

Optimizing Performance for SQL Based Applications

Lab 1 – Exploring Page Allocation Structures

Overview

You have reviewed the AdventureWorksLT database and, amongst other things, noticed high wait statistics for I/O. Before investigating further, you want to explore page and allocation structure.

Before starting this lab, you should view **Module 1 – Optimizing Database Structures** in the course *Optimizing Performance for SQL Based Applications*. Then, if you have not already done so, follow the instructions in the **Getting Started** document for this course to set up the lab environment.

What You'll Need

To complete the labs, you will need the following:

- A SQL Server instance with the AdventureWorksLT sample database. Review the Getting Started document for information about how to provision this.
- The lab files for this course

Challenge 1: Explore Page Structure

In this exercise, you will explore the page structure for the Product table.

Explore the Page Structure

1. Start Microsoft SQL Server Management Studio and connect to your database instance.
2. Click **New Query**, select the AdventureWorksLT database, type the following Transact-SQL into the query window, then click **Execute**:

```
SELECT db_name(database_id) AS Database_Name,  
       object_name([object_id]) AS Table_Name,  
       allocation_unit_type,  
       allocation_unit_type_desc,  
       allocated_page_file_id,  
       allocated_page_page_id,  
       page_type,  
       page_type_desc  
FROM sys.dm_db_database_page_allocations(db_id('AdventureWorksLT'), NULL, NULL, NULL,  
     'DETAILED')  
WHERE object_name([object_id])='Product'  
AND page_type IS NOT NULL;
```

3. Examine the query results, noting that there are four page types: IAM pages, index pages, and data pages.

Explore the Index Statistics

1. In the Microsoft SQL Server Management Studio query window, type the following Transact-SQL to examine the indexes for the **Product** table, then highlight the code and click **Execute**:

```
SELECT db_name(database_id) AS Database_Name,  
       object_name([object_id]) AS Table_Name,  
       index_id,  
       partition_number,  
       index_type_desc,  
       alloc_unit_type_desc,  
       index_depth,  
       index_level,  
       avg_fragmentation_in_percent,  
       fragment_count,  
       avg_fragment_size_in_pages,  
       page_count  
from sys.dm_db_index_physical_stats (db_id('AdventureWorksLT'), NULL, NULL, NULL, NULL)  
WHERE object_name([object_id])='Product'
```

2. Examine the query results, noting the index types and fragmentation.