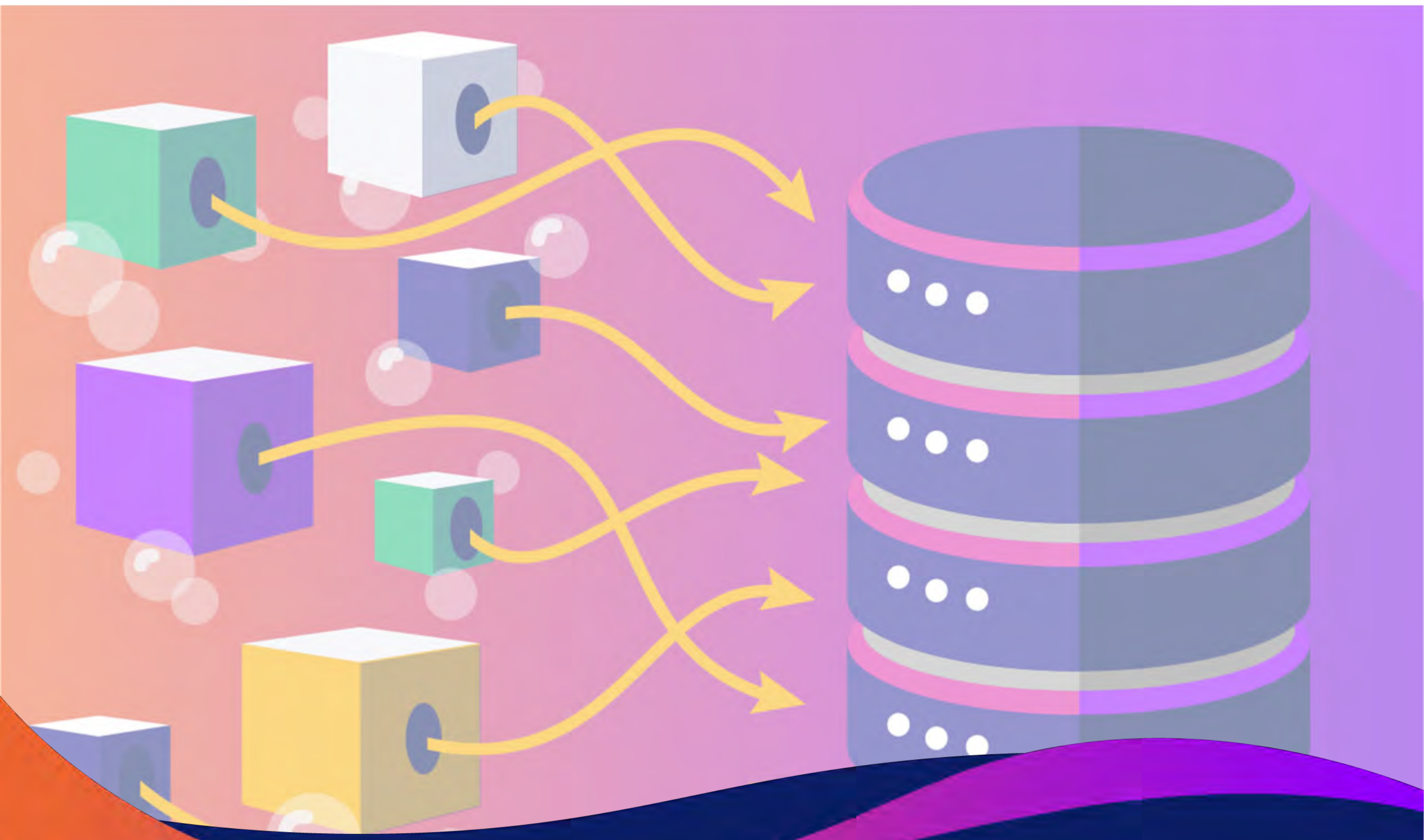
eBook

Database Migrations: 6 Answers to Know Before You Start



CodeLogic

Reveal. Optimize. Innovate.

The Challenge

Many organizations pay a hefty price for their database management platform. With more cost-friendly, feature-rich alternatives on the market now, organizations are eager to abandon expensive vendors and migrate to a less expensive option. While there are many tools available to help teams migrate data from an old platform to a new one, there's more to it to achieve success. Teams must be able to see all the connection points between the app and the database in order to see lasting results within their database migration – something that tools from even web services vendors such as Amazon, Google and Microsoft cannot achieve. Without an accurate, up to date view of all the app-to-database connections and dependencies, database migration efforts are likely to be costly, time consuming and prone to endless errors.

Our Solution

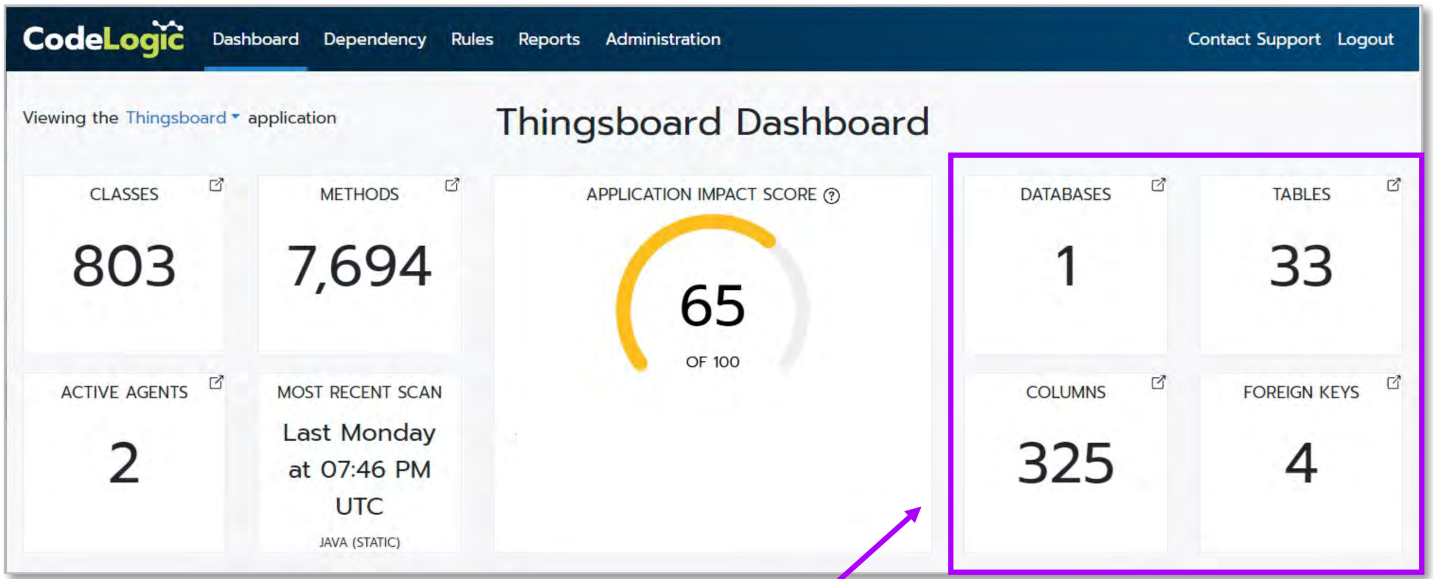
CodeLogic visualizations and dashboards make it easier see every database and dependency connection across an application. With CodeLogic, application teams no longer have to spend hours or even days crawling through code to identify and document where every method connects to a database. Using binary and runtime scanning, CodeLogic identifies every connection point between an application and a database in minutes. Our solution documents these connections in an intelligent dashboard that surfaces application complexity and allows teams to analyze dependencies in visualization maps down to the code level. With this depth of application insight, teams can better estimate their platform migration effort, develop a realistic migration plan, and ensure that any error-prone loose ends are identified in advance.

Our Approach

CodeLogic provides the industry's most comprehensive, self-updating application dependency mapping and impact analysis tool available. Our approach delivers the application intelligence that teams need to be more productive, better understand code complexity, and make more informed decisions about application and database changes. Teams using CodeLogic can map, document, visualize, and understand their application connections and dependencies from class to method to database – on schedule, on-demand, or after every build.

Question 1

How many tables and columns are in my database?



Viewing the Thingsboard application

Thingsboard Dashboard

CLASSES	803
METHODS	7,694
APPLICATION IMPACT SCORE	65 OF 100
DATABASES	1
TABLES	33
ACTIVE AGENTS	2
MOST RECENT SCAN	Last Monday at 07:46 PM UTC JAVA (STATIC)
COLUMNS	325
FOREIGN KEYS	4

The Application Dashboard highlights the tables and columns in this application's database

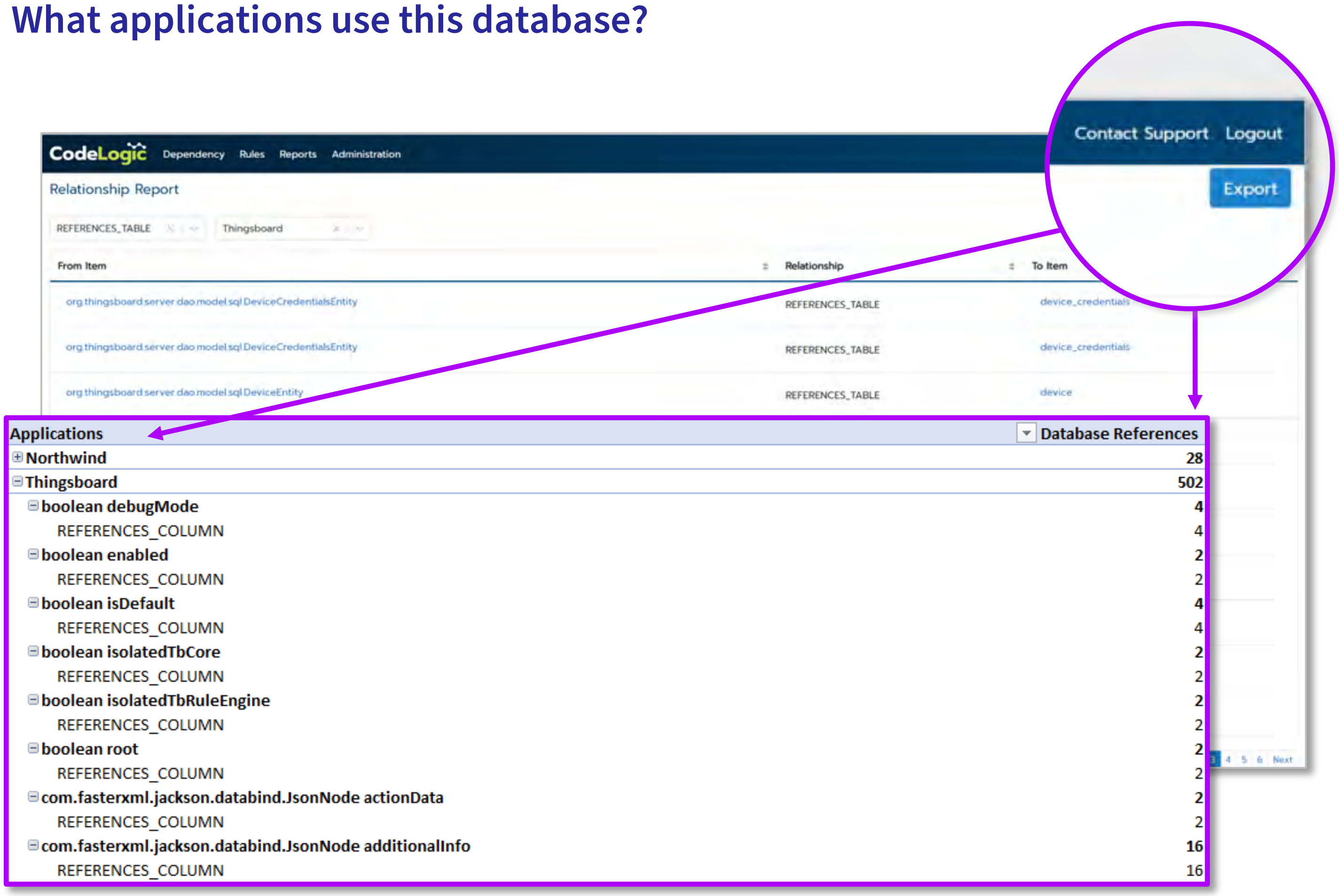
CodeLogic Answer

The CodeLogic Application Dashboard provides clear sizing of the database and the related application code.

CodeLogic

Question 2

What applications use this database?



From Item	Relationship	To Item
org.thingsboard.server.dao.model.sql.DeviceCredentialsEntity	REFERENCES_TABLE	device_credentials
org.thingsboard.server.dao.model.sql.DeviceCredentialsEntity	REFERENCES_TABLE	device_credentials
org.thingsboard.server.dao.model.sql.DeviceEntity	REFERENCES_TABLE	device

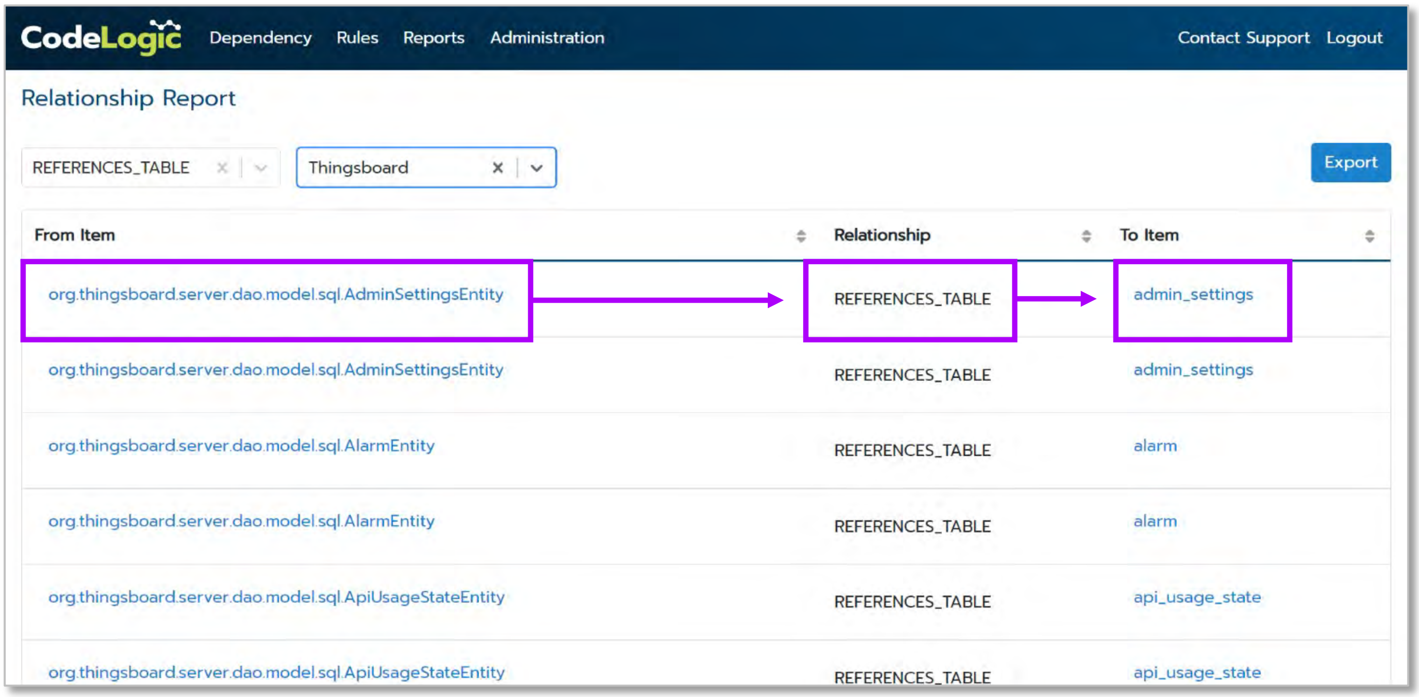
Applications	Database References
⊞ Northwind	28
⊟ Thingsboard	502
⊟ boolean debugMode	4
REFERENCES_COLUMN	4
⊟ boolean enabled	2
REFERENCES_COLUMN	2
⊟ boolean isDefault	4
REFERENCES_COLUMN	4
⊟ boolean isolatedTbCore	2
REFERENCES_COLUMN	2
⊟ boolean isolatedTbRuleEngine	2
REFERENCES_COLUMN	2
⊟ boolean root	2
REFERENCES_COLUMN	2
⊟ com.fasterxml.jackson.databind.JsonNode actionData	2
REFERENCES_COLUMN	2
⊟ com.fasterxml.jackson.databind.JsonNode additionalInfo	16
REFERENCES_COLUMN	16

CodeLogic Answer

CodeLogic tracks relationships between code entities and the database. The Relationship Report makes it easy to export this detail as a CSV and generate a simple pivot table on Application name.

Question 3

Where does the application connect to the database?



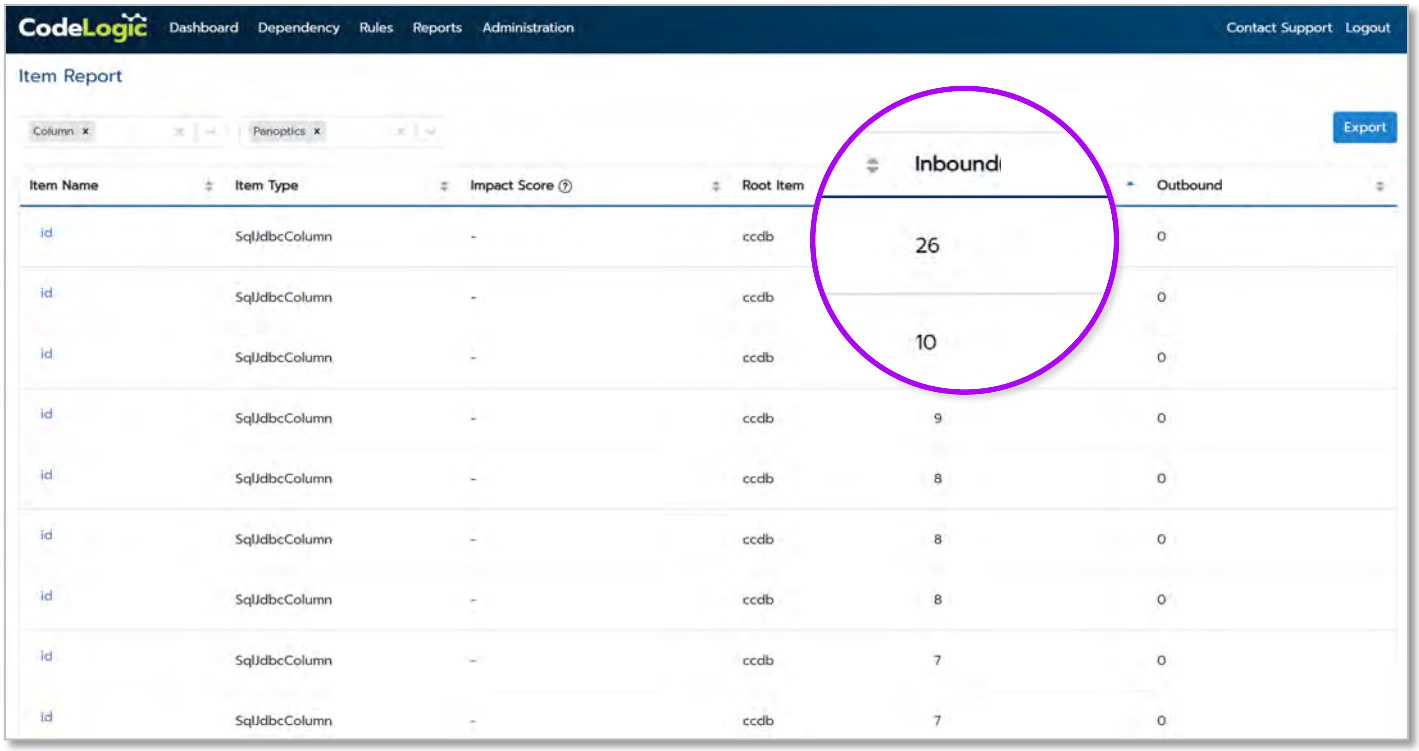
From Item	Relationship	To Item
org.thingsboard.server.dao.model.sql.AdminSettingsEntity	REFERENCES_TABLE	admin_settings
org.thingsboard.server.dao.model.sql.AdminSettingsEntity	REFERENCES_TABLE	admin_settings
org.thingsboard.server.dao.model.sql.AlarmEntity	REFERENCES_TABLE	alarm
org.thingsboard.server.dao.model.sql.AlarmEntity	REFERENCES_TABLE	alarm
org.thingsboard.server.dao.model.sql.ApiUsageStateEntity	REFERENCES_TABLE	api_usage_state
org.thingsboard.server.dao.model.sql.ApiUsageStateEntity	REFERENCES_TABLE	api_usage_state

CodeLogic Answer

The CodeLogic Relationship Report makes it easy to search for code-to-database relationships. In this example the AdminSettingsEntity class maps to the admin_settings table.

Question 4

Which database tables and columns have the most connections to my application?



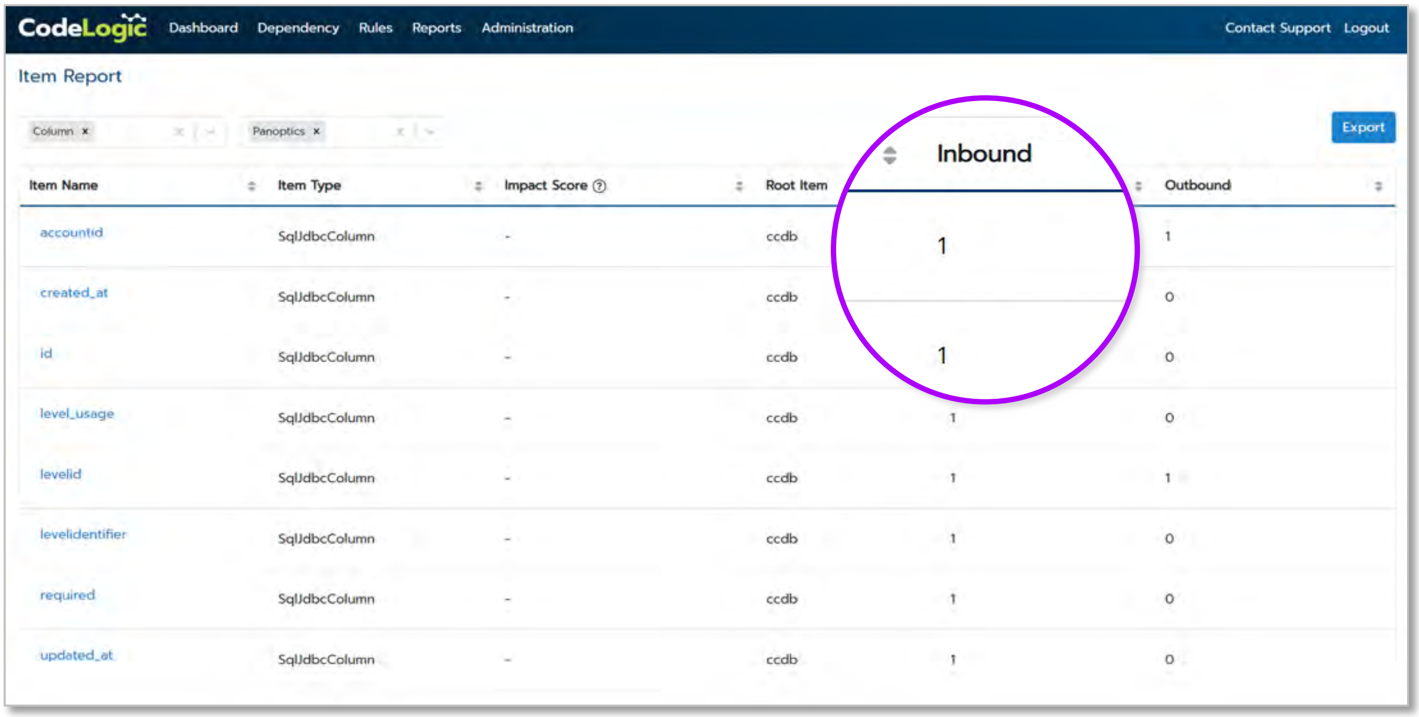
Item Name	Item Type	Impact Score	Root Item	Inbound	Outbound
id	SqlJdbcColumn	-	ccdb	26	0
id	SqlJdbcColumn	-	ccdb	10	0
id	SqlJdbcColumn	-	ccdb		0
id	SqlJdbcColumn	-	ccdb	9	0
id	SqlJdbcColumn	-	ccdb	8	0
id	SqlJdbcColumn	-	ccdb	8	0
id	SqlJdbcColumn	-	ccdb	8	0
id	SqlJdbcColumn	-	ccdb	7	0
id	SqlJdbcColumn	-	ccdb	7	0

CodeLogic Answer

The CodeLogic Item Report allows you to identify which columns have the most inbound relationships, or references. This indicates frequent references from the code, or other parts of the database such as foreign keys.

Question 5

Does my application have any unused tables or columns?



Item Report

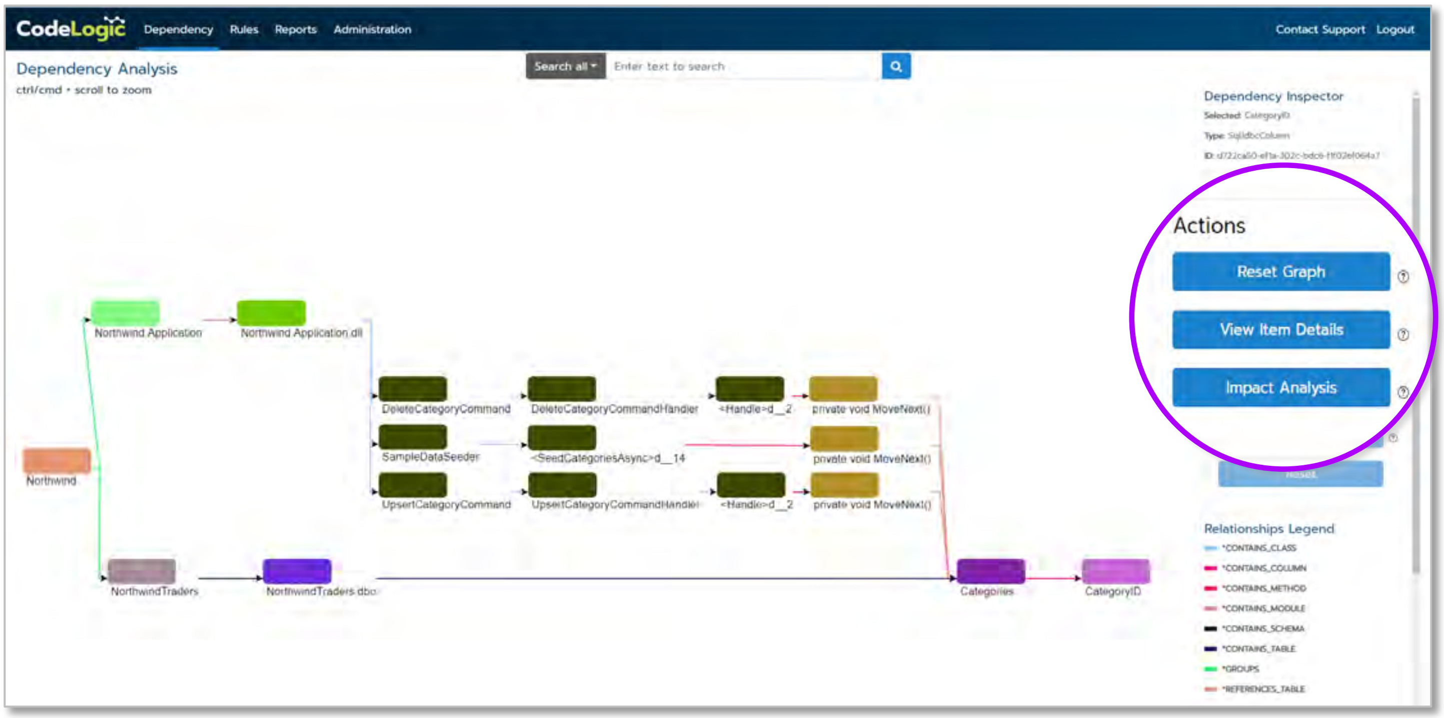
Item Name	Item Type	Impact Score	Root Item	Inbound	Outbound
accountid	SqlJdbcColumn	-	ccdb	1	1
created_at	SqlJdbcColumn	-	ccdb		0
id	SqlJdbcColumn	-	ccdb	1	0
level_usage	SqlJdbcColumn	-	ccdb	1	0
levelid	SqlJdbcColumn	-	ccdb	1	1
levelidentifier	SqlJdbcColumn	-	ccdb	1	0
required	SqlJdbcColumn	-	ccdb	1	0
updated_at	SqlJdbcColumn	-	ccdb	1	0

CodeLogic Answer

The Item Report can be sorted to highlight a low number of Inbound relationships. A single relationship suggests low use. One Inbound relationship will always be present to indicate that a table is part of a schema or that a column is part of a table. The lack of additional relationships conveys that the scan was either incomplete, or that the data is orphaned.

Question 6

What parts of my application are impacted if there is a data type change in the new database platform?



CodeLogic Answer

CodeLogic's Impact Analysis highlights code that depends on the columns being impacted, even across multiple binaries and applications. If changes cascade, suggesting updates to APIs, CodeLogic helps teams follow the threads to which methods in other applications are calling those endpoints.