

DOWNLOAD PDF MICROSOFT ACCESS DEVELOPERS GUIDE TO SQL SERVER

Chapter 1 : Microsoft Access Developer's Guide to SQL Server | InformIT

Microsoft Access Developer s Guide to Microsoft SQL Server September 29th, - Microsoft Access Developer s Guide to Microsoft SQL Server provides much needed information and guidance for the.

In the Connect to Server window, do the following: For Server type, select Database Engine usually the default option. For Authentication, select Windows Authentication. If you select SQL Login, you will be prompted for a username and password. For more information about authentication types, see Connect to server database engine. You can also modify additional connection options by selecting Options. This article uses the default values for all the options. Examples of successful connections To verify that your SQL Server connection succeeded, expand and explore the objects within Object Explorer. Create a database Create a database named TutorialDB by doing the following: Right-click your server instance in Object Explorer, and then select New Query: Into the query window, paste the following T-SQL code snippet: After the query is complete, the new TutorialDB database appears in the list of databases in Object Explorer. Create a table in the new database In this section, you create a table in the newly created TutorialDB database. Because the query editor is still in the context of the master database, switch the connection context to the TutorialDB database by doing the following: In the database drop-down list, select the database that you want, as shown here: Paste the following T-SQL code snippet into the query window, select it, and then select Execute or select F5 on your keyboard. You can either replace the existing text in the query window or append it to the end. To execute everything in the query window, select Execute. To execute a portion of the text, highlight that portion, and then select Execute. Insert rows into the new table Insert some rows into the Customers table that you created previously. To query the Customers table and view the rows that were previously inserted, do the following: Customers; The results of the query are displayed under the area where text was entered: Modify the way results are presented by selecting one of the following options: The middle button displays the results in Grid View, which is the default option. The first button displays the results in Text View, as shown in the image in the next section. The third button lets you save the results to a file whose extension is. Verify your connection properties by using the query window table You can find information about the connection properties under the results of your query. After you run the previously mentioned query in the preceding step, review the connection properties at the bottom of the query window. You can also view the query duration and the number of rows that are returned by the previously executed query. In the image, note that the results are displayed in Text View. Change the server that the query window is connected to You can change the server that your current query window is connected to by doing the following: The Connect to Server window opens again. Change the server that your query is connected to. Note This action changes only the server that the query window is connected to, not the server that Object Explorer is connected to. Go to the next article to learn more:

DOWNLOAD PDF MICROSOFT ACCESS DEVELOPERS GUIDE TO SQL SERVER

Chapter 2 : SQL Server Performance Microsoft Access Developer's Guide to SQL Server

Microsoft Access Developer's Guide to Microsoft SQL Server provides much needed information and guidance for the Access power user or developer who wants to exploit the power of SQL Server.

Logins and Built-in Accounts. Granting Database Access to Logins. Understanding Fixed Server Roles. Understanding Fixed Database Roles. Working with User-Defined Database Roles. Keeping Track of Logins and Roles. Removing Users and Logins. The Three Types of Permissions. Implementing Security Using Stored Procedures. Implementing Security Using Views. Encrypting Functions, Views, and Stored Procedures. What About Access Security? Upsizing Access Security Permissions on Tables. Advantages of Access Projects. Disadvantages of Access Projects. Creating a New Access Project. Connecting to SQL Server. Forms in Access Projects. Working Efficiently with Linked Tables. Creating Efficient Bound Forms. Restricting Data with Prefiltering. Loading a Single Record from an Unbound Form. Minimizing the Number of Recordsets Open. Creating a Pass-Through Query. Caching Data in Local Tables. Refreshing Data in Local Tables. Using the Options Dialog Box. Using the Windows Registry. Using an MSysConf Table. Getting Started with ADO. Declaring and Creating a Connection Object. Opening the Connection Object. Connection Properties and Methods. Updating Data Using Recordsets. Navigating, Searching, Filtering, and Sorting. Command Object Properties and Methods. Executing a Simple Command. Executing a Stored Procedure. Executing a Stored Procedure with Parameters. Mixing and Matching the Three Objects. Opening Recordsets with Command Objects. Reusing Command Objects with Recordsets. Going Solo with the Connection Object. Microsoft Jet and Replication Objects 2. Choosing the Best Tool for the Job at Hand. Designing a Relational Database. Review of Relational Database Design Principles. How Databases Are Stored in Files. How the Transaction Log Works. Creating Databases Using the Enterprise Manager. Creating and Modifying Tables. Setting Relationships Between Tables. Summarizing and Aggregating Data. Updating Data with Action Queries. Functions to Return Information. Control of Flow Syntax. Creating Distributed Queries with Linked Servers. Creating and Optimizing Views. What Views Are Good For. Rules for Creating and Using Views. Creating and Modifying Views. Creating Views Using the Enterprise Manager. Creating Views Using an Access Project. Creating Views Using the Query Analyzer. Views, Subqueries, and Outer Joins. Sorting and Top Values in Views. Using Views with User-Defined Functions. Using Derived Tables and Nested Views. Using Views to Update Data. Restrictions on Updating Data Through Views. When to Use Indexed Views. When Not to Use Indexed Views. Requirements for Indexed Views. Indexed Views and Inline Functions. Optimizing Views and Queries. Using the Query Analyzer to Tune Performance. Programming Effective Stored Procedures. Understanding Batches, Scripts, and Stored Procedures. Basic Stored Procedure Syntax. Stored Procedures and Security. Tools for Creating Stored Procedures. Creating Parameterized Stored Procedures. Declaring and Using Variables in Stored Procedures. Input and Output Parameters. Error Handling in Stored Procedures.

DOWNLOAD PDF MICROSOFT ACCESS DEVELOPERS GUIDE TO SQL SERVER

Chapter 3 : Learn to Develop with Microsoft Developer Network | MSDN

If you are an Access developer and you want to start writing professional database programs with SQL Server then this is the book to get. Even if you're already writing programs with SQL Server as your database, believe me, you need this book, too.

Any forms, reports, queries, or code that previously worked with the original Office Access tables are automatically connected to the new SQL Server tables. In an application that uses linked SQL Server tables, two different database engines are at work: The interaction of these two engines can sometimes yield results that are inferior to those obtained by using only the Jet database engine with native Office Access tables. This white paper discusses several of these issues and presents strategies for resolving them. Most of these issues relate to performance or updatability. Although performance does often improve, there are many cases where it remains the same or even degrades. In some cases, performance of certain queries degrades to an unacceptable level. The major cause of query performance degradation is when a query involving very large tables requires that all of the data from one or more tables be downloaded to the client. This can happen even when joins or criteria appear to limit the result set to a small number of records. This occurs because sometimes the Office Access database engine determines that it cannot submit an entire query to SQL Server. Instead, it submits multiple queries, often including queries that request all of the rows in a table, and then it combines or filters the data on the client. If the criteria require local processing, even queries that should return only selected rows from a single table can require that all the rows in the table be returned. The primary strategy for improving performance is to minimize the amount of data returned to the Office Access client and maximize the amount of processing that occurs on the server. To accomplish this, you need to be able to analyze the SQL commands that Office Access is submitting. To listen in on the conversation from the server side, you can open the SQL Server Profiler and create a new trace. Select a template that shows TSQL to see all the statements being processed by the server. From the client side, you can edit a Microsoft Windows registry setting that allows you to see the commands that the Office Access database engine is submitting to ODBC. As always, be very careful when editing the Windows registry. For more information on backing up and editing the registry, see *How to Modify the Windows Registry*. From the Windows Start menu, select Run. Type Regedit to open the Registry Editor. If you are using a version of Office Access prior to Office Access 2000, navigate to the following registry key, which appears as a folder in the Registry Editor. If you are using Office Access 2000, navigate to the following registry key. If Office Access is open when you make this change, you must close and reopen Office Access for the change to take effect. After making this change in the registry, queries submitted to any ODBC data source are logged in a text file named Sqlout. Unless you delete this file or its contents, it continues to grow as new queries are executed and the tracing activity degrades performance. It is very important to return to the Registry Editor and turn the feature off by changing the TraceSQLMode setting back to 0 when you are done testing. Running SQL Profiler also has a negative impact on performance, so try to avoid using it on a production server and close your Profiler traces when you are done testing. Before you can make productive use of these diagnostic tools, you must understand how Office Access interacts with SQL Server. Without that understanding, the SQL statements that you see in Profiler traces and in Sqlout logs can be quite puzzling. Suppliers, and that explicitly names the three columns in the table. Instead, both the Sqlout. The following is what is written to Sqlout. All the quotation marks that you see around object names are comparable to the brackets that Office Access uses and that also can be used in SQL Server to handle spaces or other illegal characters in names. The question marks are placeholders for parameters. A Profiler trace shows the three corresponding Transact-SQL statements that are processed on the server. This is often referred to as a keyset. Then Office Access prepares a parameterized SQL statement to select all the columns in the table or query for 10 rows at a time. The final step is to execute this statement, which is assigned a number on the server 6 in the example , as many times as needed, passing in 10 bookmark

values at a time. If there are two columns in the bookmark, 20 values are passed in at a time to specify the next 10 rows. In this example, there are only three rows in the table, so the final bookmark value, 3, which corresponds to the last ShipperID in the table, is submitted eight times, because the rows are always fetched in sets of 10. The statement that fetches 10 rows of data is repeated as many times as necessary to fill the current screen and to provide some room for scrolling in either direction. The remaining rows are not fetched immediately unless the user performs an action such as scrolling that brings additional rows into view. In the background during idle time, the remaining rows are gradually filled in until the recordset is complete. In addition, any rows that remain visible are continually refreshed according to a configurable refresh interval that defaults to 60 seconds. Long memo and OLE object values are retrieved in separate queries only when their columns and rows are visible. Dynasets support a continuous two-way conversation between Office Access and SQL Server for each recordset that is open. The rows that are visible are continually refreshed to show the latest data, creating extra network traffic. However, the dynamic nature of these recordsets can also reduce traffic by immediately retrieving only those rows in the vicinity of data the user is actually viewing. If you create a form that is bound to a table containing a million rows of data not a recommended practice and the form shows the data from only one row at a time, only 20 rows are retrieved when the form opens. If the user keeps only the first record visible, Office Access continually retrieves the first 10 rows, every 60 seconds by default. If the form is left open long enough, all the rows are eventually retrieved during idle time in many separate batches, but a snapshot retrieves all the rows right away. Because they work with only a few rows at a time, dynasets minimize the duration that read locks are held on the server. This allows other users to modify data without having to wait as long as is necessary for locks to clear. When the user edits or deletes a row, Office Access executes an update or delete query with a WHERE clause containing not only the bookmark value, which is used to locate the row to update or delete, but also the values for all the other columns. If the table contains a timestamp column, which is a column that SQL Server automatically updates when the row is modified, only that one column value is added to the WHERE clause. Issues related to updatability are discussed later in this white paper. The following is a summary of the statements shown in a Sqlout.

DOWNLOAD PDF MICROSOFT ACCESS DEVELOPERS GUIDE TO SQL SERVER

Chapter 4 : Microsoft Access Developer's Guide to SQL Server |

In Access , Microsoft's made it possible for Microsoft Access developers to leverage SQL Server as their native database engine while still using Access's familiar application design environment, giving their desktop database applications a solid migration path and unprecedented scalability.

You can use the DSVM in two different ways: As a workstation If you want to get started with R in the cloud quickly and easily, this is your best bet. The environment will be familiar to anyone who has worked with R on a local workstation. However, instead of using local resources, the R environment runs on a VM in the cloud. If your data is already stored in Azure, this has the added benefit of allowing your R scripts to run "closer to the data. Instead of investing in powerful workstations for each developer and requiring team members to synchronize on which versions of the various software packages they will use, each developer can spin up an instance of the DSVM whenever needed. As a compute platform In addition to being used as a workstation, the DSVM is also used as an elastically scalable compute platform for R projects. You can form the instances into a cluster and deploy a distributed analysis to be performed in the cloud. This entire process can be controlled by R code running on your local workstation. This solution provides the latest capabilities for R-based analytics on datasets of virtually any size, loaded to either Azure Blob or Data Lake storage. This is an enterprise-grade solution that allows you to scale your R code across a cluster. This allows R to crunch data on a much larger scale than is possible with single-threaded R running on a workstation. It provides a flexible and scalable platform for running your R scripts in the cloud. Azure Databricks Azure Databricks is an Apache Spark-based analytics platform optimized for the Microsoft Azure cloud services platform. Designed with the founders of Apache Spark, Databricks is integrated with Azure to provide one-click setup, streamlined workflows, and an interactive workspace that enables collaboration between data scientists, data engineers, and business analysts. Users can create, share, and edit notebooks with other users of the systems. These notebooks allow users to write code that executes against Spark clusters managed in the Databricks environment. These notebooks fully support R and give users access to Spark through both the SparkR and sparklyr packages. Since Databricks is built on Spark and has a strong focus on collaboration, the platform is often used by teams of data scientists that work together on complex analyses of large data sets. Because the notebooks in Databricks support other languages in addition to R, it is especially useful for teams where analysts use different languages for their primary work. The article "What is Azure Databricks? Azure Machine Learning Studio Azure Machine Learning Studio is a collaborative, drag-and-drop tool you can use to build, test, and deploy predictive analytics solutions in the cloud. It enables emerging data scientists to create and deploy machine learning models without the need to write much code. ML Studio supports both R and Python. You can use R with ML Studio in two ways. Custom R scripts in your experiments First, you can extend the data manipulation and machine learning capabilities of ML Studio by writing custom R scripts. Although ML Studio includes a wide variety of modules for preparing and analyzing data, it cannot match the capabilities of a mature language like R. Therefore, the service was designed to allow you to introduce your own custom R scripts in cases where the provided modules do not meet your needs. To leverage this capability, drag and drop an "Execute R Script" module into your experiment. Then use the code editor in the "Properties" pane to write a new R script or paste an existing script. Within the script, you can reference external R packages. You can use the script to manipulate data or to train complex ML models that are not part of the standard ML Studio model library. This package, which is maintained by Microsoft, allows you to upload and download datasets to and from Azure ML, to interrogate experiments, to publish R functions as Azure ML web services, and to run R data through existing web services and retrieve the output. This package makes it much easier to use Azure ML as a scalable deployment platform for your R code. Instead of clicking and dragging in the UI, you can automate the entire deployment process using tools you already know. This service provides cloud-scale job scheduling and compute management so you can scale your R workload

DOWNLOAD PDF MICROSOFT ACCESS DEVELOPERS GUIDE TO SQL SERVER

across tens, hundreds, or thousands of virtual machines. Since it is a generalized computing platform, there are a few options for running R jobs on Azure Batch. This R package is a parallel backend for the foreach package. It allows each iteration of the foreach loop to run in parallel on a node within the Azure Batch cluster. For an introduction to the package, please read the "doAzureParallel: For a detailed walk-through, consult "R Workloads on Azure Batch. This provides an economical way to run Spark jobs in Azure. Azure Notebooks Azure Notebooks is a low-cost, low-friction method for R developers who prefer working with notebooks to bring their code to Azure. It is a free service for anyone to develop and run code in their browser using Jupyter, which is an open-source project that enables combining markdown prose, executable code, and graphics onto a single canvas. While Azure Notebooks is a viable option for small-scale projects, it has some limitations that make it inappropriate for large-scale data science projects. However, for publishing smaller analyses, this is an easy, no-cost option. It allows you to use the full power of SQL Server without any hassle of setting up the infrastructure. Instead of extracting data from the database and loading it into the R environment, you load your R code directly into the database and let it run right alongside the data. It is currently in limited preview but will continue to evolve.

Chapter 5 : SQL Data Developer | Microsoft Docs

Discuss: Microsoft Access Developer's Guide to SQL Server - Developer's Guide - reference book Sign in to comment. Be respectful, keep it clean and stay on topic.

Chapter 6 : www.nxgvision.com: Customer reviews: Microsoft Access Developer's Guide to SQL Server

Microsoft Access Developer's Guide to SQL Server Mary Chipman and Andy Baron ISBN No. This is THE book to have if you're developing applications using Microsoft Access linked to a Microsoft SQL Server database.

Chapter 7 : SQL Server – Pricing and Licensing | Microsoft

If you don't have access to a SQL Server instance, select your platform from the following links. If you choose SQL Authentication, use your SQL Server login credentials. Windows: Download SQL Server Developer Edition.

Chapter 8 : Free eBooks from Microsoft Press - Microsoft Virtual Academy

SQL Server licensing makes choosing the right edition simple and economical. Unlike other major vendors, there's no having to pay for expensive add-ons to run your most demanding applications – because every feature and capability is already built in.

Chapter 9 : SQL Server | Microsoft

Azure SQL Database is Microsoft's intelligent, fully managed relational cloud database service. It allows you to use the full power of SQL Server without any hassle of setting up the infrastructure. It allows you to use the full power of SQL Server without any hassle of setting up the infrastructure.