

Overview

- Introduction to Indexes
- Index Architecture
- How SQL Server Retrieves Stored Data
- How SQL Server Maintains Index and Heap Structures
- Deciding Which Columns to Index

Introduction to Indexes

How SQL Server Stores and Accesses Data

Whether to Create Indexes

How SQL Server Stores and Accesses Data

How Data Is Stored

- Rows are stored in data pages
- Heaps are a collection of data pages for a table

How Data Is Accessed

- Scanning all data pages in a table
- Using an index that points to data on a page

Page 4		Page 5		Page 6		Page 7		Page 8		Page 9	
Con		Rudd		Akhtar		Smith		Martin		Ganio	
Funk		White		Funk		Ota		Phua		Jones	
White		Barr		Smith		Jones		Jones		Hall	
				Martin				Smith			

Data Pages

Whether to Create Indexes

Why to Create an Index

- Speeds up data access
- Enforces uniqueness of rows

Why Not to Create an Index

- Consumes disk space
- Incurs overhead

SQL Server Index Architecture

Using Heaps

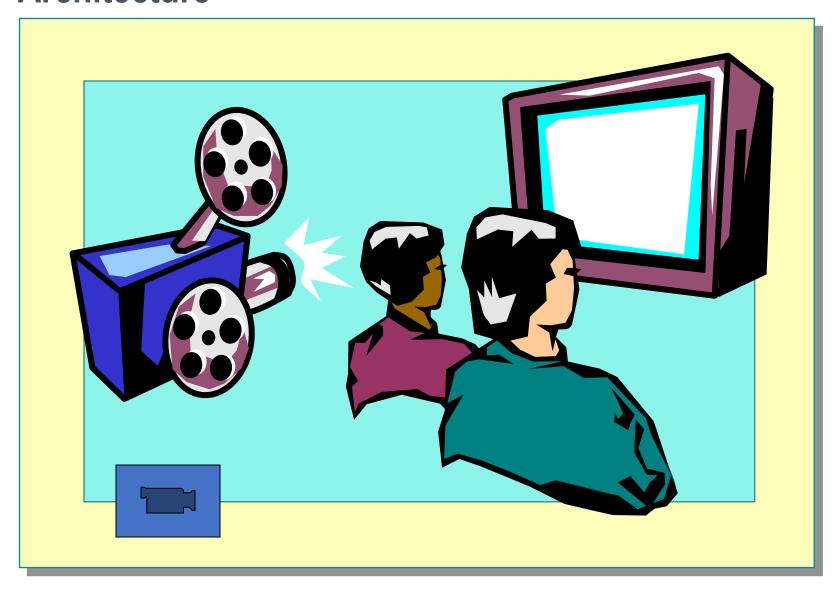
Using Clustered Indexes

Using Nonclustered Indexes



6

Multimedia Presentation: SQL Server Index Architecture



Using Heaps

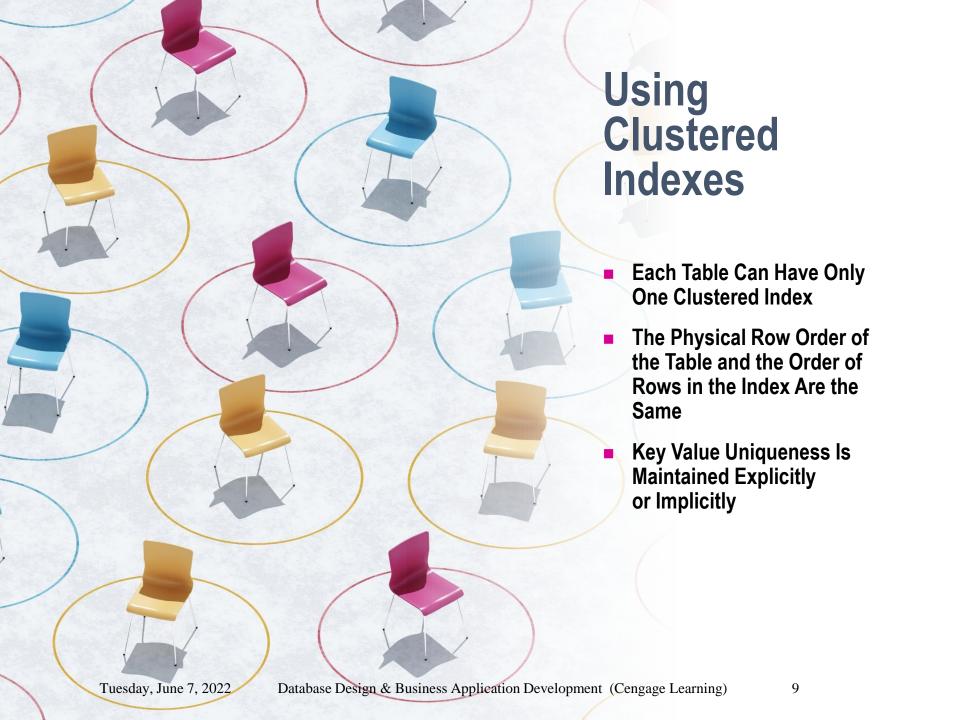
SQL Server:

Uses Index Allocation Map Pages That:

Reclaims Space for New Rows in the Heap When a Row Is Deleted Contain information on where the extents of a heap are stored

Navigate through the heap and find available space for new rows being inserted

Connect data pages



Using Nonclustered
Indexes



Nonclustered Indexes Are the SQL Server Default

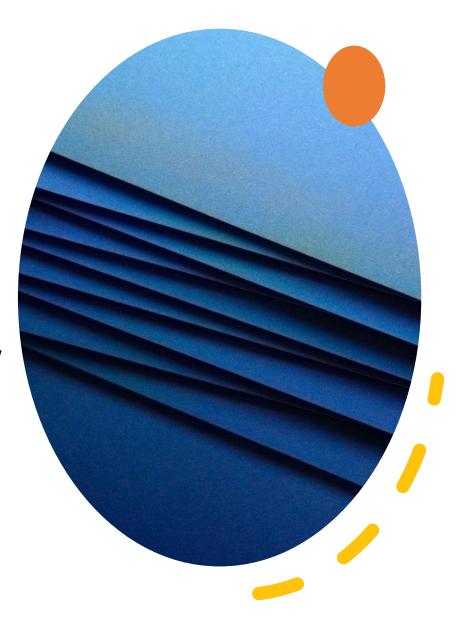


Existing Nonclustered Indexes Are Automatically Rebuilt When:

An existing clustered index is dropped

A clustered index is created

The DROP_EXISTING option is used to change which columns define the clustered index



How SQL Server Uses the sysindexes Table

Finding Rows Without Indexes

How SQL Server Retrieves Stored Data

Finding Rows in a Heap with a Nonclustered Index

Finding Rows in a Clustered Index

Finding Rows in a Clustered Index with a Nonclustered Index

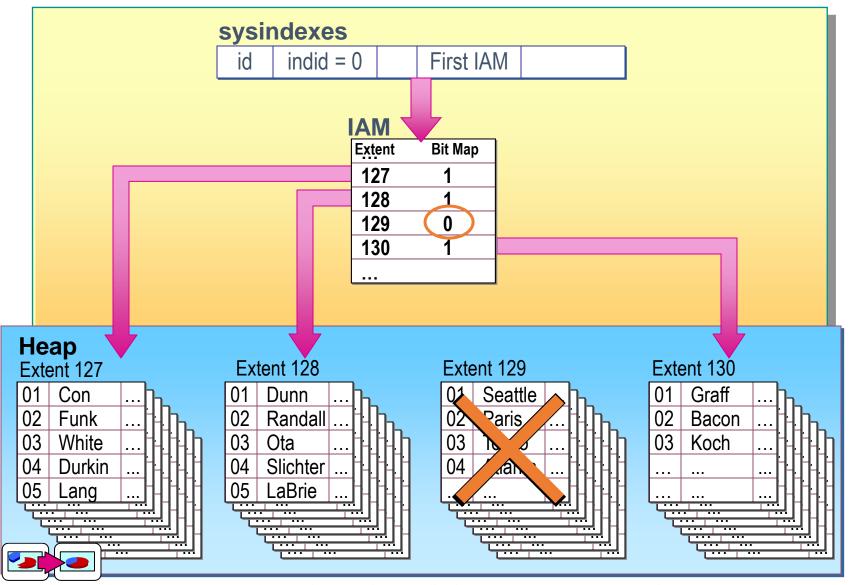
How SQL Server Uses the sysindexes Table

Describes the Indexes

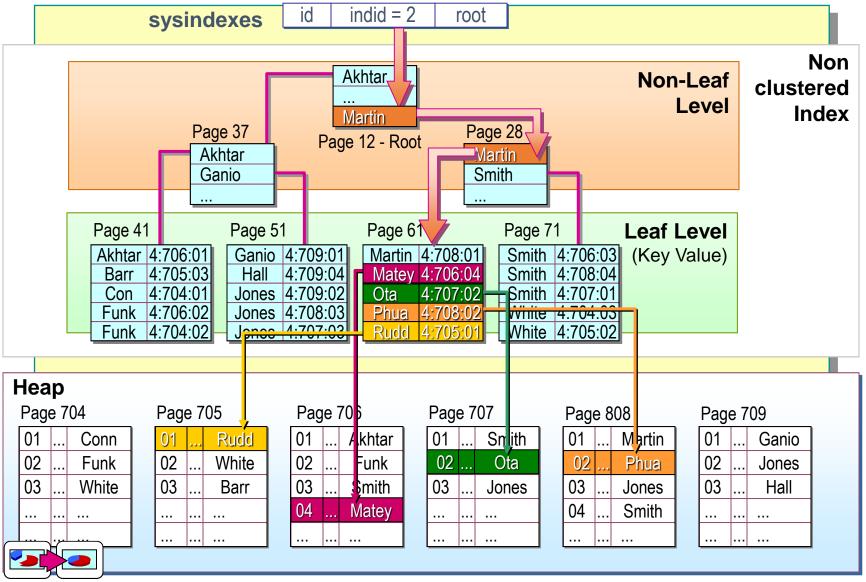
indid	Object Type			
0	Неар			
1	Clustered Index			
2 to 250	Nonclustered Index			
255	text, ntext, or image			

- Location of IAM, First, and Root of Indexes
- Number of Pages and Rows
- Distribution of Data

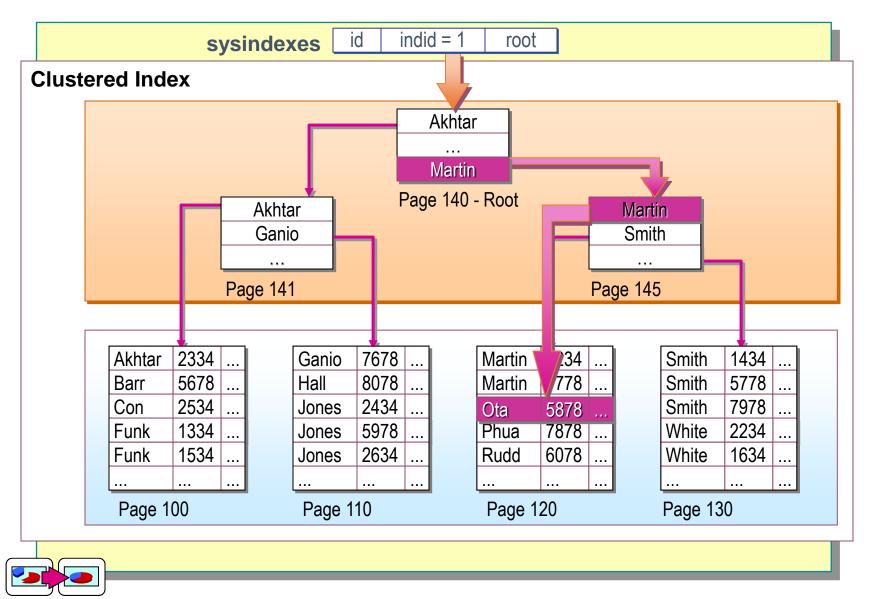
Finding Rows Without Indexes



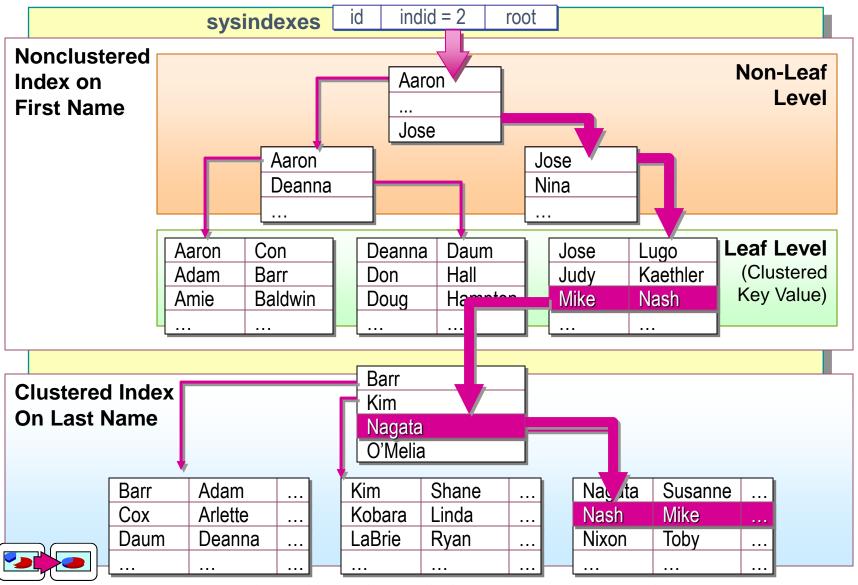
Finding Rows in a Heap with a Nonclustered Index



Finding Rows in a Clustered Index



Finding Rows in a Clustered Index with a Nonclustered Index



How SQL Server Maintains Index and Heap Structures

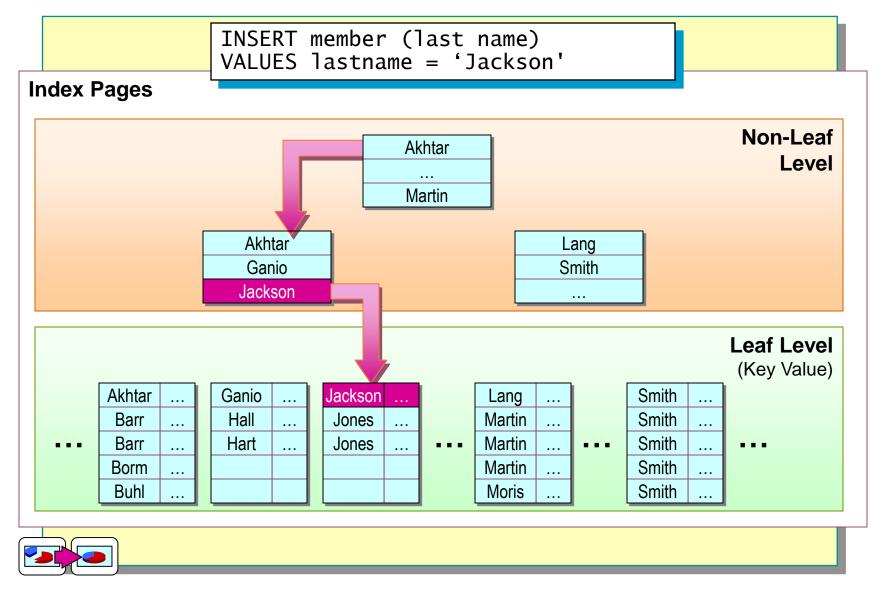
Page Splits in an Index

Forwarding Pointer in a Heap

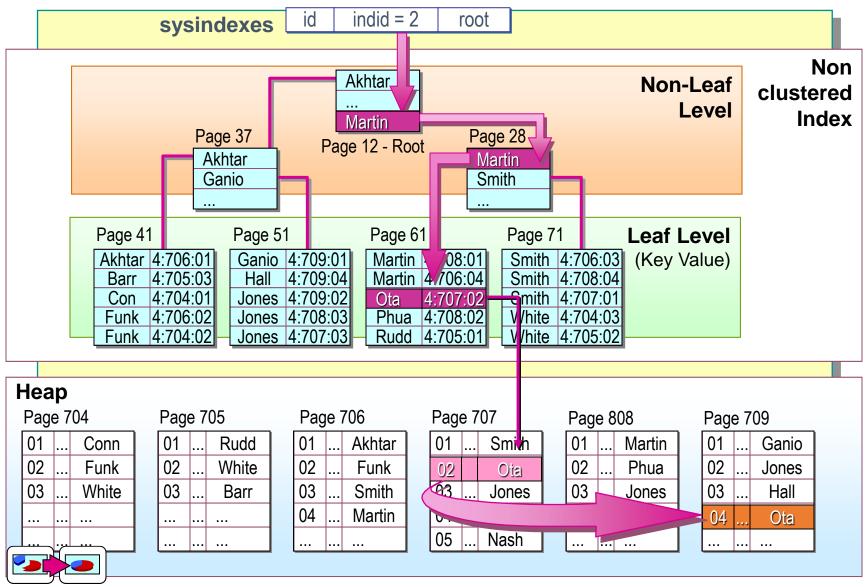
How SQL Server Updates Rows

How SQL Server Deletes Rows

Page Splits in an Index



Forwarding Pointer in a Heap



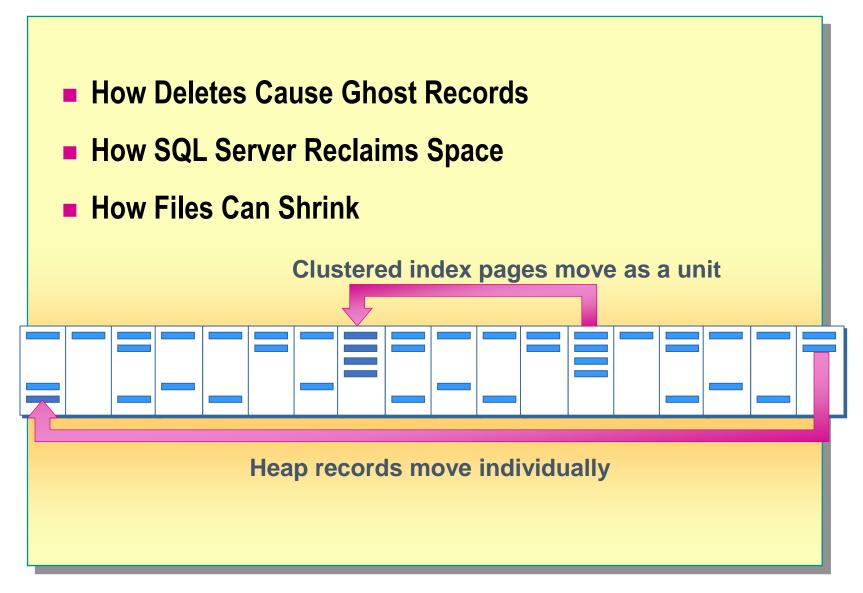
How SQL Server Updates Rows

An Update Generally Does Not Cause a Row to Move

An Update Can Be a Delete Followed by an Insert

Batch Updates Touch Each Index Only Once

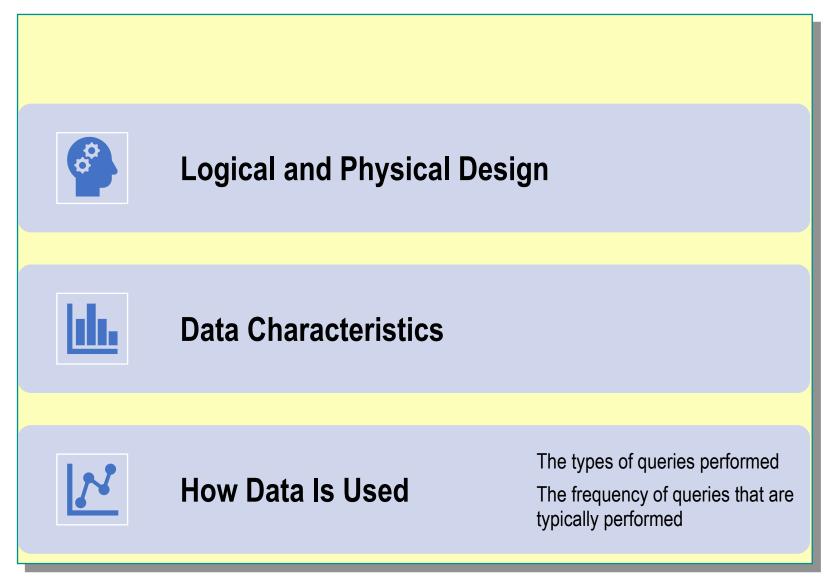
How SQL Server Deletes Rows



Deciding Which Columns to Index

- Understanding the Data
- Indexing Guidelines
- Choosing the Appropriate Clustered Index
- Indexing to Support Queries
- Determining Selectivity
- Determining Density
- Determining Distribution of Data

Understanding the Data



Indexing Guidelines

Columns to Index

- Primary and foreign keys
- Those frequently searched in ranges
- Those frequently accessed in sorted order
- Those frequently grouped together during aggregation

Columns Not to Index

- Those seldom referenced in queries
- Those that contain few unique values
- Those defined with text, ntext, or image data types

Choosing the Appropriate Clustered Index

Heavily Updated Tables

 A clustered index with an identity column keeps updated pages in memory

Sorting

 A clustered index keeps the data presorted

Column Length and Data Type

- Limit the number of columns
- Reduce the number of characters
- Use the smallest data type possible

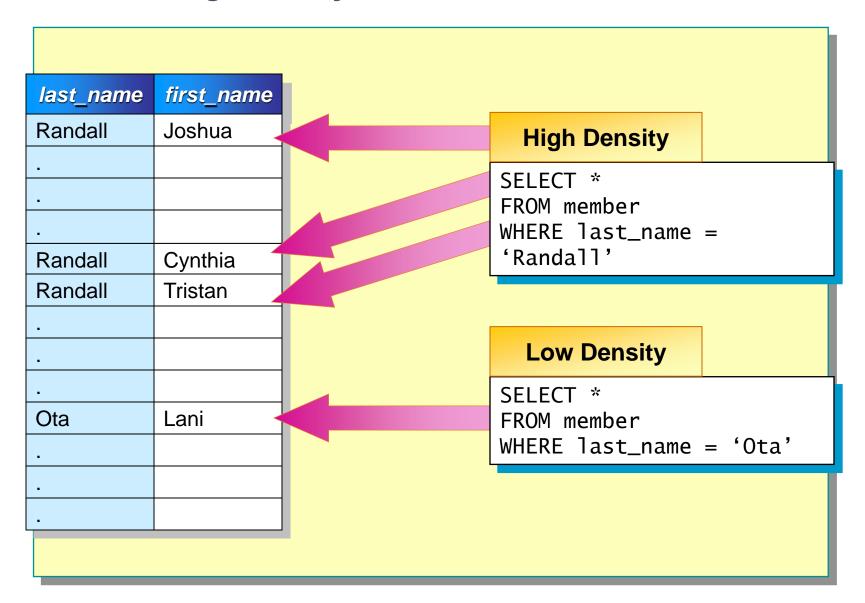


Determining Selectivity

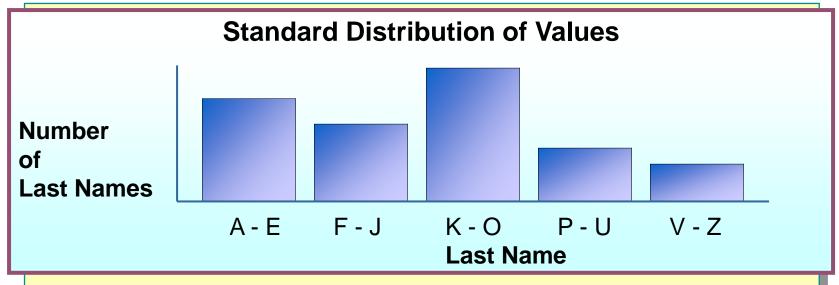
member_no	last_name	first_name			High selec	tivity
1	Randall	Joshua	Number of rows mee		= 1000	· = 10%
2	Flood	Kathie				1070
		SELECT * FROM mem	ber			
-		WHERE me	mber_no > 8999			
10000	Anderson	Bill				

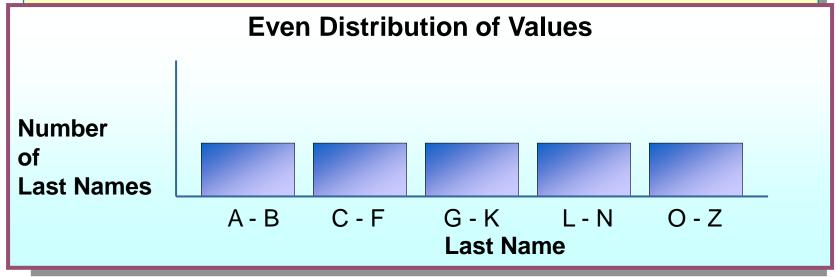
member_no	last_name	first_name			Low select	ivity
1	Randall	Joshua Number of rows meeting cri			9000	= 90%
2	Flood	Kathie	Total number of rows	in table	= 10000	. 30 /0
•		SELECT *				
•		FROM mem	ber mber_no < 9001			
10000	Anderson	Bill				

Determining Density



Determining Distribution of Data





Recommended Practices



Create Indexes on Columns That Join Tables











Lab A: Determining the Indexes of a Table



Introduction to Indexes

Index Architecture

Review

How SQL Server Retrieves Stored Data

How SQL Server Maintains Index and Heap Structures

Deciding Which Columns to Index

Module References

- Clustered and nonclustered indexes described https://docs.microsoft.com/en-us/sql/relational-databases/indexes/clustered-and-nonclustered-indexes-described?view=sql-server-ver15
- Create Index in SQL Server https://docs.microsoft.com/en-us/sql/t-sql/statements/create-index-transact-sql?f1url=%3FappId%3DDev15IDEF1%26l%3DEN-US%26k%3Dk(create_index_TSQL);k(sql13.swb.tsqlresults.f1);k(sql13.swb.tsqlquery.f1);k(MiscellaneousFilesProject);k(DevLang-TSQL)%26rd%3Dtrue&view=sql-server-ver15
 - Oracle Database Indexes https://docs.oracle.com/cd/A91034_01/DOC/server.901/a90192/ch02.htm#10055
 - Create Index Statement in MySQL https://dev.mysql.com/doc/refman/8.0/en/create-index.html