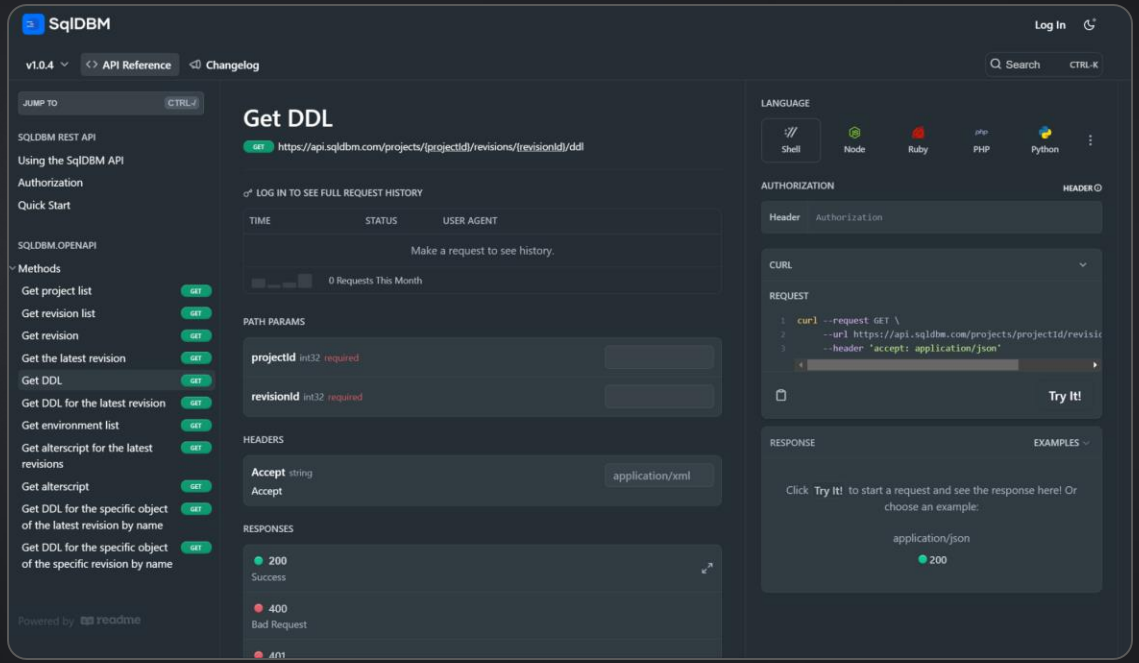


sqldbm.com

API

Interacting with your data model through a programmatic interface unlocks exciting options for data developers and engineers.

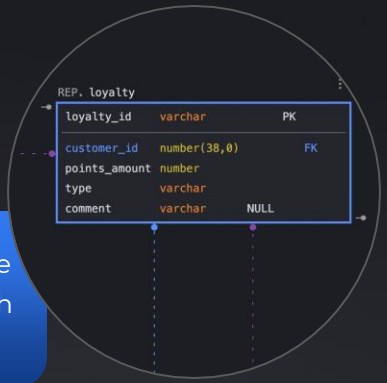
From sharing documentation with data catalogs to automating a CICD pipeline, APIs unlock a world of possibilities for engaging with SqlDBM models.



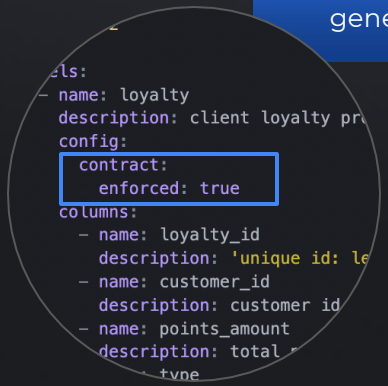
Get it right the first time!

1. Self-serve data discovery and relationship insights

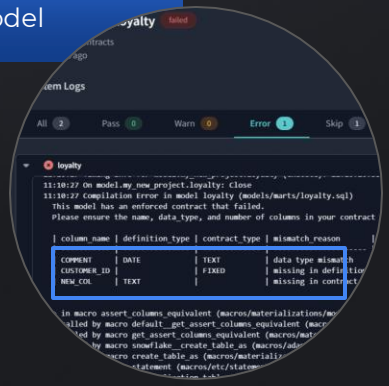
2. Architect and validate high-level design with stakeholders



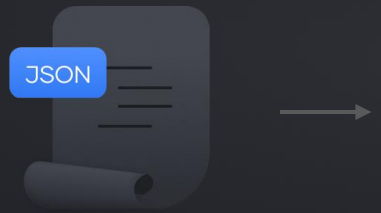
3. Data model becomes dbt contract in generated YAML



4. dbt contact enforces agreed-upon design in the dbt model



Future Enhancements



Manifest.json