SuiteView Data Manager

# Connections Screen

## Overview

The Connections screen serves as your gateway to all organizational data sources, providing a centralized hub for managing and accessing databases across your enterprise. This is where users establish secure connections to diverse data systems and select which tables they want to work with.

## What It Can Do

**Universal Data Source Support**

The Connections screen accommodates virtually any data source in your organization, including SQL Server databases, IBM DB2 mainframes, Microsoft Access files, Microsoft Excel files, legacy systems accessible through ODBC, and even flat files like CSV or fixed-width text files. This vendor-agnostic approach eliminates the need for multiple tools.

**Secure Connection Management**

Users can add new connections, test them for validity, and manage existing ones through an intuitive interface. All connection credentials are encrypted using Windows Data Protection API (DPAPI), ensuring sensitive information remains secure while stored locally on the user's machine.

**Real-Time Schema Discovery**

Once connected, the system automatically discovers and displays all available tables, their structure, column definitions, data types, and relationships. This metadata is cached locally for fast access, yet can be refreshed on-demand to capture recent schema changes.

**Selective Table Access**

Rather than overwhelming users with hundreds of tables, the Connections screen allows selective table management. Users check boxes next to tables they frequently use, which are then saved to their My Data workspace for quick access in query building.

## How They Would Use It

**Initial Setup**

Business users begin by clicking Add New Connection and providing basic connection details: server name, database name, and authentication method. The system tests the connection immediately to verify access.

**Browsing Data**

The left panel displays a hierarchical tree of all configured connections. Clicking on a database like ProductionDB reveals its tables in the middle panel. Selecting any table displays its complete schema information in the right panel, including column names, data types, and constraints.

**Curating Workspace**

As users browse, they check boxes next to relevant tables. For instance, a sales analyst might select Customers, Orders, and OrderDetails. These selections automatically populate their My Data screen, creating a personalized workspace with only the tables they need.

## What It Looks Like

The interface is divided into three panels. The left panel (narrow, 200px) shows a connection tree with expandable nodes for each data source, using familiar folder-style icons. The middle panel displays a searchable, checkbox-enabled table list. The right panel provides detailed schema information including table statistics (row counts, last modified dates) and complete column listings with data types.

Action buttons appear at strategic locations: Add New Connection and Manage Connections buttons at the bottom of the connections panel, and Test Connection, Refresh Metadata, and Edit Connection buttons in the toolbar above the schema details.

A screenshot of a computer

AI-generated content may be incorrect.

# My Data Screen

## Overview

The My Data screen serves as a personalized workspace where users access their curated collection of frequently-used tables and saved queries. Unlike the Connections screen which displays everything available, My Data shows only the data sources and queries users have explicitly selected for their daily work, creating an efficient, clutter-free environment for query building and data analysis.

## What It Can Do

**Organized Data Access**

My Data organizes saved items into three distinct categories: 1) My Connections, which have the saved Tables/Views from the Connections page, 2) DB Queries for single-database queries, and 3) XDB Queries for cross-database queries. This three-tier structure reflects the complexity hierarchy, making it easy to understand what type of data operation you're working with at a glance.

**Deep Schema Exploration**

When selecting a saved table, users can explore its complete structure including column names, data types, keys, and constraints. The Find Unique Values feature allows users to analyze specific columns on-demand, retrieving and displaying all distinct values. This capability helps users understand data distributions, identify lookup values for filtering, and validate data quality before building queries.

**Query Management and Execution**

Saved queries display comprehensive metadata including when they were last saved, when they were last executed, how long execution took, and how many records were returned. Users can review the query structure showing which database it targets, which fields are displayed, which fields are used in selection criteria, and what filters are applied. The prominent Run button executes the query immediately, opening results in a separate optimized data grid window.

**Cross-Database Query Visibility**

For cross-database queries, My Data provides transparent visibility into the complexity. The interface clearly identifies all databases involved, lists every table and query being accessed with full database qualification, shows which fields are being selected from which sources, and displays the complete filter criteria. This transparency helps users understand the backend processing required to execute multi-source queries.

## How They Would Use It

**Building a Workspace**

Users populate My Data by selecting tables in the Connections screen. As they check boxes next to frequently-used tables, these selections automatically appear in the Saved Tables category. Over time, as they create and save queries using the DB Query and XDB Query builders, those queries appear in their respective categories, building a personalized library of data tools.

**Analyzing Table Data**

When exploring a saved table, users click Find Unique to select columns for analysis, then click Find Unique Values. The system queries each selected column and displays results showing the count of unique values and a scrollable list of the actual values. For columns with many distinct values, only the first 50 display with a total count indicator. Users can see when this analysis was last performed through the Last Updated timestamp, helping them decide whether a refresh is needed.

**Reviewing and Running Queries**

Before executing a saved query, users can review its complete definition. They see which database it uses, which fields will appear in results, which fields participate in the selection logic, and what filter conditions apply. The metadata section shows performance history, helping users gauge expected execution time. When ready, clicking Run executes the query and displays results in a separate window with full filtering and sorting capabilities.

In addition to running or reviewing a saved query, users can edit the query by clicking the Edit button. This action takes the user directly to the Query building screen, where they can modify the query structure, adjust selection criteria, and update filters to refine results before saving changes.

**Managing Complexity**

For cross-database queries, users benefit from the detailed breakdown showing exactly which databases and tables contribute to the results. This visibility is crucial for understanding why a query might run slower than single-database queries, as it reveals the complexity of data integration happening behind the scenes. The hierarchical display helps users verify that all necessary data sources are included and properly qualified.

## What It Looks Like

The interface uses a two-panel layout. The left panel contains a hierarchical tree view organized into three collapsible sections: My Connections, DB Queries, and XDB Queries. My Connections displays a familiar folder structure showing server, database, and table names. The DB Queries group items by the database they use. The XDB Queries group items by user-defined categories such as Reinsurance or Default.

The right panel adapts based on selection type. For My Connection tables/views, it displays a data grid showing column schema with checkboxes for unique value analysis, data type information, and a scrollable Unique Values column that populates on demand. For queries, it transitions to a structured form layout showing metadata at top, followed by clearly labeled sections for database information, display fields, selection fields, and filter criteria. The Run button appears prominently at the top for immediate query execution.

Visual cues distinguish complexity levels. Single-database queries use standard formatting with simple field lists. Cross-database queries employ additional visual indicators including database name prefixes in monospace font, color-coded tags for different databases, and hierarchical indentation showing the relationship between databases, tables, and fields. This progressive disclosure ensures simple queries remain simple to view while complex queries reveal their full structure.

Action buttons appear contextually. The Find Unique Values button activates when working with tables. The Run button displays when viewing any saved query. Right clicking a tree item reveils a short context menu that gives the user option to delete/remove the dataset from My Data. The interface maintains consistent spacing and typography throughout, using a clean data-dense style that maximizes information visibility without overwhelming the user.

A screenshot of a data manager

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# DB Query Screen

## Overview

The DB Query screen provides a visual, form-based interface for building queries against a single database. Designed for business users who need structured data access without writing SQL, this screen guides users through query construction using drag-and-drop interactions and intelligent, type-aware filter controls. The interface enforces single-database operations, ensuring straightforward query execution and optimal performance while maintaining a low learning curve for non-technical users.

## What It Can Do

**Intelligent Filter Building**

The Criteria tab adapts dynamically to the data type of each field dragged into it. For string fields, users see a dropdown offering match options including exact match, starts with, ends with, and contains, paired with a text input box. Numeric fields present an exact value checkbox along with low and high range inputs for boundary-based filtering. Date fields provide date pickers for exact dates or date ranges. When a field has a limited set of unique values already discovered, a checkbox list appears showing those values with special options for selecting none or all values at once, enabling rapid multi-value filtering without typing.

**Visual Column Selection**

The Display tab provides a simple stacking interface where users drag fields to define their result set columns. Fields appear in the order they are added, showing the fully qualified table and field name. Users can remove fields by clicking the remove icon, allowing iterative refinement of the output column list. This visual approach eliminates the need to remember column names or construct SELECT lists manually.

**Comprehensive JOIN Management**

The Tables tab automatically tracks every table referenced in criteria or display fields, maintaining a running list of involved tables. Users establish the primary table using the FROM clause dropdown. The Add Join button creates new join blocks where users specify the join type through a dropdown offering INNER, LEFT OUTER, RIGHT OUTER, and FULL OUTER joins. Each join block includes ON condition builders where users select fields from dropdowns to define join relationships. Users can add multiple AND conditions within a single join block and create multiple join blocks to establish complex multi-table relationships, all through dropdown selections rather than SQL syntax.

**Direct Data Source Integration**

DB Query connects directly to the My Data workspace, displaying the user's curated collection of saved tables and previously saved queries. Users can drag fields from raw tables or from saved queries, treating queries as reusable building blocks. The interface enforces the single-database constraint by only displaying items from the selected datasource, preventing accidental cross-database query attempts while keeping the experience straightforward.

## How They Would Use It

**Setting Up the Query Context**

Users begin by clicking DB Query in the ribbon. This opens the right panel to a blank query (to edit an existing query see “Edit Existing Query”). The user may begin building a new query. The user begins by selecting a datasource from the My Data panel on the left, which displays tables and saved queries organized by database. Selecting a table populates the Fields panel with all available columns, showing data types for reference. Users can double-click any field to expand its unique values inline, helping them understand the data distribution before building filters.

**Building Filter Criteria**

On the Criteria tab, users drag fields from the Fields panel into the filter area. For example, dragging a State field creates a filter control showing a match type dropdown and a text box, plus a checkbox list of all states if unique values were previously discovered. Dragging an OrderDate field creates date range pickers. Dragging a CustomerTypeID field creates numeric range inputs with an exact match option. Each filter control stacks vertically, allowing users to combine multiple criteria for precise data selection.

**Selecting Output Columns**

Switching to the Display tab, users drag the fields they want to see in results. For a customer orders report, they might drag CustomerName, OrderDate, and Total. The interface shows the fully qualified field names like Customers.Name and Orders.Total, maintaining clarity about data sources. Users can rearrange display fields by removing and re-adding them in the desired order.

**Configuring Table Relationships**

On the Tables tab, users see a list of all tables involved based on the fields they've dragged. They select the primary table from the FROM dropdown, then click Add Join to create relationships. For each join, they choose the join type, select the table to join, and configure ON conditions by selecting matching fields from dropdowns. Additional AND conditions can be added within a join to handle composite keys or complex relationships.

**Executing and Saving**

With criteria, display fields, and joins configured, users click Run Query to execute. Results appear in a separate optimized data grid window with filtering and sorting capabilities. The Save Query button stores the complete query definition in My Data under the DB Queries category, making it available for future use or as a component in cross-database queries. The system tracks metadata including last run time, execution duration, and record count for performance monitoring.

**Edit Existing Query**

To edit an existing query, find the user find the query they want in the list and double clicks it. The right panel will now be populate with the query. The user can make edits and then click Save Query when they are ready.

## What It Looks Like

The interface divides into four main areas. Three narrow panels on the left show a hierarchical My Data Source tree, a filtered list of tables and queries from the selected datasource, and a detailed Fields panel listing all columns with data types displayed in monospace font. The main panel on the right contains action buttons for Run Query and Save Query, followed by a three-tab interface.

The Criteria tab presents a large drag-and-drop zone that transforms into a vertical stack of filter controls as fields are added. Each control shows the table and field name as a header with a remove button, followed by data type-appropriate input controls. String filter controls display horizontally with the match type dropdown on the left and text input taking most of the space. Numeric and date controls stack their components vertically with clear labels. Unique value checkbox lists appear to the right of the main filter controls when available, bordered and scrollable for long lists.

The Display tab shows a simpler interface with stacked field items, each displaying the fully qualified field name with a remove button aligned to the right. The vertical stacking makes the output column order immediately obvious. The Tables tab combines informational displays with interactive builders. At the top, a list of involved tables appears as colored tag-style badges. Below that, the FROM clause section shows a labeled dropdown. Join blocks appear as bordered containers, each containing a row of dropdowns for join type and table selection, followed by indented ON condition rows with field selection dropdowns connected by equals signs. Add buttons allow expanding both the ON conditions within a block and the number of join blocks overall.

Visual feedback appears throughout the interface. Drag-and-drop zones show dashed borders that highlight when fields hover over them. Active filter controls use a light background color to distinguish them from empty states. The status bar at the bottom continuously updates showing the count of criteria, display fields, and tables involved, giving users constant awareness of their query complexity. The single-database constraint remains visible through the datasource label at the top, clearly indicating which database all operations target.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# XDB Query Screen

## Overview

The XDB Query screen extends the DB Query functionality to support cross-database operations, enabling users to build queries that span multiple databases or database types. While maintaining the same intuitive drag-and-drop interface as DB Query, XDB Query adds critical visibility into data source complexity, shows database qualification throughout, and manages the additional constraints required for successful cross-database execution.

## What It Can Do

**Cross-Database Query Construction**

XDB Query breaks the single-database limitation, allowing users to combine data from SQL Server, DB2, Oracle, and other sources in a single query. The system handles the complex backend coordination required to execute these distributed queries, while presenting users with the same familiar drag-and-drop interface they know from DB Query.

**Intelligent Data Source Management**

The screen provides enhanced visibility into which databases contribute to each query component. Every table, field, and saved query displays its source database through visual badges and database qualification (database.table.field notation). This transparency helps users understand query complexity and make informed decisions about performance tradeoffs.

**Application-Level Join Management**

When queries span multiple databases, XDB Query automatically manages application-level joins. The system retrieves data from each source database independently, then performs join operations in the application layer. Users receive clear warnings about these operations, as they execute differently than native database joins and may have performance implications for large datasets.

**Saved Query Integration**

XDB Query treats previously saved DB Queries as queryable data sources. Users can drag saved queries into their work just like regular tables, and the system expands them to show the fields they provide. This enables powerful query composition, where complex queries become reusable building blocks for even more sophisticated analysis. The only My Connection tables/view or DB Queres can bet used in a XDB Query. Other XDB queries not yet support composition of other XDB Queries.

## How to Use It

**Building Cross-Database Queries**

The workflow mirrors DB Query: build a new query and save it or find an existing query and double click to edit it. The key difference is database awareness— the query builder will now allow the user to add tables or queries from different datasources or data bases, and the interface highlights when operations will require application-level processing rather than native database execution (which occurs when more than one database is required in the query definition.

**Working with the Databases Tab**

XDB Query introduces a new Databases tab that replaces DB Query's simpler Tables tab. This tab provides comprehensive visibility into all databases involved in the current query. The "Databases Involved" section displays badges for each source, color-coded by database type (teal for SQL Server, yellow for saved queries, etc.). The "Queries Involved" section lists any saved queries being used as data sources, showing their names and database origins.

The "Tables Involved" section shows regular tables, while the FROM and JOIN configuration areas handle query structure. When users configure joins between tables from different databases, yellow warning banners appear automatically, explaining that these joins will execute at the application level and may perform differently than single-database joins.

**Understanding Query Execution**

XDB Query provides clear feedback about query execution strategies. Single-database queries execute exactly as they would in DB Query, with full database optimization. Cross-database queries trigger visual indicators showing which operations require application-level processing. The status bar includes database counts and execution mode information, helping users understand the scope and nature of their queries before execution.

## What It Looks Like

**Enhanced Visual Hierarchy**

The XDB Query interface maintains DB Query's clean, modern design while. Each database node displays a colored status indicator matching its badge color scheme—teal for SQL Server connections, yellow for saved queries, and other colors for different database types.

Field listings throughout the interface show database qualification in a monospace font (database.table.field), making the technical structure readable while distinguishing it from regular labels. These qualified names appear consistently in the Fields panel, in criteria controls, in display field listings, and in all dropdown selections, ensuring users always know which database they're working with.

**The Databases Tab**

The Databases tab presents information in distinct sections with clear visual separation. The "Databases Involved" area at the top displays colored badges for each source database, creating an at-a-glance inventory of query complexity. Teal badges mark involved databases, yellow styling highlights queries used as data sources, and standard text shows regular tables. Each entry displays its database qualification in monospace font, reinforcing the technical nature of cross-database operations.

The Query Display Fields containers expand when saved queries are involved, showing a nested list of fields that those queries provide, giving users complete visibility into the data structure. This expandable design prevents information overload while making detailed field information available when needed.

**Visual Feedback**

Despite the added complexity, the interface maintains the same clean, approachable design as DB Query. The drag-and-drop zones, tab structure, and control layouts remain familiar, ensuring that users comfortable with single-database queries can transition smoothly to cross-database work. The additional database information layers onto the existing interface without overwhelming the core workflow, striking a careful balance between power and usability.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Additional items:  
PLease add a button to the right of Connectoins (see screen shot where i circled in red, thats where i want and Add button that will bring up a Connection Dialogue. And this is where i need your help - i need you to figuare out everythign we need for a connection dialogue. Specifcally we need to allow the user to have the program scan there local ODBC and add connections that way. Also, the user should be able to add a connection MS Access database, Excel sheets and structured stables, CSV files, Fixed Width flat files (with the ability to import or paste in all the column starting points and lengths). Also, the user should have the abliity to double click any existing connection and have an edit dialogue appropreiate for taht type of connection

Come up with something with a regal royal blue and gold theme in various shades. And give me a little more dimensionality, i dont like everythig so flat,

Look and feel of a information dense datatool:

This design maximizes information density while maintaining readability. It follows the **"data tool" paradigm** (like SQL Server Management Studio) rather than a consumer app - professional, efficient, no wasted space!