

Veeva CDB — Clinical Data Review UI

Clinical Database · Veeva Vault Platform

THE PROBLEM

Clinical trial data is generated across multiple disconnected systems — EDC, RTSM, eCOA, labs, imaging, and external vendors. Before Veeva CDB, data managers had to:

- reconcile data using spreadsheet trackers
- run batch exports to review updates
- manually detect discrepancies
- communicate queries across siloed tools

This created **slow review cycles, high manual effort, and delayed database lock**, directly impacting trial timelines.

At the same time, reviewers from different functions (data management, medical, biostats) needed a **real-time, collaborative UI** to:

- detect new or changed data
- raise and resolve queries
- monitor study health

within a single system.

THE SOLUTION

I built a **full-stack system for the Clinical Data Review workflows** for the CDB (Clinical Database) application.

The core challenge was designing a **high-performance, data-dense interface** that could:

- *display harmonized clinical data from multiple sources*
- *surface change detection in real time*
- *support query lifecycle actions*
- *enable cross-functional collaboration*

Frontend

- *Developed modular UI in **React.js***
- *Implemented complex, state-driven data review screens*
- *Built interactive tables, filters, and review workflows for large clinical datasets*
- *Focused on responsiveness and usability for data managers reviewing high-volume studies*

Backend Integration

- *Consumed **Java APIs** for clinical data and query management*
- *Worked with a **SQL data model** aligned to the study backbone*
- *Leveraged **Java Streams** in service logic to process and transform data in a declarative, functional style*
 - *improved readability and maintainability*
 - *enabled efficient aggregation and parallelizable operations*

Data Experience

- *Enabled **real-time visibility of new and updated data***
- *Supported **automated discrepancy detection and query generation***

- Helped replace spreadsheet-based workflows with an in-app workbench

The screenshot shows a web-based application window titled "CDB WORKBENCH". The URL bar indicates the site is www.veeva.com. The main content area displays a table of clinical study data. A callout bubble on the left side says "Clean data centrally with smart automation". Another callout bubble on the right side says "Direct access to clean concurrent data".

Subject	Subject Status	Last Event	Outstanding Queries	Overdue Events	Incomplete SDV	Clean	Summary
CA-401 -0001	Enrolled	DAY8	4	0	7	False	
CA-401 -0002	Screen Failure	SCR	2	0	1	False	
CA-401 -0003	In Screening	SCR	3	0	2	False	
CA-401 -0004	Randomized	SCR	6	0	2	False	
CA-401 -0005	In Screening	SCR	7	0	2	False	
CA-401 -0006	In Screening	SCR	1	1	2	False	
CA-401 -0007	In Screening	SCR	0	0	0	True	
CA-401 -0008	Withdrawn	SCR	1	1	2	False	
CA-401 -0009	Enrolled	DAY16	0	1	2	False	
CA-401 -0010	End of Treatment	DAY16	1	0	2	False	
CA-401 -0011	Pre Screen	SCR	1	0	2	False	
CA-401 -0012	Enrolled	DAY1	1	1	2	False	

THE PROOF

The UI shipped as part of the production release of Veeva Systems's clinical data platform and is used by sponsor and CRO study teams to:

- review and clean data across sources in one environment
- collaborate on queries in real time
- monitor study progress through dashboards

Contributing to outcomes such as:

- **30–50% reduction in manual data cleaning effort**

- *faster time to database lock*
 - *real-time access to aggregated clinical data*
-

STACK

React.js · Java · Java Streams · SQL · REST APIs · Veeva Vault Platform

TL;DR (Summary)

As a Software Engineer at Veeva, I built React-based data review workflows for Veeva's clinical data platform, integrating with Java/SQL services to enable real-time aggregation, discrepancy detection, and cross-functional query management across EDC and third-party sources.