LAB 5

WORKING WITH DATABASES

This lab contains the following exercises and activities:

|  |  |
| --- | --- |
| Exercise 5.1 | Creating a Database |
| Exercise 5.2 | Selecting and Setting a Recovery Model |
| Exercise 5.1 | Creating a Database |
| Scenario | You are the DBA for a midsize company with offices in various cities throughout the United States and Canada. You have just installed a new instance of SQL Server, and now you need to create a database to hold data for your sales department. |
| Duration | This task should take approximately 30 minutes. |
| Setup | All you need for this task is access to the machine you installed SQL Server. |
| Caveats | This task doesn’t have any caveats. |
| Procedure | In this task, you will create a database that will hold data for the sales department. You will use this database in later tasks for storing other database objects as well. |
| Equipment Used | All you need for this task is access to the machine you installed SQL Server. |
| Objective | To decide where to put the data and log files. Use these guidelines:* Data and log files should be on separate physical drives so that, in case of a disaster, you have a better chance of recovering all data.
* Transaction logs are best placed on a RAID-1 array because this has the fastest sequential write speed together with redundancy.
* Data files are best placed on a RAID-5 array because they have faster read speed than other RAID-arrays together with redundancy.
* If you have access to a RAID-10 array, you can place data and log files on it because it has all the advantages of RAID-1 and RAID-0.
 |
| Criteria for Completion | You have completed this task when you have a database named Sales that you can see in SQL Server Management Studio. |

 PART A: Calculating the Storage Requirements

1. Calculate the space used by a single row of the table.

* To do this, add the storage requirements for each data type in the table.
* Add the null bitmap using this formula: null\_bitmap = 2 + ((number of columns + 7) /8).
* Calculate the space required for variable length columns using this formula: variable\_datasize = 2 + (num\_variable\_columns X 2) + max\_varchar\_size.
* Calculate the total row size using this formula: Row\_Size = Fixed\_Data\_Size + Variable\_Data\_Size + Null\_Bitmap + Row\_Header. The row header is always 4 bytes.

2. Calculate the number of rows that will fit on one page. Each page is 8,192 bytes with a header, so each page holds 8,096 bytes of data. Therefore, calculate the number of rows using this formula: 8096 ÷ (RowSize + 2).

3. Estimate the number of rows the table will hold. No formula exists to calculate this; you just need to have a good understanding of your data and user community.

4. Calculate the total number of pages that will be required to hold these rows. Use this formula: Total Number of Pages = Number of Rows in Table / Number of Rows Per Page.

|  |  |
| --- | --- |
| Question 1 | Why bother with calculations? You can always allocate more disk storage, right?**Answer: Right. Log space, though, needs to be allocated with the maximum space needed to avoid unnecessary virtual log files—too many VLFs slow processing. Data space growth requires significant overhead which might occur in the middle of a time critical transaction. You may find you need to add another disk before your financial manager can provide you authority to make another purchase—and emergency requests have negative consequences.** |

 PART B: Creating a Database named Sales

1. Start **SQL Server Management Studio**.

2. Connect to your default instance of SQL Server.

3. Expand your **Databases** folder.

4. Right-click either the **Databases** folder in the console tree or the white space in the right pane, and choose **New Database** from the context menu.

5. You should now see the General tab of the Database properties sheet. Enter the database name **Sales**, and leave the owner as <default>.

6. In the Database files grid, in the Logical Name column, change the name of the Sales file to **Sales\_Data**. Use the default location for the file, and make sure the initial size is 3.

7. Click the **ellipsis button** (the one with three periods) in the Autogrowth column for the Sales\_Data file. In the dialog box that opens, check the Restricted File Growth radio button, and restrict the filegrowth to 20 MB. Click **OK**.

8. To add a secondary data file, click the **Add** button, and change the logical name of the new file to Sales\_Data2. Here, too, use the default location for the file, and make sure the initial size is 3.

9. Restrict the filegrowth to a maximum of 20 MB for Sales\_Data2 by clicking the **ellipsis** button in the Autogrowth column.

10. Leave all of the defaults for the Sales\_Log file.

11. Click **OK** when you are finished. You should now have a new Sales database.

 PART C: Verifying Results

1. In the Object Browser, click **Databases**

2. Press **F5** or right-click database and choose **Refresh**.

3. Verify that your new database named Sales now appears.

|  |  |
| --- | --- |
| Exercise 5.2 | Selecting and Setting a Recovery Model |
| Scenario | You have created a new database on your SQL Server, and you need to make sure it is being backed up as quickly and efficiently as possible. You know that, to ensure this, you need to configure the database to use the correct recovery model, so you decide to set the recovery model for the new database.  |
| Duration | This task should take approximately 15 minutes.  |
| Setup | For this task, you need access to the machine you installed SQL Server on and the AdventureWorks database installed with the sample data.  |
| Caveat | This task doesn’t have any caveats.  |
| Procedure | In this task, you will configure the AdventureWorks database to use the Full recovery model.  |
| Equipment Used | For this task, you need access to the machine you installed SQL Server on and the AdventureWorks database installed with the sample data |
| Objective | To set the recovery model for the AdventureWorks database. |
| Criteria for Completion | This task is complete when the AdventureWorks database is configured to use the Full recovery model as outlined in the details of this task. |

 PART A: Setting the Recovery Model

1. Open **SQL Server Management Studio**, and in Object Explorer, expand **Databases** under your server.

2. Right-click **AdventureWorks**, and click **Properties**.

3. On the Options page, select **Full** from the Recovery Model drop-down list.

4. Click **OK** to configure the model.

|  |  |
| --- | --- |
| Question 2 | Which configuration consumes the most log space?**Answer: Simple consumes the least; full consumes the most.** |

 PART B: Verifying Results

1. Open **Management Studio**.

2. Open your **Query Editor**.

3. Run this code:

SELECT DATABASEPROPERTYEX('AdventureWorks', 'Recovery')