



## **Embarcadero® DB Optimizer™ 1.0 Evaluation Guide**

Published: July 14, 2008

Embarcadero Technologies, Inc.  
100 California Street, 12th Floor  
San Francisco, CA 94111 U.S.A.

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Embarcadero Technologies, Inc. on the issues discussed as of the date of publication. Because Embarcadero must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Embarcadero, and Embarcadero cannot guarantee the accuracy of any information presented after the date of publication.

This reviewers guide is for informational purposes only. EMBARCADERO MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Embarcadero Technologies, Inc..

Embarcadero may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Embarcadero, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2008 Embarcadero Technologies, Inc. All rights reserved.

Embarcadero Technologies, Inc., and all of the company's products are either registered trademarks or trademarks of Embarcadero Technologies, Inc. in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

# Contents

<b>INTRODUCTION TO DB OPTIMIZER .....</b>	<b>5</b>
Product Benefits .....	5
<b>ABOUT THIS EVALUATION GUIDE.....</b>	<b>6</b>
<b>SESSION 1: GETTING STARTED WITH DB OPTIMIZER .....</b>	<b>7</b>
Install DB Optimizer.....	7
Start DB Optimizer: .....	7
User Interface Overview.....	7
<b>SESSION 2: WORKING WITH DATA SOURCE EXPLORER.....</b>	<b>9</b>
Adding Data Sources .....	10
Browsing Data Sources.....	11
<b>SESSION 3: WORKING WITH SQL TUNER .....</b>	<b>13</b>
Creating and running a new tuning session .....	13
Adding tuning candidates .....	13
Viewing the statements and cases extracted from the tuning candidates .....	14
Experimenting on your own.....	15
Providing bind variable values.....	15
Executing a statement to gather additional statistics.....	15
Opening an explain plan on a statement.....	15
<b>SESSION 4: RUNNING A PROFILING SESSION .....</b>	<b>16</b>
Creating a launch configuration.....	17
Running a profiling session using a launch configuration.....	17
Viewing session results in the Profiling editor .....	18
Experimenting on your own.....	19
Investigating the event category tabs .....	19
Zooming in on the graph .....	19
Displaying detailed statistics for a subset of the graph.....	20
Drilling down into statements and events .....	20
Saving, closing, and opening profiling archives.....	20
<b>SESSION 5: WORKING WITH SQL CODE ASSIST .....</b>	<b>21</b>
Code Extraction.....	23
Code Highlighting.....	23
Automatic Error Detection .....	25

Code Complete .....	26
Hyperlinks .....	26
Code Formatting .....	26
Code Folding.....	27
<b>SESSION 6: EXECUTING SQL CODE .....</b>	<b>28</b>
Configuring SQL Execution Parameters.....	29
<b>SESSION 7: WORKING WITH SQL PROJECT INSIGHT .....</b>	<b>30</b>
<b>SESSION 8: WORKING WITH SQL CODE ASSIST .....</b>	<b>33</b>
Code Extraction.....	35
Code Highlighting.....	35
Automatic Error Detection .....	37
Code Complete .....	38
Hyperlinks .....	38
Code Formatting .....	38
Code Folding.....	39
<b>SESSION 9: EXECUTING SQL CODE .....</b>	<b>40</b>
Configuring SQL Execution Parameters.....	41
<b>ADDITIONAL RESOURCES.....</b>	<b>42</b>
Licensing Your Embarcadero Technologies Product.....	42
Embarcadero Technologies Product Support.....	42
Embarcadero Technologies Technical Support.....	42
Embarcadero Technologies on the Web .....	42

## Introduction to DB Optimizer

Embarcadero® DB Optimizer™ simplifies SQL optimization and development for application developers with many features for improving productivity and reducing errors. A rich SQL IDE with statement tuning, data source profiling, code completion, real-time error checking, code formatting and sophisticated object validation tools help streamline coding tasks. DB Optimizer's user interface helps improve overall productivity with integrated development, monitoring, and tuning components. DB Optimizer offers native support for IBM® DB2® for LUW, Oracle®, Microsoft® SQL Server and Sybase® as well as JDBC support for other DBMS .

### Product Benefits

- Running profiles against data sources to locate and diagnose wait-based bottlenecks and inefficiencies.
- Tuning multiple SQL statements using explain plan costing, execution statistics, and transformation-based and SQL Optimizer hint-based suggestions
- SQL Project Insight provides project-level SQL file cataloging and search features to streamline project organization and maintenance.
- SQL Code Assist ensures 100% object name accuracy without the requirement to connect to a database, plus real-time SQL syntax validation.
- SQL Project Insight provides project-level SQL file cataloging and search features to streamline project organization and maintenance.
- The Migration Wizard imports data sources from Eclipse DTP (Data Tools Project) or Quest TOAD.
- Data Source Explorer enables you to navigate, search, extract DDL, execute commands, and browse an outline, without having to open the SQL file.
- Formatting Profiles ensure consistent, quality code layout for easy review and extension. Profiles can be customized and shared.

## About This Evaluation Guide

This evaluation guide enables you to get started with DB Optimizer.

Its purpose is to provide you with the foundation needed to fully utilize the features and benefits of the products, and apply them to your own enterprise. This guide contains information on how to get the application up and running, and how to browse data sources, and create, modify, execute, and tune SQL code for the purpose of database development.

This guide is divided into five sections:

- Session 1: Getting Started with DB Optimizer**
- Session 2: Working with Data Source Explorer**
- Session 3: Working with SQL Tuner**
- Session 4: Working with SQL Profiler**
- Session 5: Working with SQL Code Assist**
- Session 6: Executing SQL Code**
- Session 7: Working with SQL Project Insight**
- Session 8: Working with SQL Code Assist**
- Session 9: Executing SQL Code**

Once you have started, you can select **Help** from the Menu Bar to find additional resources that complement and build upon the features and tasks presented in this guide.

## Session 1: Getting Started with DB Optimizer

### Install DB Optimizer

You can obtain the latest version of the DB Optimizer software from the Embarcadero website at <http://www.embarcadero.com/downloads/downloaddboptimizer.jsp>.

- Launch the Installer and follow the instructions in the Wizard to complete it.

You may be required to restart your machine to complete the installation of DB Optimizer. It is also recommended that you copy the installer to the desktop of your machine prior to running the Installer.

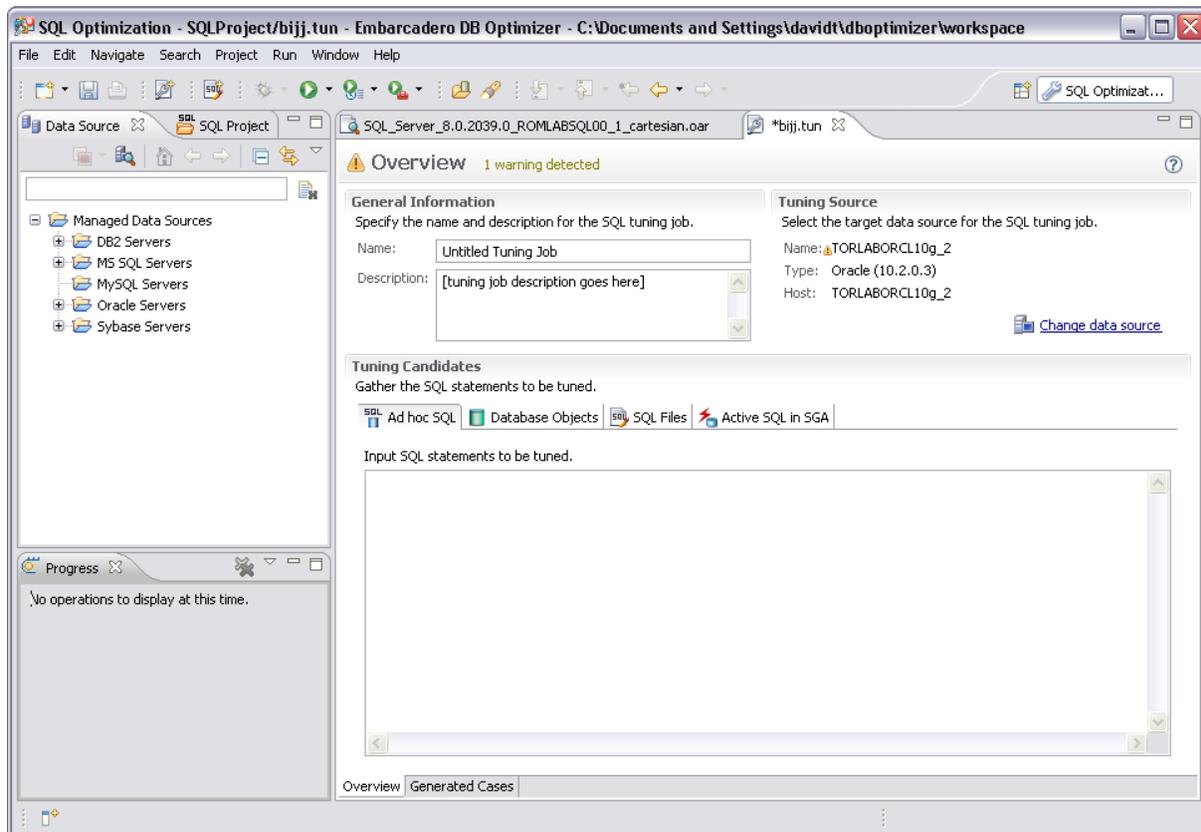
**Note:** DB Optimizer can be installed with a 14-day trial license, after which you must purchase a permanent license to continue using the product.

### Start DB Optimizer:

- From the Desktop, choose the Windows Start menu and then select **Programs > Embarcadero DB Optimizer 1.0 > DB Optimizer**

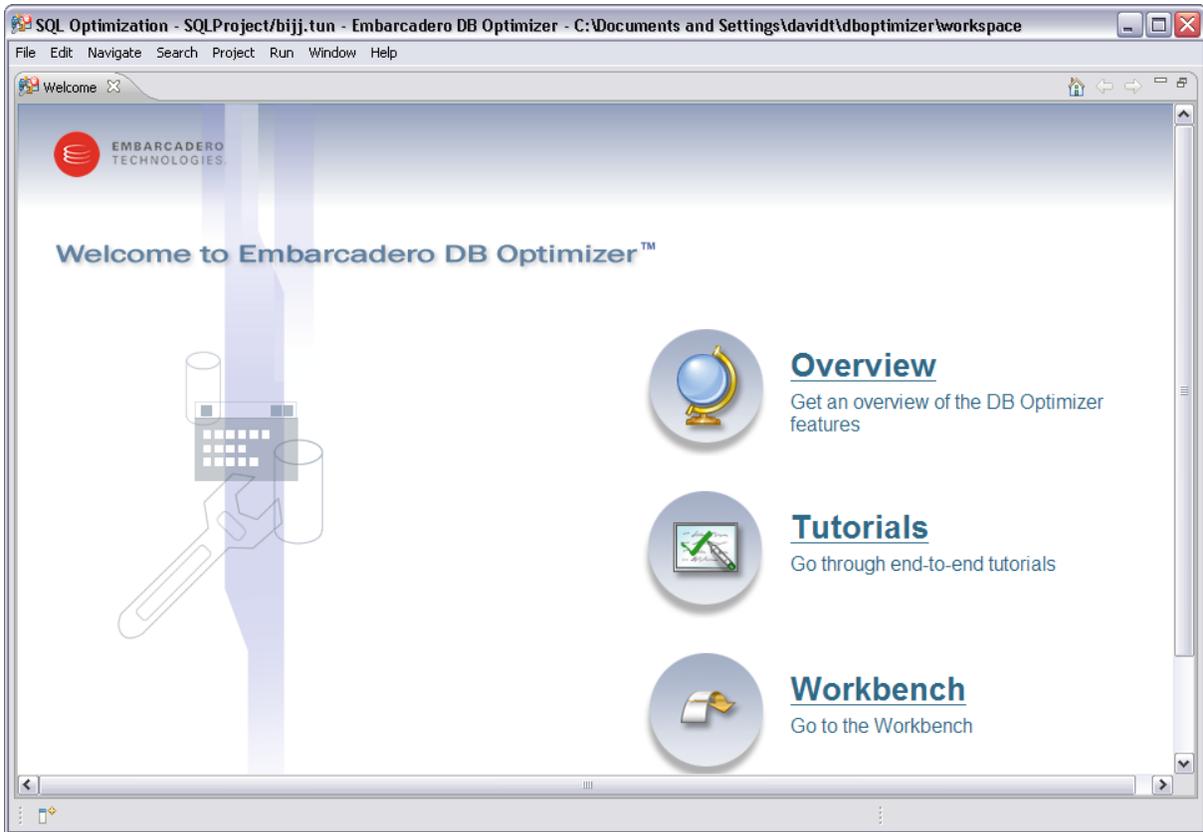
### User Interface Overview

The DB Optimizer application environment is known as the **Workbench**. The Workbench is composed of common interface elements that provide tools to accomplish the creation, tuning, management, and execution of SQL code. These objects provide a uniform system for working with jobs and data sources.



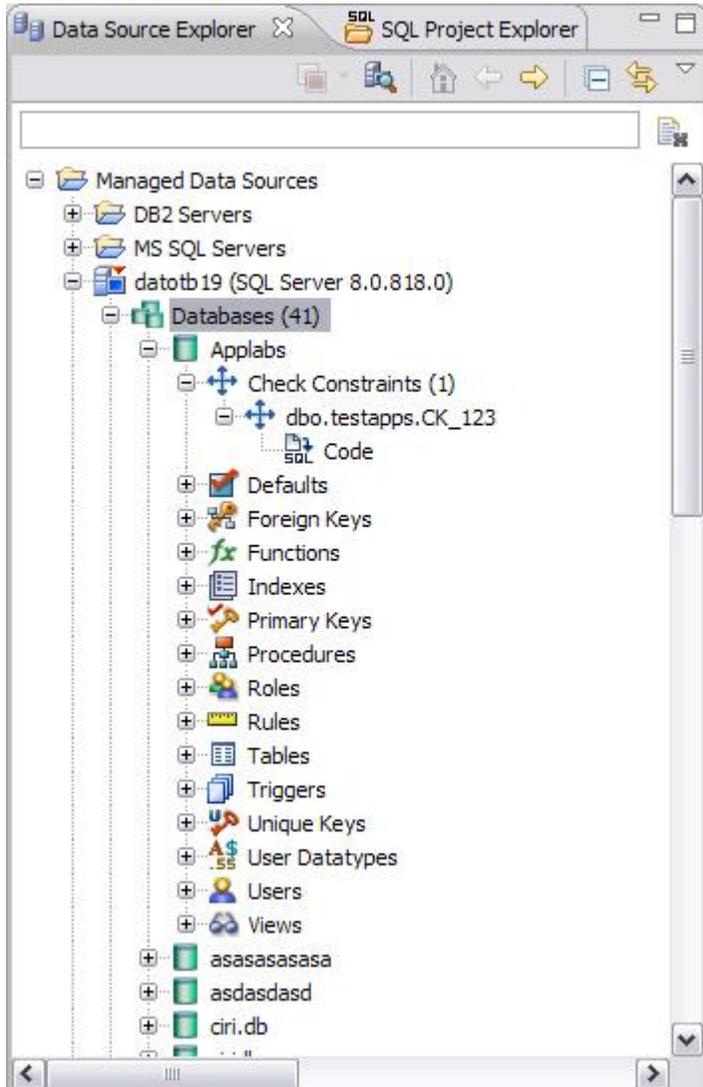
The Workbench provides an environment to tune, define, manage, and execute SQL code.

- The **Welcome Page** is the first screen you will see when you initially launch the application. This screen provides links to documentation and other help files/tutorials to further familiarize you with the product. It provides easy access to information that may be of value to new users.



## Session 2: Working with Data Source Explorer

Data Source Explorer provides a tree view of all data sources registered in DB Optimizer. In addition, it breaks down the components of each data source and categorizes databases and corresponding database objects by object type and underlying SQL code. This feature provides you with a view of the contents of data sources in your enterprise in an easily navigated and cataloged interface.



*SQL Project Insight sorts databases and database objects by category within Data Source Explorer.*

If data sources are particularly large or complex, or you are only developing specific objects, you can apply database object filters on the view in Data Source Explorer.

## Adding Data Sources

In order to view and edit data source objects, you need to register data sources to be used as sources in the environment by providing connection information and other details to DB Optimizer.

Data sources are registered and managed in **Data Source Explorer**. Each time you register a new data source you need to specify its connection information and organize it in the View, as needed. Once a data source has been registered, it remains stored in a catalog and does not need to be re-registered each time you open DB Optimizer. It can be used in multiple jobs, archived, or otherwise “shared” with regards to the general functionality of the application.



*The Data Source Explorer View provides an organizational tree of registered data sources and the comparison jobs and archives associated with them.*

### To add a data source:

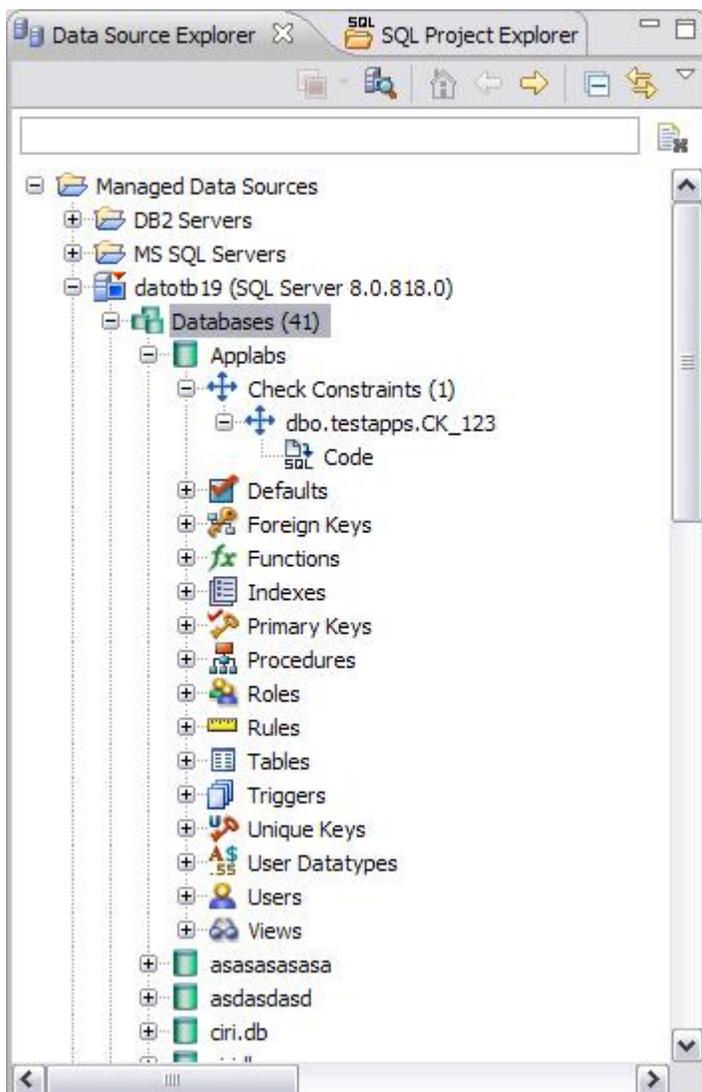
Highlight Data Source Explorer and choose **File > New > Data Source** from the Menu Bar. The Data Source Wizard appears.

NOTE: If you are going to use the SQL Tuner feature, introduced later in the guide, you must register an Oracle data source.

## Browsing Data Sources

In order to develop and maintain SQL code on data sources within your enterprise, you need to access database objects within the application environment. It is important to be able to view databases, database objects, and the underlying code in an organized manner, especially when maintaining a large system.

You use Data Source Explorer's tree structure to view and access databases, tables, and other objects on data sources, by expanding the nodes of each registered data source. When you create new SQL files, you can execute them on a specified data source, and these new objects automatically appear in the tree.



The Data Source Explorer tree provides a list of databases, database objects, and underlying code stored per individual registered data source.

All objects listed in Data Source Explorer are organized, in descending order, by data source, by database or login, by database object, by object type, by underlying SQL code. If you double-click an object, DB Optimizer automatically extracts the code and places it in an editor.

**To browse a data source:**

Expand the node of the data source whose databases, database objects and SQL code you want to view. If you want to view the code of a specified object, double-click the object to open the code in the editor.

## Session 3: Working with SQL Tuner

SQL Tuner lets you view cost details on groups of SQL statements. As a means of identifying bottlenecks or problem queries, SQL Tuner lets you:

- View high-level explain plan costs for statements as well as offering the option to view operation-by-operation explain plan details
- Generate execution statistics such as CPU time as well as Read and Sort details
- View explain plan and execution statistics for automatically-generated SQL Optimizer hint-based variations on all statements in a tuning session

Used as a development aid, SQL Tuner lets you fine tune queries before putting them into production. As a maintenance tool, SQL Tuner can help you isolate and optimize queries in production systems.

### Creating and running a new tuning session

In order to walk through this exercise, you must first register an Oracle data source. If an Oracle data source is not available, you should skip to the next session.

#### To start a new tuning session:

1. Select **File > New > Other**.  
A **New** dialog opens
2. Click **Tuning Job** and then click **Finish**.  
A new, untitled Tuning editor instance opens.
3. In the **Name** box type **My First Tuning Job** and in the **Description** box type a description for your job.  
  
Later, when you save this session, a tuning archive will be created and named according to the value you used in the **Name** field.
4. In the **Tuning Source** area click **Select** data source  
A Data Source viewer opens.
5. Click an Oracle data source and if prompted, confirm that you want to connect.

### Adding tuning candidates

The lower part of the Tuning editor contains the **Tuning Candidates** area. This is where you add the SQL that you want SQL Tuner to analyze. Next you will add some tuning candidate and view the results.

#### To add a single ad hoc statement to your tuning candidates:

1. Click the **Overview** tab and then click **Ad Hoc SQL**.
2. Type a SELECT statement into the **Input SQL statements to be tuned** area. For example:

```
select employee_id, MIN(SALARY)
from hr.employees WHERE employee_id = 50
group by employee_id
having MIN(SALARY) > 1000;
```

The **Database Objects** tab lets you add SQL-containing database objects to your tuning session. As you add objects, SELECT, INSERT, DELETE, and UPDATE statements are extracted and added to your tuning candidates.

**To add SQL extracted from database objects to your tuning candidates:**

1. Click the **Database Objects** tab.
2. In the Data Source Explorer pane, use the expand controls to display individual resources under the following database object types:
  - Functions
  - Materialized Views
  - Package bodies
  - Procedures
  - Stored Outlines
  - Triggers
  - Views
3. Use the expand/collapse icons to show the specific resources in each category and then drag one or two from each of these categories from the Data Source Explorer pane into the **Specify database objects containing SQL ...** area of the Database Objects tab.



Since you will be providing any required bind variables and executing SQL statements from these sources in a later step, you should choose resources that you are familiar with.

You can also add SQL files or scan for active SQL in the Oracle SGA but for now you should have enough tuning candidates.

**Viewing the statements and cases extracted from the tuning candidates**

As you add tuning candidates to a tuning job, SELECT, INSERT, DELETE, and UPDATE statements are extracted from the tuning candidates and added to the **Generated Cases** tab.

**To view the SQL extracted from the various tuning candidates:**

1. Click the **Generated Cases** tab.

For each supported DML statement extracted from the tuning candidates, SQL Tuner creates a numbered statement.

SQL Statements and Cases					Cost	Elapsed Time	Other Execution Statistics	
Name	Text	Source	Index Analysis	Value	Value (s)	Physical Reads	Logical Reads	
Statement 1	select from hr.EMPLOYEES	Ad Hoc	Click to optimize	0.0				
Statement 2	select from CATA.PRODUSE,	Ad Hoc	OK	10.0				
Statement 3	select from CATA.PRODUSE,	Ad Hoc	Click to optimize	4.0				
Statement 4	select from hr.employees	Ad Hoc	OK	4.0				
Statement 5	select from CATA.TBA,	Ad Hoc	Unable to					
Statement 6	select from Faketable	Ad Hoc	OK					

In addition to the numbered statement name, the SQL entry shows an excerpt of the statement and the tuning candidate's source. While an explain plan cost is calculated as you

add tuning candidates, the elapsed time and other execution statistics are not available until you execute SQL statements. In a subsequent step, you will execute some of the statements on the **Generated Cases** tab for the purpose of comparing relative costs.

An expand/collapse icon beside a statement record, indicates that a transformation case or one or more SQL Optimizer hint-based cases have been applied.

2. Use one of the expand/collapse icons to display all case variations for that statement.

SQL Statements and Cases				Cost	Elaps...Time	Other Execution Statistics	
Name	Text	Source	Value	Value (s)	Physical Reads	Logical Reads	
SQL Statement 3	select from hr.employees	Ad Hoc	1.0				
ALL_ROWS			1.0				
FIRS...OWS			1.0				
RULE			1.0				
CHOOSE			1.0				
FULL			2.0				

3. Collapse the hint variations on the statement you expanded.

## Experimenting on your own

The following topics cover additional experiments you can perform to get familiar with the SQL Tuner feature. They are all operations you perform on the **Generated Cases** tab.

### Providing bind variable values

The Tuning Status Indicator for your tuning candidates may show that bind variables must be set before you can execute that statement or case:



If so, right-click in the text box for that statement to open a SQL Viewer, scroll down to the set of controls for setting bind variables, and use the Data Type and Value controls to set bind variables. When finished, use the collapse icon in the text box of the original statement to close the SQL Viewer.

### Executing a statement to gather additional statistics

When a statement is ready to execute, the Tuning Status Indicator shows the following state:



Right-click on a SELECT statement, and select **Execute** from the context menu. When execution completes the Elapsed Time and Other Execution Statistics values are available.

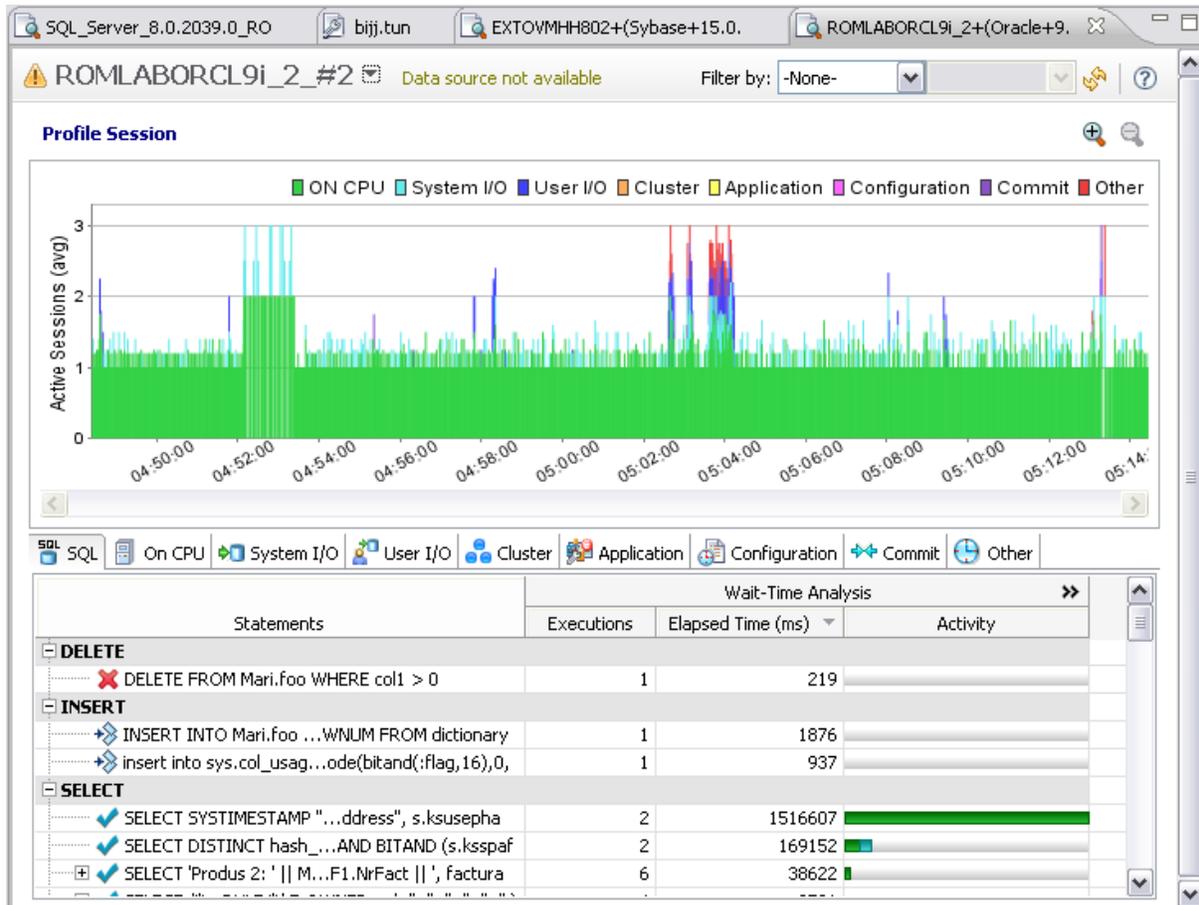
### Opening an explain plan on a statement

You can open an explain plan for any valid SQL statement showing a value **Cost** field. Right-click in the Name field and select **Explain Plan** from the context menu.

Plan Cost		Additional Information				
Operation	Cost	Cardinality	Bytes	CPU Cost	IO Cost	Optimizer
SELECT STATEMENT	1.0	1	8	8461	1	ALL_ROWS
FILTER	0.0					
SORT (GROUP BY NOSORT)	1.0	1	8	8461	1	
TABLE ACCE...EMPLOYEES	1.0	1	8	8461	1	ANALYZED
INDEX (LIN...MP_ID_PK)	0.0	1		1050	0	ANALYZED

## Session 4: Running a profiling session

SQL Profiler lets you locate and diagnose slow-running or otherwise problematic SQL statements by letting you see how they run on a running system. When you run a profiling session, you are taking a fixed-duration snapshot of query activity on a live data source. As the session runs, or optionally when the session completes, results are displayed in a profiling editor.



Results are shown in both graphical and tabular formats. The graphical portion of the editor shows the distribution of sessions executing and sessions waiting to execute over the length of the session, with waiting sessions grouped according to DBMS-specific wait event categories. The tabbed view presents detailed statistics in each event category and for individual SQL statements.

SQL Profiler's key functionality is in its ability to let you drill down to specific problem areas. Selecting a subset of the graph populates the tabbed area with statistics pertaining only to the selected bars of the chart. Similarly, the tabbed view lets you drill down to investigate associations between individual SQL statements and specific wait events.

In this session you will learn how to run a profiling session against a data source. Through this session you will:

- Creating a launch configuration
- Running a profiling session using a launch configuration
- Viewing session results in the Profiling editor
- Experiment with analysis options

## Creating a launch configuration

A DB Optimizer profiling session runs against a particular data source for a specified session length. A launch configuration is an effective shortcut that lets you store the data source and session length profiling parameters. Creating launch configurations for your profiling sessions has the following benefits:

- It lets you launch a profiling session with a single menu command
- It provides a default file name for a profiling archive used to store the results of your session
- As a named resource, it complements DB optimizer project management, letting you use file names as an organizational method (data\_source\_1\_\_30\_sec and data\_source\_1\_\_3\_min, for example).

Your first step in running a profiling session will be to create a pair of launch configurations.

### To create a launch configuration:

6. Ensure that you are connected to a data source.
7. In the Data Source Explorer, right-click the data source, select **Profile As** from the context menu, and then select **Open Profile Dialog**.  
A **Profile** dialog opens.
8. In the **Name** box type **My First Focus Profile\_30 sec**.  
Later, when you save this session, a profiling archive will be created and named according to the value you used in the **Name** field.
9. Ensure that the **Data source** control specifies the data source that you want to profile.
10. Use the **Session Length** controls to set a profile duration of 30 **seconds**.
11. Click **Apply** to create this launch configuration.

It is worth noting that you can use this dialog to run an ad hoc profiling session. Providing **Data source** and **Session Length** details and then clicking the **Profile** button starts a profiling session against that data source and for the specified session length.

Now you are ready to finish up with this dialog by creating a second launch configuration.

### To create a second launch configuration:

1. Use the same method to create a second launch configuration named **My Second Focus Profile\_3 min** with a duration of 3 minutes.
2. After clicking **Apply** to create the second launch configuration, click **Close**.

## Running a profiling session using a launch configuration

Now that you have launch configurations, you can quickly start a profiling session.

### To launch a profiling session using a launch configuration:

1. In the Data Source Explorer, right-click the data source, select **Profile As** from the context menu, and then select **1 Data Source**.  
A **Profiling Session Launch Configurations** dialog opens.
2. Select the **My First Focus Profile\_30 sec** and click **OK**.

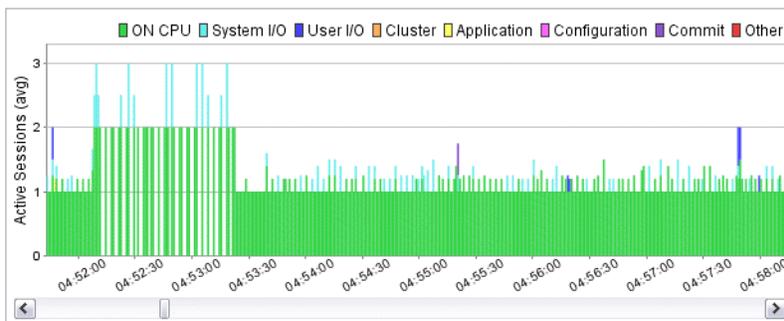
The profile runs for 30 seconds. By default, the Profiling editor opens immediately and is updated as results are gathered. This is referred to as real-time profiling. A **Profile** dialog option lets you choose to open the Profiling editor only after the session completes if that is your preference.

## Viewing session results in the Profiling editor

There are two key parts of the Profiling editor results:

- A bar chart in the upper part of the profiling editor
- A tabbed view in the lower part of the editor

The bar chart in the upper part of the profiling editor shows the distribution of executing versus waiting sessions across the session length, broken down into intervals



The X-axis of the bar chart represents time. DB Optimizer displays the session, broken down into one-second intervals. Within each interval, the Y-axis shows the distribution of sessions executing (green) and those waiting to execute. The color-coding legend in the upper right-hand corner of the chart shows the event categories depicted on the chart. Depending on the DBMS platform you are running a profiling session against, the set of event categories will differ.

The bar chart provides a starting point in diagnosing wait-based bottlenecks. Against a Microsoft SQL Server data source, it could show you a common symptom such as a high instance of lock-based wait events occurring at different points across the session. Similarly, running against an Oracle data source, the bar chart might show an unusually high number of sessions waiting in System IO or User IO events.

In and of themselves, symptoms like these can be caused by load issues, hardware deficiencies, and so on. Since DB Optimizer's primary purpose is to help you tune and optimize queries, the tabbed view gives you the drill-down tools to isolate problem queries. Tuning your queries is a far less expensive approach to performance boosts than infrastructure cash layouts.

The tabbed view provides detailed execution statistics for the individual SQL statements, executing sessions, and sessions waiting to execute during the session

Executed SQL	Executions	CPU (%)	Lock (%)	Memory (%)	Buffer (%)	I/O (%)	Other (%)	Total (%)
<b>INSERT</b>								
insert [#sysperfinfo...,64] option (maxdop 1)	1	2.021	0.000	0.000	0.000	0.000	0.000	2.021
insert [#sysperfinfo...,66] option (maxdop 1)	1	0.390	0.000	0.000	0.000	0.000	0.000	0.390
insert [#sysperfinfo...,64] option (maxdop 1)	1	0.614	0.000	0.000	0.000	0.000	0.000	0.614
insert [#sysperfinfo...,66] option (maxdop 1)	1	2.021	0.000	0.000	0.000	0.000	0.000	2.021
insert [#sysperfinfo...,67] option (maxdop 1)	1	0.614	0.000	0.000	0.000	0.000	0.000	0.614
insert [#sysperfinfo...,75] option (maxdop 1)	1	0.600	0.000	0.000	0.000	0.000	0.000	0.600
insert [#sysperfinfo...,73] option (maxdop 1)	1	0.263	0.000	0.000	0.000	0.000	0.000	0.263
INSERT dbo.Employees..._loyeeID, c.LastName	1	32.216	0.000	0.000	1.083	3.055	10.527	46.881
<b>SELECT</b>								
SELECT * FROM Northwind..Employee	2	0.614	0.000	0.000	0.000	0.380	0.000	0.994
WHERE EmployeeID <= 10	1	0.000	0.000	0.000	0.000	0.380	0.000	0.380
WHERE Northwind..EmployeeID = 1	1	0.614	0.000	0.000	0.000	0.000	0.000	0.614
SELECT * FROM Northwind.dbo.Orders	1	0.000	0.000	0.000	0.000	2.021	0.000	2.021

By default, when you run a profile or open a profiling session archive, the **SQL** tab is displayed. It displays statistics on the DB Optimizer profiling-supported statements that were executing or waiting to execute over the length of the profiling session.

Statements are grouped according to DML statement type: INSERT, SELECT, DELETE, and UPDATE. Within each statement type, statements are displayed in a tree structure with the following statement components:

**Subject** - the DML statement type (and FROM clause where appropriate)

**Predicate** - the WHERE clause

**Remainder** - any statement component following the WHERE clause

For example, all statements with common subjects are shown as a single entry with multiple children; one child for each unique predicate. Predicates are similarly broken down by remainders.

**NOTE:** In addition to the **INSERT**, **SELECT**, **DELETE**, and **UPDATE** statement groupings, the SQL tab can display two other groupings. The **OTHER** grouping includes all recognized SQL statements other than INSERT, SELECT, UPDATE, and DELETE statements. The **UNKNOWN** grouping represents statements that could not be captured.

At a glance the **SQL** tab:

- Can help you recognize statements that may have been a problem in the past
- Can help you quickly isolate important or troublesome queries using the graphical representation of relative time spent executing or waiting

At a more detailed level, for each statement or statement component you can view:

- The number of executions
- Time spent executing
- Time spent waiting to execute in each wait event category

Similarly, with the tree structure, you can make side-by-side comparisons of entire statements or statements components such as WHERE clauses.

## Experimenting on your own

The following topics cover additional experiments you can perform to get familiar with the SQL Profiler feature.

### Investigating the event category tabs

The event category tabs displayed differ by DBMS platform as follows:

- IBM DB/2 for Windows, Unix, and Linux - **Fetch, Cursor, Execution, Operation, Transaction, Connectivity, and Other**
- Oracle - **On CPU, System I/O, User I/O, Cluster, Application, Configuration, Commit, and Other**
- SQL Server - **CPU, Lock, Memory, Buffer, I/O, and Other**
- Sybase - **CPU, Lock, Memory, I/O, Network, and Other**

Click the various tabs to investigate the details provided. Entries on the wait event-associated tabs show how long sessions were waiting in events in that category over the length of the profiling session. The execution-based tab provides similar information.

### Zooming in on the graph

For lengthy profiling sessions, resolution for the one-second interval bars may be inadequate for your analysis. Use the zoom controls to zoom in and out:



While the default view is useful for finding general problem areas, zooming in can help you isolate more specific problem areas.

### **Displaying detailed statistics for a subset of the graph**

Once you have spotted a problem area on the graph, you can update the tabbed area to display only statistics for that part of the graph. Click-drag the mouse across several interval bars of the graph to select only those bars. Then investigate the changes to details on the various tabs in the lower part of the profiling editor.

### **Drilling down into statements and events**

For entries on the event category tab, SQL Profiler lets you drill down to find associations between wait events and SQL statements. On one of the event category tabs, click an entry to activate the drilldown control and then click the entry a second time to open the **SQL** tab on the SQL statement associated with the event you clicked on. This lets you find specific statements resulting in long wait event delays or that are associated with problematic wait event types.

Also try drilling down on a statement on the **SQL** tab to open a **SQL Text** tab displaying the full text of the statement.

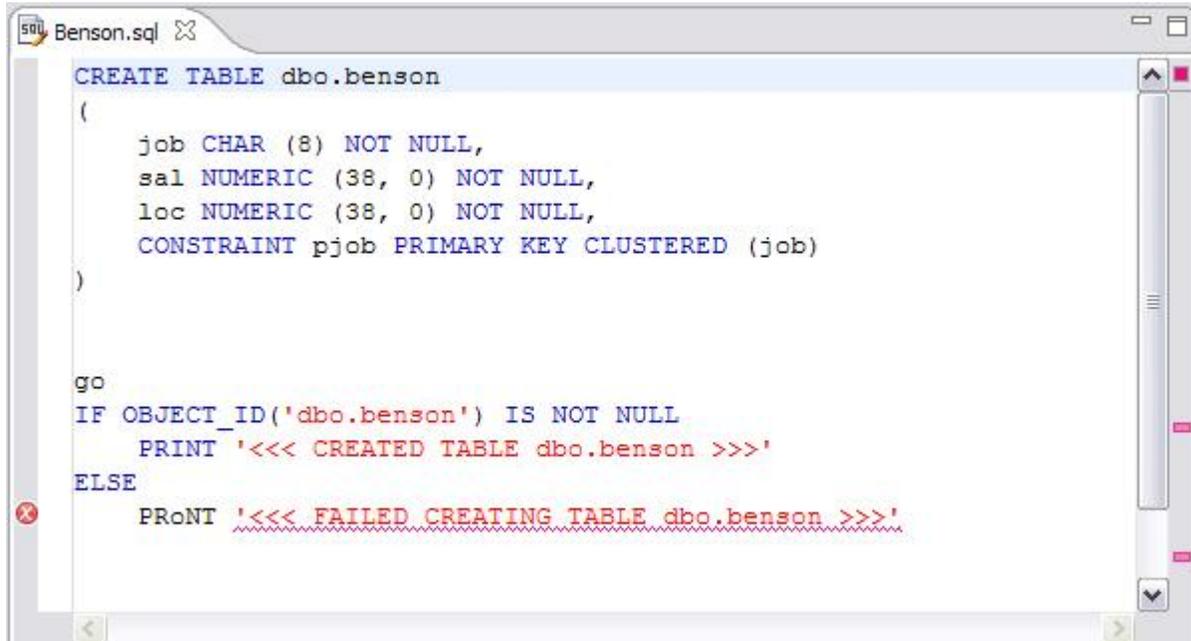
### **Saving, closing, and opening profiling archives**

You can save a profiling session in an archive, a file with a **.oar** suffix. This lets you return to your analysis at a later time. Experiment with the **File** menu's **Save**, **Close**, and **Open** options.

## Session 5: Working with SQL Code Assist

SQL Code Assist is an interface component that enables the development and formatting of SQL code for the purposes of creating and modifying database objects. It provides a front end application for the delivery of code through its object code extraction capabilities.

Code Assist provides a number of key features that assist in the coding process, ensuring greater accuracy in coding, faster development cycles, and a general increase in efficiency overall.



```

CREATE TABLE dbo.benson
(
    job CHAR (8) NOT NULL,
    sal NUMERIC (38, 0) NOT NULL,
    loc NUMERIC (38, 0) NOT NULL,
    CONSTRAINT pjob PRIMARY KEY CLUSTERED (job)
)

go
IF OBJECT_ID('dbo.benson') IS NOT NULL
    PRINT '<<< CREATED TABLE dbo.benson >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE dbo.benson >>>'
  
```

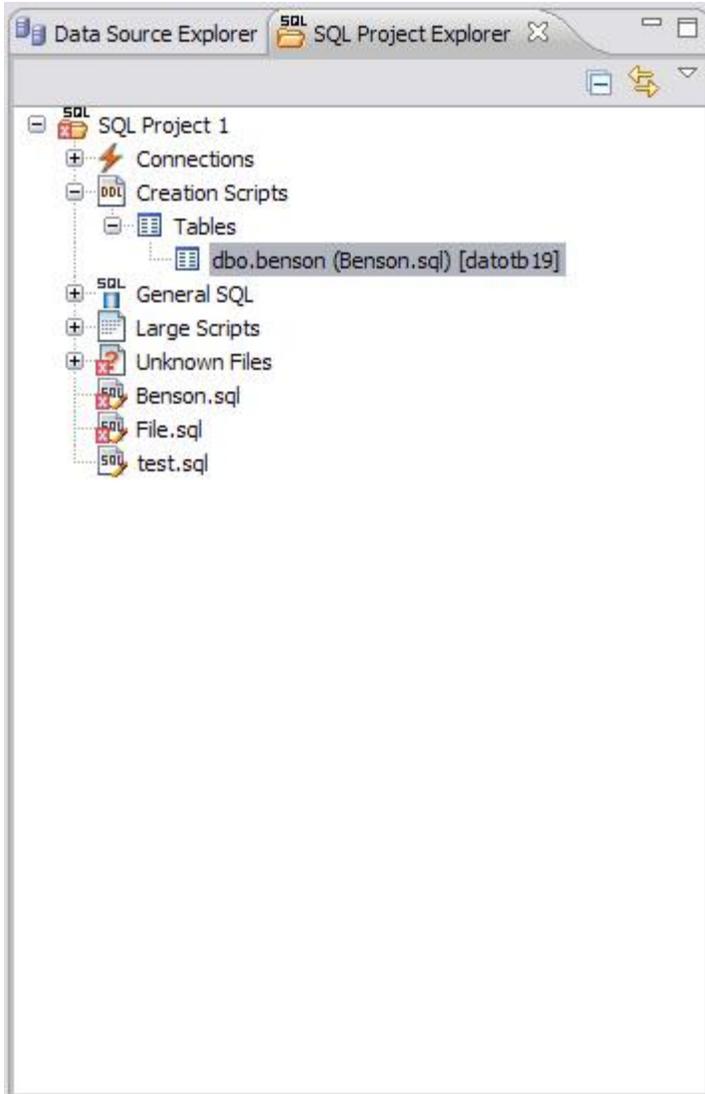
SQL Editor provides key features to ensure an increase in code accuracy and overall development efficiency.

The following key features are provided with SQL Code Assist:

- Code Extraction
- Code Highlighting
- Automatic Error Detection
- Code Complete
- Hyperlinks
- Code Formatting
- Code Folding

In order to access SQL Editor, you need to create a new file or edit existing code from Data Source Explorer.

Additionally, **SQL Project Explorer** provides a tree structure for all files created in DB Optimizer. You can access files from here by double-clicking the file name you want to open as well.



*SQL Project Explorer provides a tree of all files developed and saved in DB Optimizer.*

#### **To create a new file:**

Choose **File > New > Other**.

A **New** dialog opens.

Expand the **SQL Development** folder, select **SQL File**, and then click **Finish**.

A blank instance of SQL Editor appears in the Workbench. If you save this file, it is automatically added to the SQL Project Explorer.

#### **To edit an existing file:**

Use Data Source Explorer or Project Explorer to navigate to the code you want to modify and double-click it.

An instance of SQL Editor appears in the Workbench, populated with the extracted code of the specified object or SQL project file.

## Code Extraction

SQL Editor provides the ability to extract the underlying SQL code of database objects registered in DB Optimizer to provide a front end application for the development and modification of data sources in your enterprise.

### To extract underlying SQL code:

- Navigate to a database object in **Data Source Explorer** and select **Extract** via the right-click menu.

The object's underlying SQL code appears in SQL Editor and is ready for editing and further modification.

## Code Highlighting

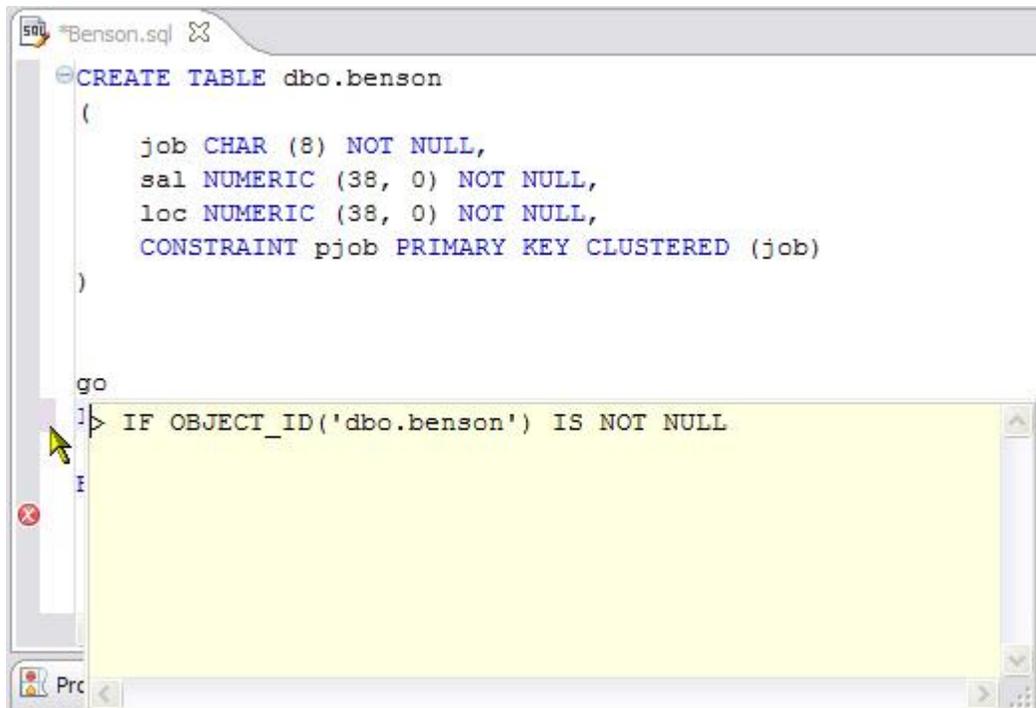
SQL Editor identifies commands and provides syntax highlighting changes that are automatically added to the code as you add lines, which enables you to clearly and quickly understand code when you read it in the interface.

The following syntax highlighting is automatically added to lines of code in SQL Editor:

Code	Formatting
Comments	Green italic font
SQL Commands	Dark blue font
Syntax Errors	Red underlined font
Coding Errors	Red font

- **Comments:** green italics
- **SQL Commands:** dark blue
- **Syntax Errors:** red underline
- **Coding Errors:** red

Additionally, SQL Editor provides a purple change bar in the left-hand column that indicates if a line of code has been modified from the original text. You can hover over this change bar to view the original code line. A red square icon in the right-hand column indicates that there are errors in the code line. You can hover over the icon to view the error count.



The screenshot shows a SQL editor window with a tab titled "Benson.sql". The main text area contains the following SQL code:

```
CREATE TABLE dbo.benson
(
    job CHAR (8) NOT NULL,
    sal NUMERIC (38, 0) NOT NULL,
    loc NUMERIC (38, 0) NOT NULL,
    CONSTRAINT pjob PRIMARY KEY CLUSTERED (job)
)

go

IF OBJECT_ID('dbo.benson') IS NOT NULL
```

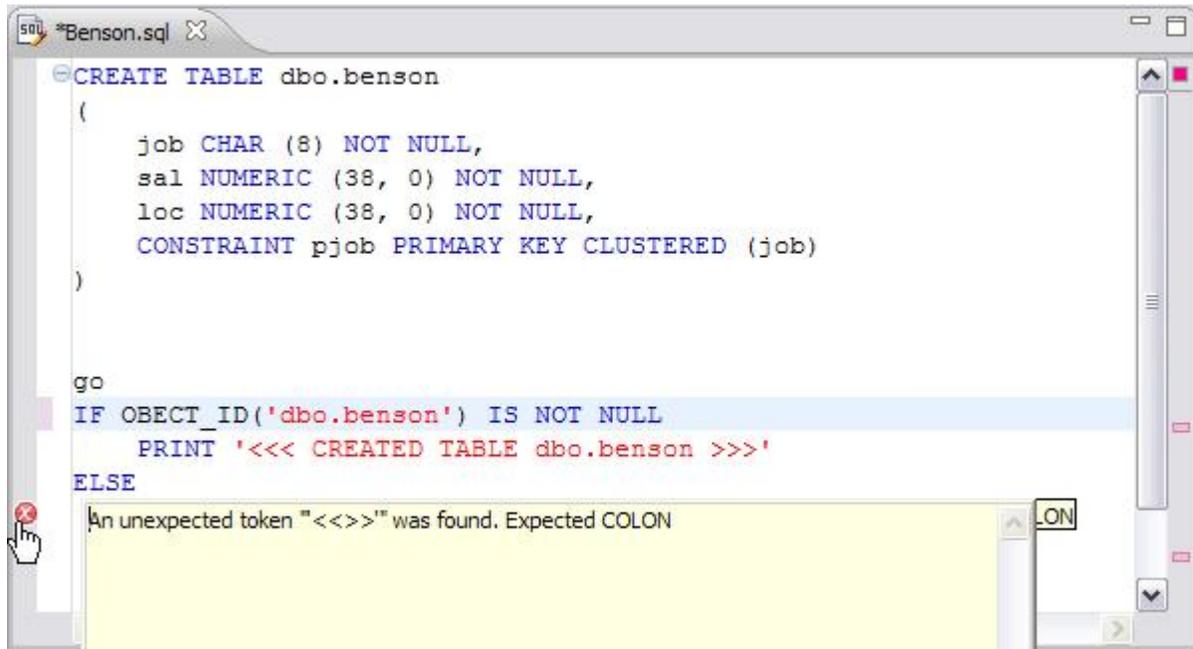
A yellow highlight covers the bottom portion of the code, specifically the line `IF OBJECT_ID('dbo.benson') IS NOT NULL`. A purple vertical bar is visible on the left side of the editor, indicating a change in the text. A mouse cursor is hovering over this bar.

The purple change bar indicates if a line of code has been changed from its original text. Hover your mouse over the change bar to view the original text.

## Automatic Error Detection

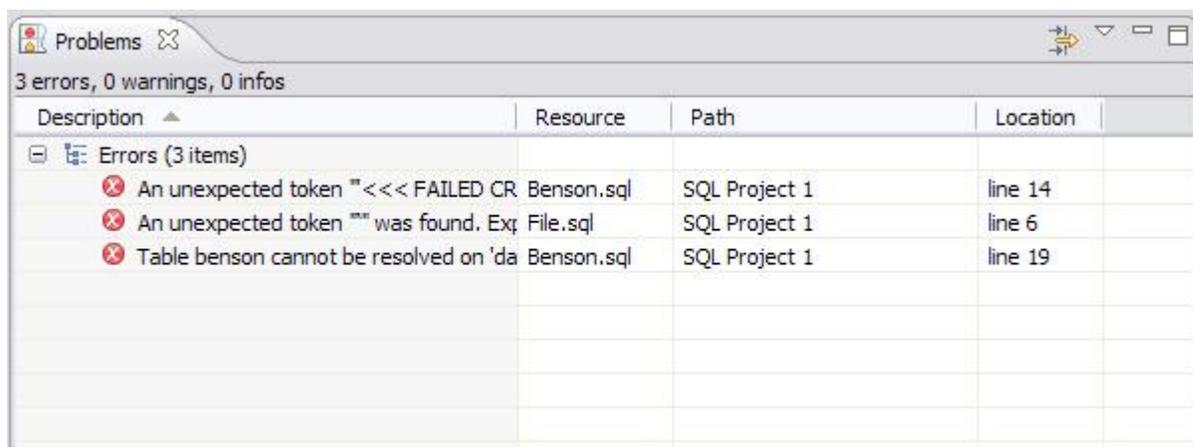
The automatic error detection functionality of the editor highlights errors and typos in the code as you work in “real time”.

Automatic error detection automatically identifies and analyzes SELECT, FROM, WHERE, GROUP BY, HAVING, and ORDER BY statements. If it detects any syntax errors while you type these statements, the line is automatically flagged by the error icon in the left-hand column of SQL Editor. You can hover your mouse over the icon to view any errors.



Syntax errors are automatically flagged by line as you work with code in SQL Editor. Hover the mouse over the error icon to view the specific error message.

Additionally, all semantic errors are recorded in the **Problems** view, an interface component that automatically logs errors and warnings as you work with files.



The Problems view logs errors and warnings as you work with files in SQL Editor.

You can double-click on a line in the Problems view and DB Optimizer will automatically navigate you to that issue in SQL Editor.

## Code Complete

SQL Editor provides suggestion capabilities for both DML statements and objects. It has the ability to look up object names in order to avoid errors when defining table names, columns, etc. in the development process.

This feature provides lists of object and code suggestions that will make the development process more efficient as you will not be forced to manually look up object names and other statement values.

SQL Editor provides code assist for SELECT, UPDATE, INSERT, and DELETE statements and object suggestion support for tables, alias tables, columns, alias columns, schemas, and catalogs.

### To activate code assist:

1. Click the line on which you want to activate the code assist feature.
2. Press **CTRL + Spacebar** on your keyboard. Code assist analyzes the line and presents a list of suggestions as appropriate based on the elements of the statement.

## Hyperlinks

Hyperlinks are used in SQL Editor to provide links to tables, columns, packages and other reference objects. When you select a hyperlinked object from a piece of code, a new editor opens and displays the source.

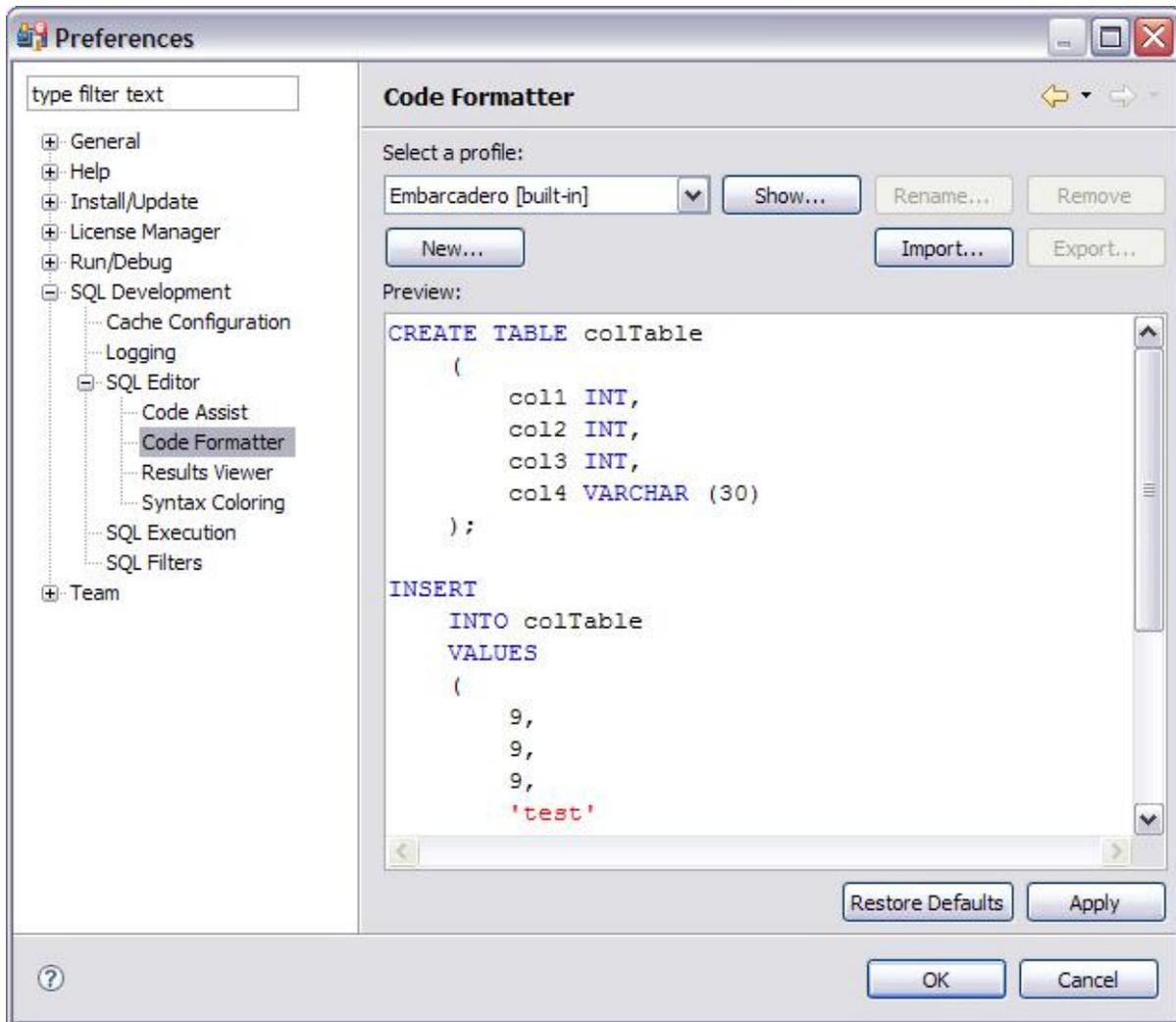
Additionally, hyperlinks can be used to link procedures or the function of a call statement, as well as function calls in DML statements.

To enable a hyperlink, hover your mouse over the object name and hold the **CTRL** key. It becomes underlined and changes color.

## Code Formatting

Code formatting is automatically applied to a file as you develop it in SQL code. This enables you to set global formatting preferences one time, and then apply it to all code development, saving time and allowing for a more efficient code development process.

The code formatter can be accessed by selecting **CTRL + Shift + F** in the editor. All code is automatically formatted based on parameters specified on the **Code Formatter** node of the **Preferences** panel. (Select **Window > Preferences** in the Main Menu to access this panel.)



Code formatting parameters can be globally set and then applied to all development work in SQL Editor.

In addition to formatting code per individual file, you can also format an entire group of files from **Project Explorer**. Select the directory of files that you want to apply formatting to and execute the **Format** command from the right-click menu. The files will be automatically formatted based on the global preferences.

## Code Folding

The code folding feature automatically sorts code into a tree-like outline structure in SQL Editor. This increases navigation and clarification capacities during the code development process. It ensures that the file will be easily understood should any future work be required on the code.

As you work in SQL Editor, collapsible nodes are automatically inserted into the appropriate lines of code. Statements can then be expanded or collapsed as needed, and this feature is especially useful when working on parts of particularly large or complicated files.

## Session 6: Executing SQL Code

When you have finished developing or modifying code, you can then execute the file from within the DB Optimizer environment, on the database of your choosing. This enables you to immediately execute code upon completion of its development. Alternatively, you can save files for execution at a later point in time.

In order to execute a file, you must first associate it with a target database. This is performed by using the drop down menus located in the **Toolbar**. When a SQL file is open in the Workbench, the menus are enabled. Select a data source and a corresponding database to associate the file with and then click the green arrow icon to execute the file.



*The pair of drop down menus indicate that the SQL file is associated with the dato1b19 data source and EMBCM database. When the green arrow icon on the right-hand side of the menus is selected, the file is executed on the specified data source and database.*

Additionally, if you have turned off auto-committal in the **Preferences** panel (**Window > Preferences**) you can commit and execute transactions via the **Commit Transaction** and **Start Transaction** icons located beside the **Execute** icon.

### To execute a file:

Open the file you want to run and ensure it is associated with the correct database, then click the **Execute** icon. DB Optimizer executes the code on the database you specified.

### To execute a transaction:

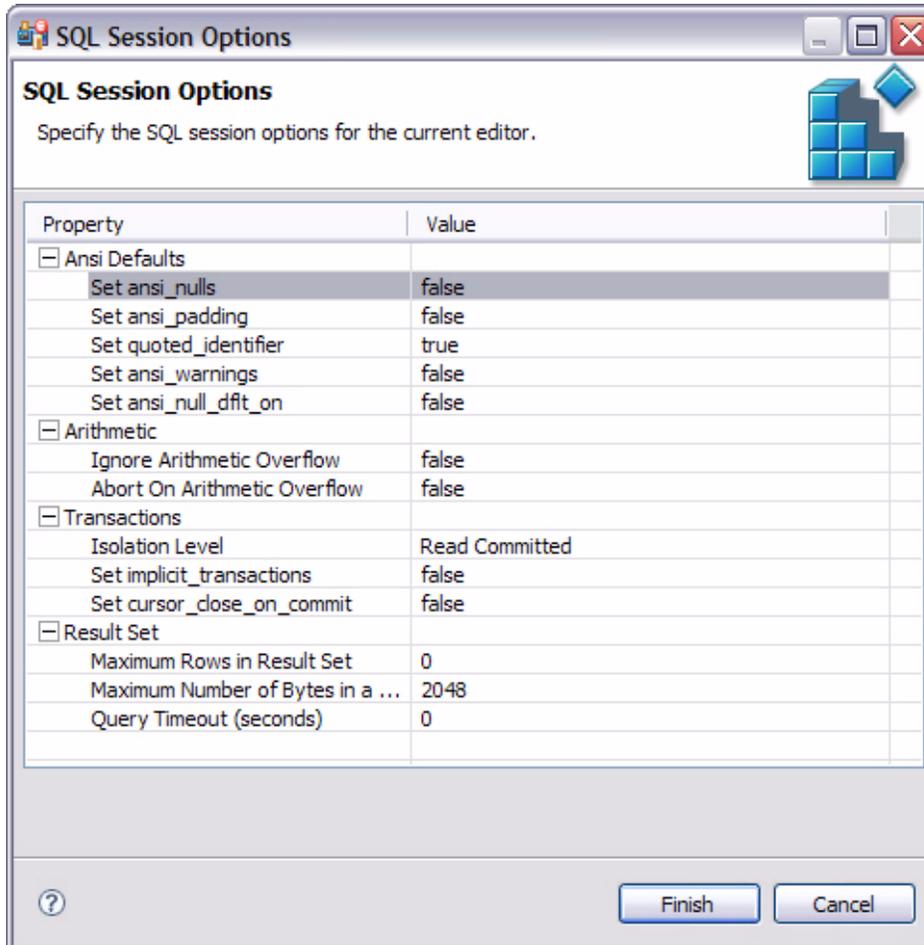
Open the transaction file you want to run and ensure it is associated with the correct database, then click the **Start Transaction** icon. DB Optimizer executes the transaction on the database you specified.

### To commit a transaction:

Open the transaction file you want to commit, ensure it is associated with the correct database, and click the **Commit Transaction** icon. DB Optimizer commits the transaction on the database you specified.

## Configuring SQL Execution Parameters

If you do not want to use the default execution options provided by DB Optimizer, use the **SQL Session Options** dialog to modify the configuration parameters that determine how DB Optimizer executes code. These options ensure that code is executed the way you want on a execution-per-execution basis, ensuring accuracy and flexibility when running new or modified code.



*SQL Sessions Options provide you with the flexibility to adjust execution parameters on a session-by-session basis.*

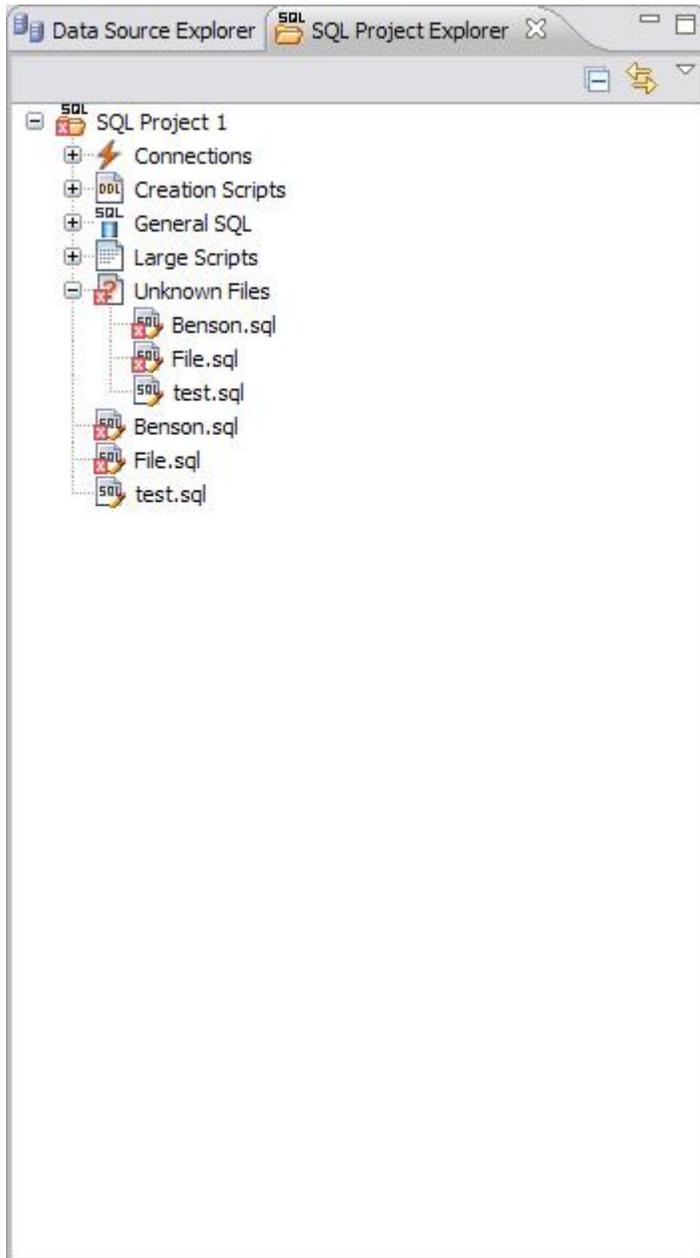
### To modify SQL session options:

1. Click the SQL Session Options icon in the Toolbar. The **SQL Sessions Options** dialog appears.
2. Click on the individual parameters in the **Value** column to change the configuration of each property, as specified.
3. Click **Finish**. The session options are changed and DB Optimizer executes the code as specified by your options.

**Note:** SQL Session Options are only applied to the currently-selected code, and are not retained across different files with regards to execution.

## Session 7: Working with SQL Project Insight

SQL Project Explorer provides a tree view of all projects created in DB Optimizer for the purpose of organizing and storing SQL development files. This provides a place to keep work in progress organized, as well as maintaining a common directory structure from which the development and execution of files can be implemented.



*SQL Project Explorer provides a directory structure to organize and maintain SQL files.*

When you save a file in SQL Editor, SQL Project Insight is able to determine the type of SQL file you created and automatically organizes it in the proper directory folder in SQL Project Explorer. Additionally, SQL Project Explorer contains a folder for all data source connections you have indicated when creating a file:

- **Connections** lists all data source connections of the files associated with the project.
- **Creation Scripts** contains DDL statements and statements that define database objects.
- **General SQL** is a category for all other SQL files not used in database object creation. This includes DML files, etc.
- **Large Scripts** contains all files larger than the currently indicated file size preference. This limit can be modified on the **Preferences** panel. (Select **Windows>Preferences** in the Menu Bar.)

Additionally, SQL Project Insight flags files that contain semantic or syntactic errors with a red X icon beside the name of the file in the tree. This enables you to quickly look through the files listed and find erroneous files without having to open each file in SQL Editor.

SQL Project Insight also features a code indexer that provides a preview look at code in each file. As with error detection, the indexer enables you to search through SQL files without having to open the files in the editor.

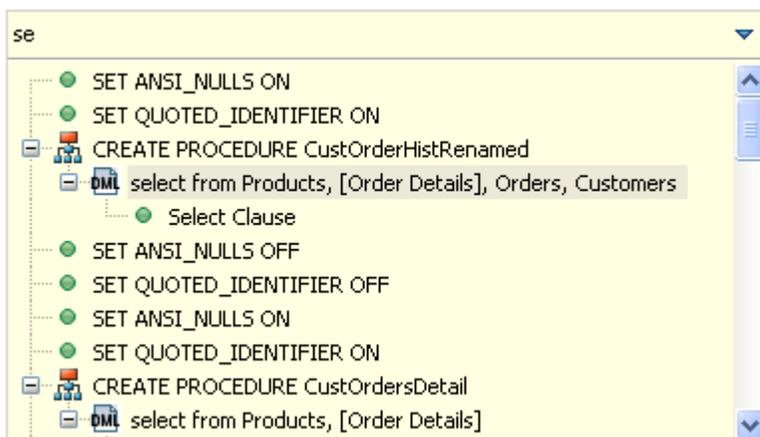
#### To create a new file:

1. Select **File > New > SQL Project** from the Main Menu.  
The **New Project Wizard** appears.
2. Enter the appropriate information in the fields provided by the wizard.
3. Click **Finish**.

The new project icon appears in the **SQL Project Explorer** view under the name you specified. If you did not indicate a default location, the project appears in the appropriate directory when you open it in DB Optimizer.

Once you have created and saved a file, it appears in the appropriate directory and node in **SQL Project Explorer**. If you specify a data source to which the file is associated, that data source name appears under the **Connections** node.

Additionally, the **Outline** view and **Quick Outline** function in SQL Editor displays the hierarchical structure of the SQL code. This provides easy navigation during the development process, especially for users who are not extremely familiar with SQL code.



*The Quick Outline function displays the hierarchical structure of the code file currently on display in SQL Editor.*

**To open the Outline view:**

- Select **Window > Show View > Outline** on the Menu Bar.

The **Outline** view appears in the Workbench, displaying the hierarchy of the code open in SQL Editor.

**To use Quick Outline:**

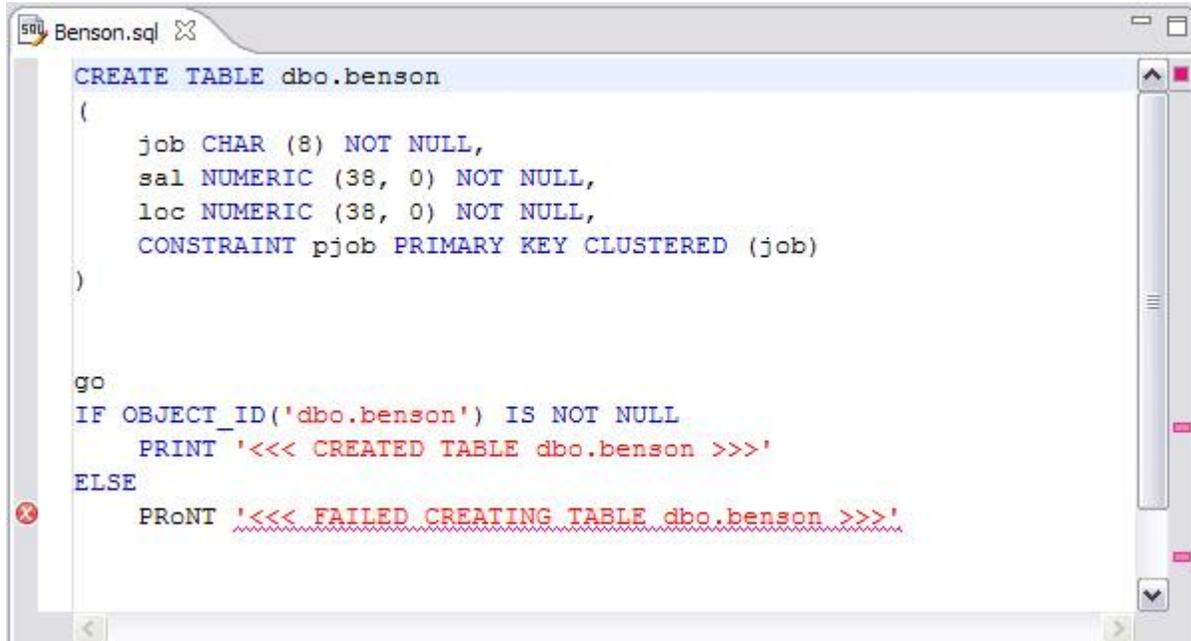
- In SQL Editor, press **Ctrl + O**.

The Quick Outline pane opens and displays a hierarchical view of the code open in SQL Editor.

## Session 8: Working with SQL Code Assist

SQL Code Assist is an interface component that enables the development and formatting of SQL code for the purposes of creating and modifying database objects. It provides a front end application for the delivery of code through its object code extraction capabilities.

Code Assist provides a number of key features that assist in the coding process, ensuring greater accuracy in coding, faster development cycles, and a general increase in efficiency overall.



```

CREATE TABLE dbo.benson
(
    job CHAR (8) NOT NULL,
    sal NUMERIC (38, 0) NOT NULL,
    loc NUMERIC (38, 0) NOT NULL,
    CONSTRAINT pjob PRIMARY KEY CLUSTERED (job)
)

go
IF OBJECT_ID('dbo.benson') IS NOT NULL
    PRINT '<<< CREATED TABLE dbo.benson >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE dbo.benson >>>'
  
```

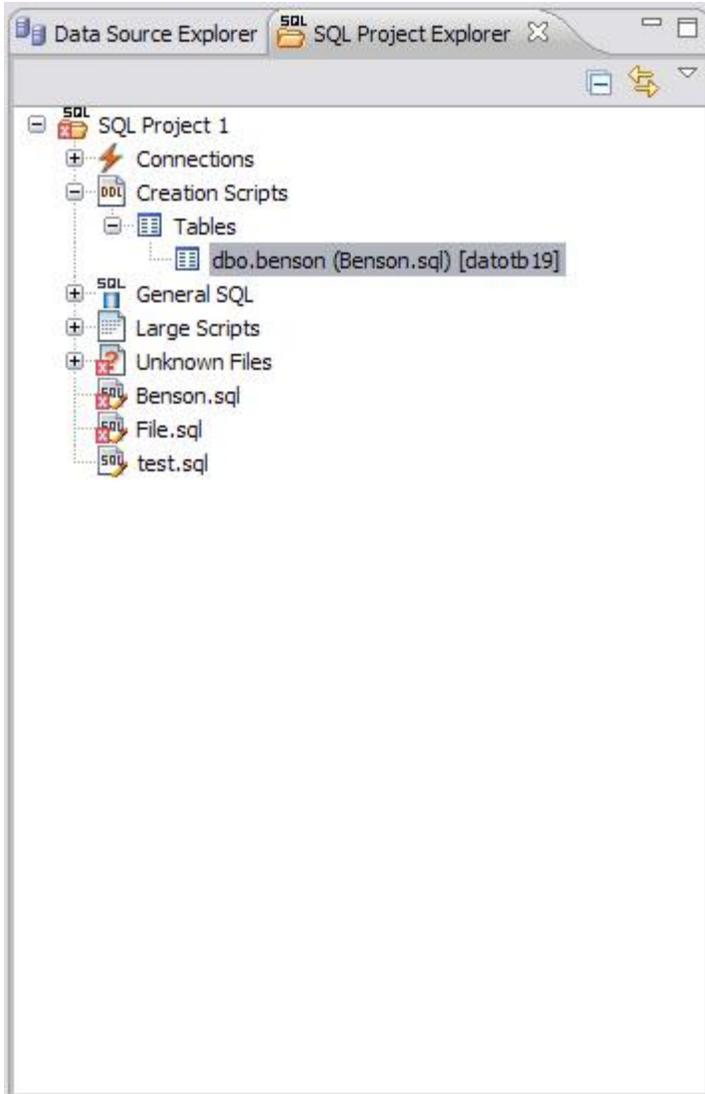
SQL Editor provides key features to ensure an increase in code accuracy and overall development efficiency.

The following key features are provided with SQL Code Assist:

- Code Extraction
- Code Highlighting
- Automatic Error Detection
- Code Complete
- Hyperlinks
- Code Formatting
- Code Folding

In order to access SQL Editor, you need to create a new file or edit existing code from Data Source Explorer.

Additionally, **SQL Project Explorer** provides a tree structure for all files created in DB Optimizer. You can access files from here by double-clicking the file name you want to open as well.



*SQL Project Explorer provides a tree of all files developed and saved in DB Optimizer.*

#### **To create a new file:**

Choose **File > New > Other**.

A **New** dialog opens.

Expand the **SQL Development** folder, select **SQL File**, and then click **Finish**.

A blank instance of SQL Editor appears in the Workbench. If you save this file, it is automatically added to the SQL Project Explorer.

#### **To edit an existing file:**

Use Data Source Explorer or Project Explorer to navigate to the code you want to modify and double-click it.

An instance of SQL Editor appears in the Workbench, populated with the extracted code of the specified object or SQL project file.

## Code Extraction

SQL Editor provides the ability to extract the underlying SQL code of database objects registered in DB Optimizer to provide a front end application for the development and modification of data sources in your enterprise.

### To extract underlying SQL code:

- Navigate to a database object in **Data Source Explorer** and select **Extract** via the right-click menu.

The object's underlying SQL code appears in SQL Editor and is ready for editing and further modification.

## Code Highlighting

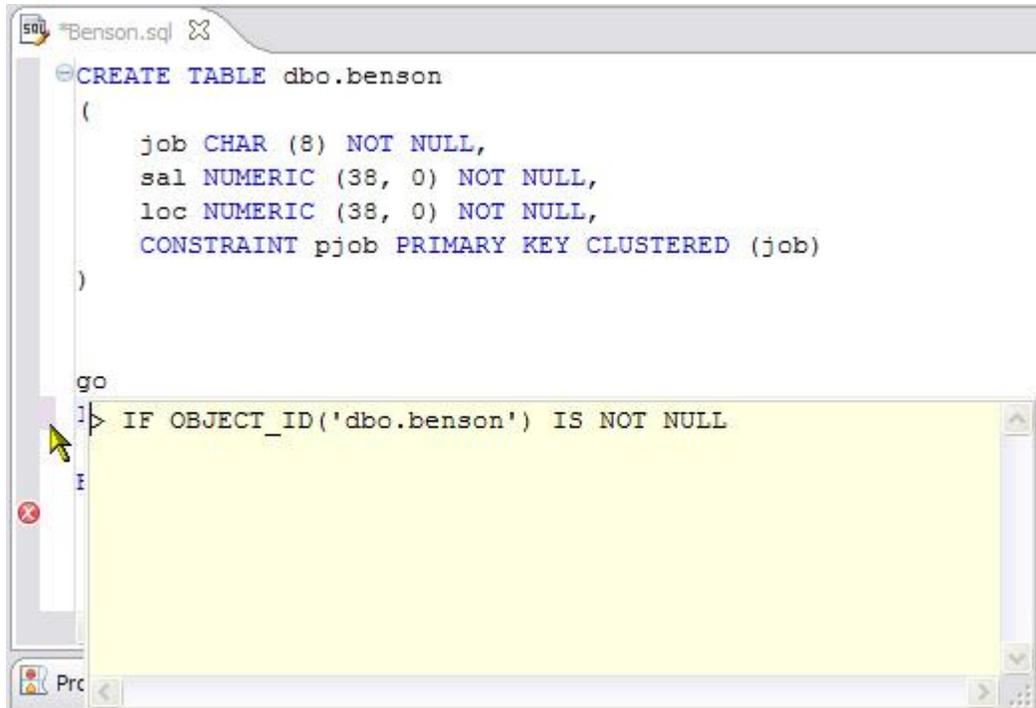
SQL Editor identifies commands and provides syntax highlighting changes that are automatically added to the code as you add lines, which enables you to clearly and quickly understand code when you read it in the interface.

The following syntax highlighting is automatically added to lines of code in SQL Editor:

Code	Formatting
Comments	Green italic font
SQL Commands	Dark blue font
Syntax Errors	Red underlined font
Coding Errors	Red font

- **Comments:** green italics
- **SQL Commands:** dark blue
- **Syntax Errors:** red underline
- **Coding Errors:** red

Additionally, SQL Editor provides a purple change bar in the left-hand column that indicates if a line of code has been modified from the original text. You can hover over this change bar to view the original code line. A red square icon in the right-hand column indicates that there are errors in the code line. You can hover over the icon to view the error count.



The screenshot shows a SQL editor window with a tab titled "Benson.sql". The main text area contains the following SQL code:

```
CREATE TABLE dbo.benson
(
    job CHAR (8) NOT NULL,
    sal NUMERIC (38, 0) NOT NULL,
    loc NUMERIC (38, 0) NOT NULL,
    CONSTRAINT pjob PRIMARY KEY CLUSTERED (job)
)

go

IF OBJECT_ID('dbo.benson') IS NOT NULL
```

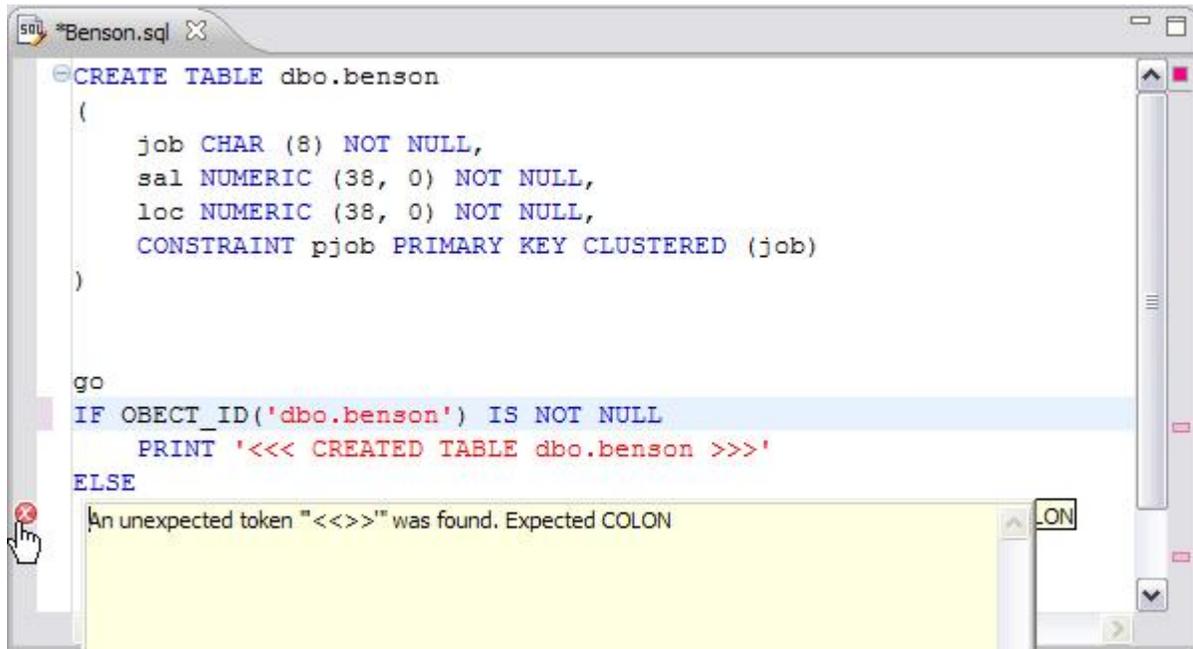
A yellow highlight is applied to the last line of code. A purple change bar is visible on the left side of the editor, indicating a change to the text. A mouse cursor is hovering over this bar.

The purple change bar indicates if a line of code has been changed from its original text. Hover your mouse over the change bar to view the original text.

## Automatic Error Detection

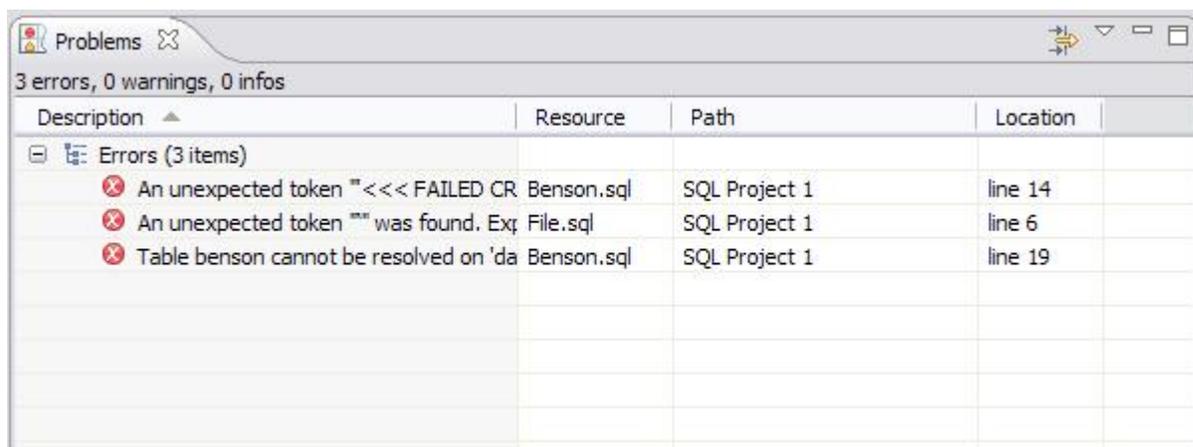
The automatic error detection functionality of the editor highlights errors and typos in the code as you work in “real time”.

Automatic error detection automatically identifies and analyzes SELECT, FROM, WHERE, GROUP BY, HAVING, and ORDER BY statements. If it detects any syntax errors while you type these statements, the line is automatically flagged by the error icon in the left-hand column of SQL Editor. You can hover your mouse over the icon to view any errors.



Syntax errors are automatically flagged by line as you work with code in SQL Editor. Hover the mouse over the error icon to view the specific error message.

Additionally, all semantic errors are recorded in the **Problems** view, an interface component that automatically logs errors and warnings as you work with files.



The Problems view logs errors and warnings as you work with files in SQL Editor.

You can double-click on a line in the Problems view and DB Optimizer will automatically navigate you to that issue in SQL Editor.

## Code Complete

SQL Editor provides suggestion capabilities for both DML statements and objects. It has the ability to look up object names in order to avoid errors when defining table names, columns, etc. in the development process.

This feature provides lists of object and code suggestions that will make the development process more efficient as you will not be forced to manually look up object names and other statement values.

SQL Editor provides code assist for SELECT, UPDATE, INSERT, and DELETE statements and object suggestion support for tables, alias tables, columns, alias columns, schemas, and catalogs.

### To activate code assist:

3. Click the line on which you want to activate the code assist feature.
4. Press **CTRL + Spacebar** on your keyboard. Code assist analyzes the line and presents a list of suggestions as appropriate based on the elements of the statement.

## Hyperlinks

Hyperlinks are used in SQL Editor to provide links to tables, columns, packages and other reference objects. When you select a hyperlinked object from a piece of code, a new editor opens and displays the source.

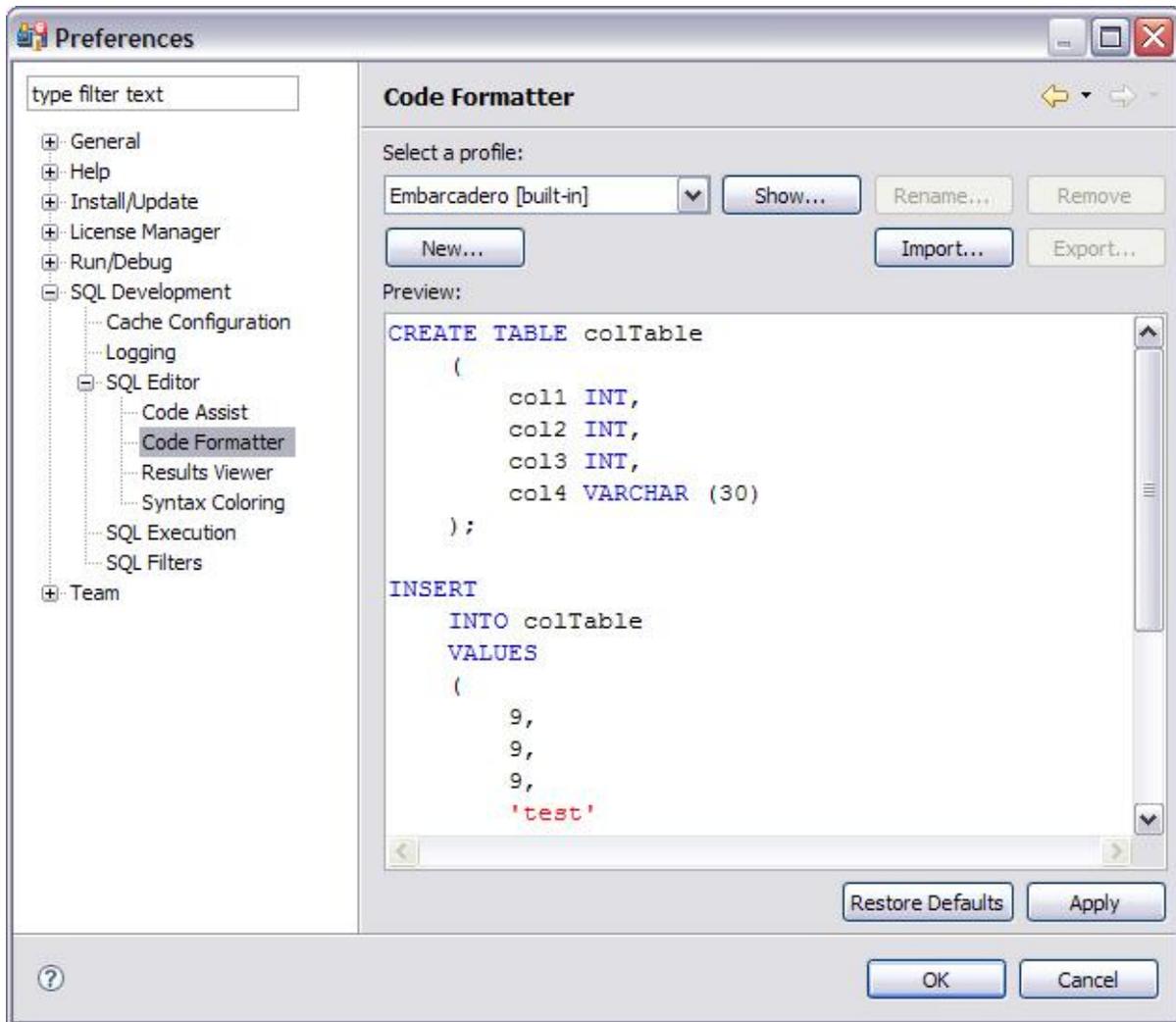
Additionally, hyperlinks can be used to link procedures or the function of a call statement, as well as function calls in DML statements.

To enable a hyperlink, hover your mouse over the object name and hold the **CTRL** key. It becomes underlined and changes color.

## Code Formatting

Code formatting is automatically applied to a file as you develop it in SQL code. This enables you to set global formatting preferences one time, and then apply it to all code development, saving time and allowing for a more efficient code development process.

The code formatter can be accessed by selecting **CTRL + Shift + F** in the editor. All code is automatically formatted based on parameters specified on the **Code Formatter** node of the **Preferences** panel. (Select **Window > Preferences** in the Main Menu to access this panel.)



Code formatting parameters can be globally set and then applied to all development work in SQL Editor.

In addition to formatting code per individual file, you can also format an entire group of files from **Project Explorer**. Select the directory of files that you want to apply formatting to and execute the **Format** command from the right-click menu. The files will be automatically formatted based on the global preferences.

## Code Folding

The code folding feature automatically sorts code into a tree-like outline structure in SQL Editor. This increases navigation and clarification capacities during the code development process. It ensures that the file will be easily understood should any future work be required on the code.

As you work in SQL Editor, collapsible nodes are automatically inserted into the appropriate lines of code. Statements can then be expanded or collapsed as needed, and this feature is especially useful when working on parts of particularly large or complicated files.

## Session 9: Executing SQL Code

When you have finished developing or modifying code, you can then execute the file from within the DB Optimizer environment, on the database of your choosing. This enables you to immediately execute code upon completion of its development. Alternatively, you can save files for execution at a later point in time.

In order to execute a file, you must first associate it with a target database. This is performed by using the drop down menus located in the **Toolbar**. When a SQL file is open in the Workbench, the menus are enabled. Select a data source and a corresponding database to associate the file with and then click the green arrow icon to execute the file.



*The pair of drop down menus indicate that the SQL file is associated with the dato1b19 data source and EMBCM database. When the green arrow icon on the right-hand side of the menus is selected, the file is executed on the specified data source and database.*

Additionally, if you have turned off auto-committal in the **Preferences** panel (**Window > Preferences**) you can commit and execute transactions via the **Commit Transaction** and **Start Transaction** icons located beside the **Execute** icon.

### To execute a file:

Open the file you want to run and ensure it is associated with the correct database, then click the **Execute** icon. DB Optimizer executes the code on the database you specified.

### To execute a transaction:

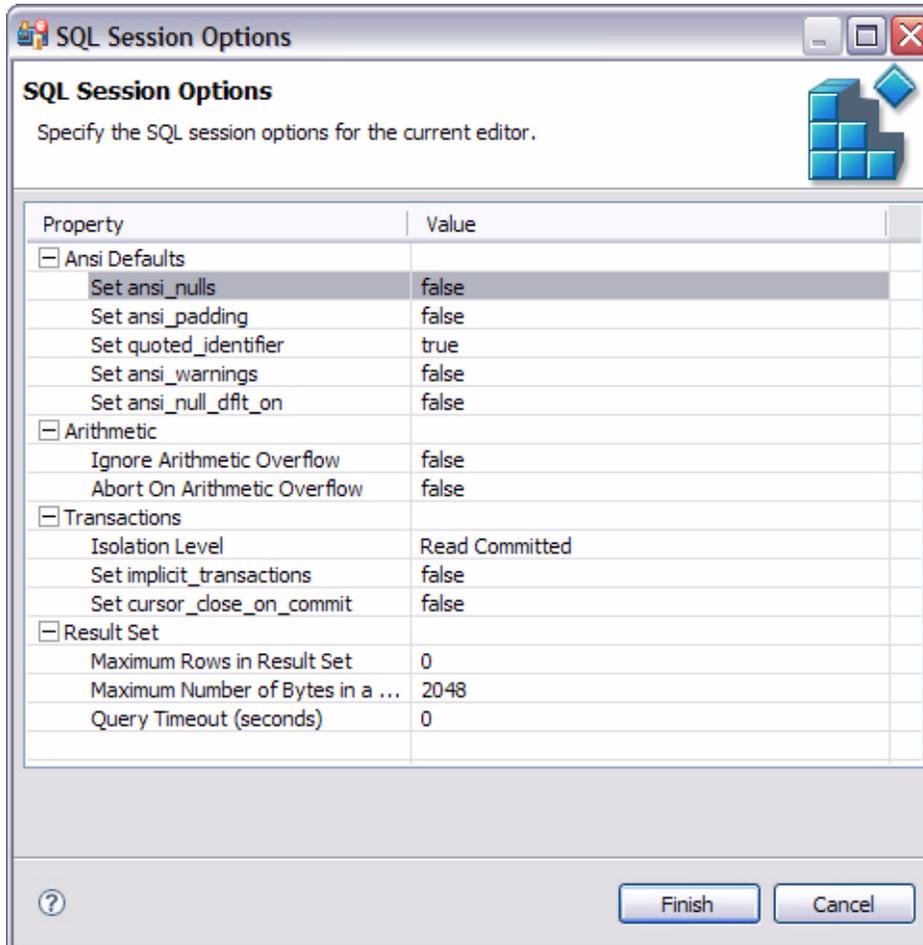
Open the transaction file you want to run and ensure it is associated with the correct database, then click the **Start Transaction** icon. DB Optimizer executes the transaction on the database you specified.

### To commit a transaction:

Open the transaction file you want to commit, ensure it is associated with the correct database, and click the **Commit Transaction** icon. DB Optimizer commits the transaction on the database you specified.

## Configuring SQL Execution Parameters

If you do not want to use the default execution options provided by DB Optimizer, use the **SQL Session Options** dialog to modify the configuration parameters that determine how DB Optimizer executes code. These options ensure that code is executed the way you want on a execution-per-execution basis, ensuring accuracy and flexibility when running new or modified code.



*SQL Sessions Options provide you with the flexibility to adjust execution parameters on a session-by-session basis.*

### To modify SQL session options:

4. Click the SQL Session Options icon in the Toolbar. The **SQL Sessions Options** dialog appears.
5. Click on the individual parameters in the **Value** column to change the configuration of each property, as specified.
6. Click **Finish**. The session options are changed and DB Optimizer executes the code as specified by your options.

**Note:** SQL Session Options are only applied to the currently-selected code, and are not retained across different files with regards to execution.

## Additional Resources

### Licensing Your Embarcadero Technologies Product

All Embarcadero Technologies products include a 14-day trial period. To continue using the product without interruption, we recommend that you license it as soon as possible. To license your product, use the License Request Wizard found in the Help menu of your respective product. If you have not yet purchased your Embarcadero Technologies product, contact [sales@embarcadero.com](mailto:sales@embarcadero.com), or [uk.sales@embarcadero.com](mailto:uk.sales@embarcadero.com) for sales in the EMEA region.

### Embarcadero Technologies Product Support

The Embarcadero Technologies Web site is an excellent source for additional product information, including white papers, articles, FAQs, discussion groups, and the Embarcadero Knowledge Base. Go to [www.embarcadero.com/resources](http://www.embarcadero.com/resources), or click any of the links below, to find:

- [Documentation](#)
- [Online Demos](#)
- [Technical Papers](#)
- [Discussion Groups](#)
- [Knowledge Base](#)
- [FAQ](#)

### Embarcadero Technologies Technical Support

If you have a valid maintenance contract with Embarcadero Technologies, the Embarcadero Technical Support team is available to assist you with any problems you have with our applications. Our maintenance contract also entitles registered users of Embarcadero Technologies products to download free software upgrades during the active contract period. Evaluators receive free technical support for the term of their evaluation (14 days).

We encourage you to open technical support cases via the [Technical Support request form](#) at the [Embarcadero Technologies Web site](#). For additional information about Embarcadero Technologies Technical Support, go to the Support page on our Web site.

### Embarcadero Technologies on the Web

To download evaluations of other Embarcadero Technologies products or to learn more about our company and our products visit us at [www.embarcadero.com](http://www.embarcadero.com)