جامعة طرابلس
كلية تقنية المعلومات

Advanced Databases قواعد البيانات المتقدمة

ITSE312

د. عبدالسلام منصور الشريف

a.abdoessalam@uot.edu.ly

المحاضرة الثالثة عاشر – الفهرسة

Indexing

Contents

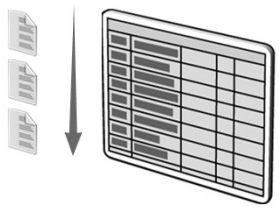
- Planning Indexes
- Creating Indexes
- Optimizing Indexes

Planning Indexes

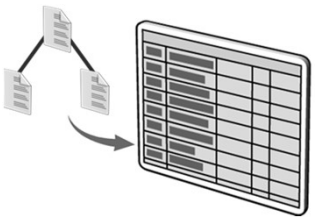
- How SQL Server Accesses Data
- What Is a Clustered Index?
- What Is a Heap?
- What Is a Nonclustered Index?

How SQL Server Accesses Data

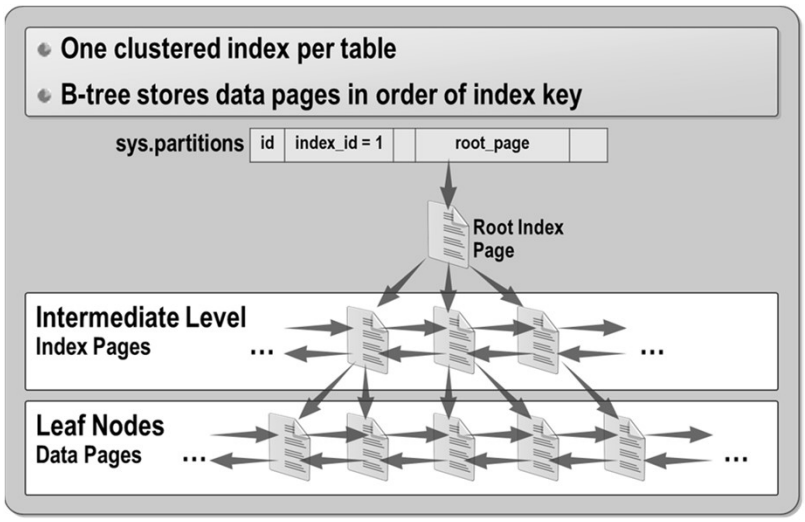
- Table scan
  - SQL Server reads all table pages



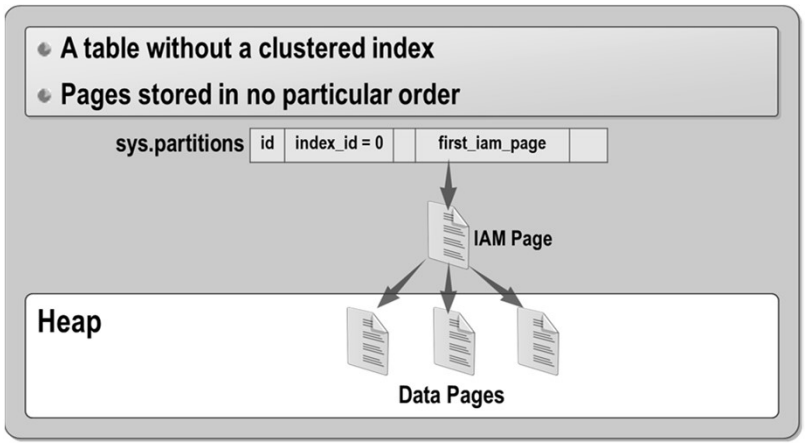
- Index
  - SQL Server uses index pages to find rows



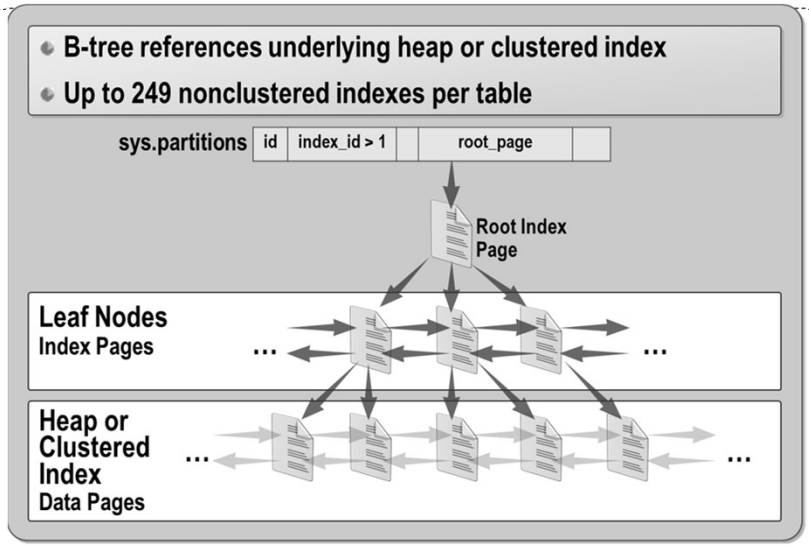
What Is a Clustered Index?



What Is a Heap?



What Is a Nonclustered Index?



Creating Indexes

- **Overview of Creating Indexes**
- **What Are Unique Indexes?**
- **Considerations for Creating Indexes with Multiple Columns**
- **When to Create Indexes on Computed Columns**
- **Options for Incorporating Free Space in Indexes**
- **Methods for Obtaining Index Information**
- **Practice: Creating Indexes**

Overview of Creating Indexes

```
CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ]
   INDEX index_name ON { table | view } ( column [ ASC | DESC
   ] [ ,...n ] )
   INCLUDE  ( column  [ ,...n ] )
   [WITH option [ ,...n ] ]
   [ON {partition_scheme (column) |  filegroup | "default" } ]
```

WITH option	Purpose
ALLOW_ROW_LOCKS	Enables/disables row-level locks on index
ALLOW_PAGE_LOCKS	Enables/disables page-level locks on index
ONLINE	Enables/disables access to index during creation
FILLFACTOR	Controls free space on leaf-level pages
PAD_INDEX	Controls free space on non-leaf-level pages

What Are Unique Indexes?

Ensures no duplicate values in index key

```
CREATE UNIQUE NONCLUSTERED INDEX [AK_Employee_LoginID]
  ON [HumanResources].[Employee] ( [LoginID] ASC)
```

EmployeeID	LoginID	Gender	MaritalStatus	...
216	mike0	M	S	...
231	fukiko0	M	M	...
242	pat0	M	S	...
...				
291	pat0	F	S	...

Duplicate key value not allowed

Considerations for Creating Indexes with Multiple Columns

- **Composite indexes**
  - **Include up to 16 columns and 900 bytes in key**
  - **Define most unique column first**

```
CREATE NONCLUSTERED INDEX K_Contact_LastName_FirstName
  ON Person.Contact ( LastName ASC, FirstName ASC)
```

- **Included columns**
  - **Nonkey columns included in index**
  - **Improve query "coverage" and therefore performance**

```
CREATE NONCLUSTERED INDEX AK_Employee_LoginID
  ON HumanResources.Employee ( LoginID ASC)
  INCLUDE ( ContactID, NationalIDNumber)
```

When to Create Indexes on Computed Columns

- **You can create indexes on computed columns when:**
- **Expression is deterministic and precise**
- **ANSI_NULLS connection–level option is ON**
- **Column does not evaluate to the text, ntext, or image data types**
- **Required options are set to ON when index is created and when changes cause index to update**
- **NUMERIC_ROUNDABORT option is set to OFF**

Query optimizer might ignore an index on a computed column

Options for Incorporating Free Space in Indexes Server-Level Roles

- Availability of free space affects performance of index updates
- FILLFACTOR determines the amount of free space on leaf nodes
  - Use low FILLFACTOR for OLTP applications
  - Use high FILLFACTOR for OLAP applications
- PAD_FILL determines the amount of free space on non-leaf index nodes

```
CREATE UNIQUE NONCLUSTERED INDEX [AK_Employee_LoginID]
ON [HumanResources].[Employee] ( [LoginID] ASC)
WITH ( FILLFACTOR = 65, PAD_INDEX = ON)
```

Index Fragmentation

- How fragmentation occurs
  - SQL Server reorganizes index pages when data is modified and causes index pages to split
- Types of fragmentation
  - Internal – Pages are not full
  - External – Pages are out of logical sequence
- Detecting fragmentation
  - SQL Server Management Studio – Index Properties window
  - System function – sys.dm_db_index_physical_stats

Options for Defragmenting Indexes

- <= 30% fragmentation = Reorganize

```
ALTER INDEX AK_Product_Name ON Production.Product
REORGANIZE
```

- > 30% fragmentation = Rebuild

```
ALTER INDEX AK_Product_Name ON Production.Product
REBUILD
```