

TABLE INDEXES AND VIEW

Instructor:



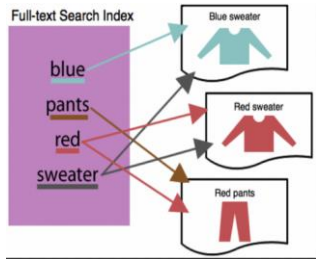
Learning Goals

By the end of this lecture
students should be able to:

✓ Create Indexes to improve query retrieval speed

✓ Automatically generate sequence numbers by using a sequence generator

✓ Create, maintain, and use View



EmployeeID	PersonID	FirstName	LastName
1	1	Eralper	YILMAZ
2	3	Robert	Vieira
3	607	Guy	Gilbert
4	608	Kevin	Brown
5	609	Roberto	Tamburello
6	610	Rob	Walters
7	611	Thierry	D'Hers
8	612	David	Bradley
9	613	JoLynn	Dobney
10	614	Ruth	Ellerbrock
11	615	Gail	Erickson
12	616	Barry	Johnson
13	617	Josief	Goldberg
14	618	Terri	Duffy
15	619	Sidney	Higa

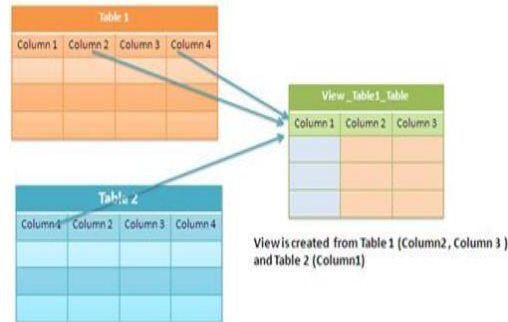


Table of contents

- Table Indexes
- View
- Naming Convention And Styles

Section1

TABLE INDEXES

Why use indexes?

- An **index** in database is similar to an index in a book
- **Indexes** in database help speed up search queries. Allow find data in a table without scanning the entire table.

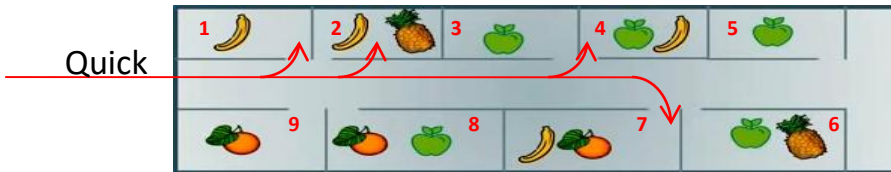
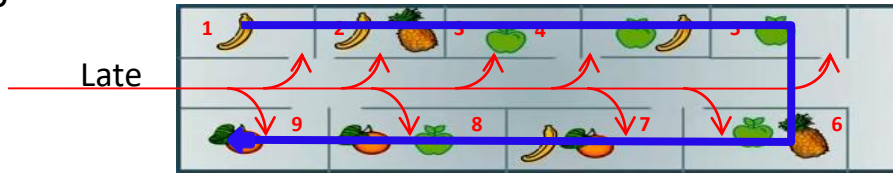


Table Indexes (1/3)

```
CREATE TABLE dbo.PhoneBook (  
    LastName      varchar(50) NOT NULL,  
    FirstName     varchar(50) NOT NULL,  
    PhoneNumber   varchar(50) NOT NULL  
);
```

```
SELECT PhoneNumber  
FROM   dbo.PhoneBook  
WHERE  LastName = 'Logan' AND FirstName = 'Todd';
```

Alexander, Mary
344-555-0133
Kurtz, Jeffrey
452-555-0179
Vessa, Robert
560-555-0171
Thames, Judy
799-555-0118

Martinez, Frank
171-555-0147
Haines, Betty
867-555-0114
Burnett, Linda
121-555-0121
Harris, Keith
170-555-0127

Kitt, Sandra
303-555-0117
Brewer, Alan
494-555-0134
Campbell, Frank
491-555-0132
Logan, Todd
783-555-0110

...

Clayton, Jane
206-555-0195
Johnson, Brian
320-555-0134
Liu, David
440-555-0132
Diaz, Brenda
147-555-0192

Table Indexes (2/3)

```
CREATE TABLE dbo.PhoneBook (  
    LastName        varchar(50) NOT NULL,  
    FirstName        varchar(50) NOT NULL,  
    PhoneNumber      varchar(50) NOT NULL  
);
```

Result:
783-555-0110

```
SELECT PhoneNumber  
FROM   dbo.PhoneBook  
WHERE  LastName = 'Logan' AND FirstName = 'Todd';
```

Alexander, Mary 344-555-0133	Martinez, Frank 171-555-0147	Kitt, Sandra 303-555-0117	Clayton, Jane 206-555-0195
Kurtz, Jeffrey 452-555-0179	Haines, Betty 867-555-0114	Brewer, Alan 494-555-0134	Johnson, Brian 320-555-0134
Vessa, Robert 560-555-0171	Burnett, Linda 121-555-0121	Campbell, Frank 491-555-0132	Liu, David 440-555-0132
Thames, Judy 799-555-0118	Harris, Keith 170-555-0127	Logan, Todd 783-555-0110	Diaz, Brenda 147-555-0192

■ There are 2 types of major Indexes:

✓ ***Clustered***

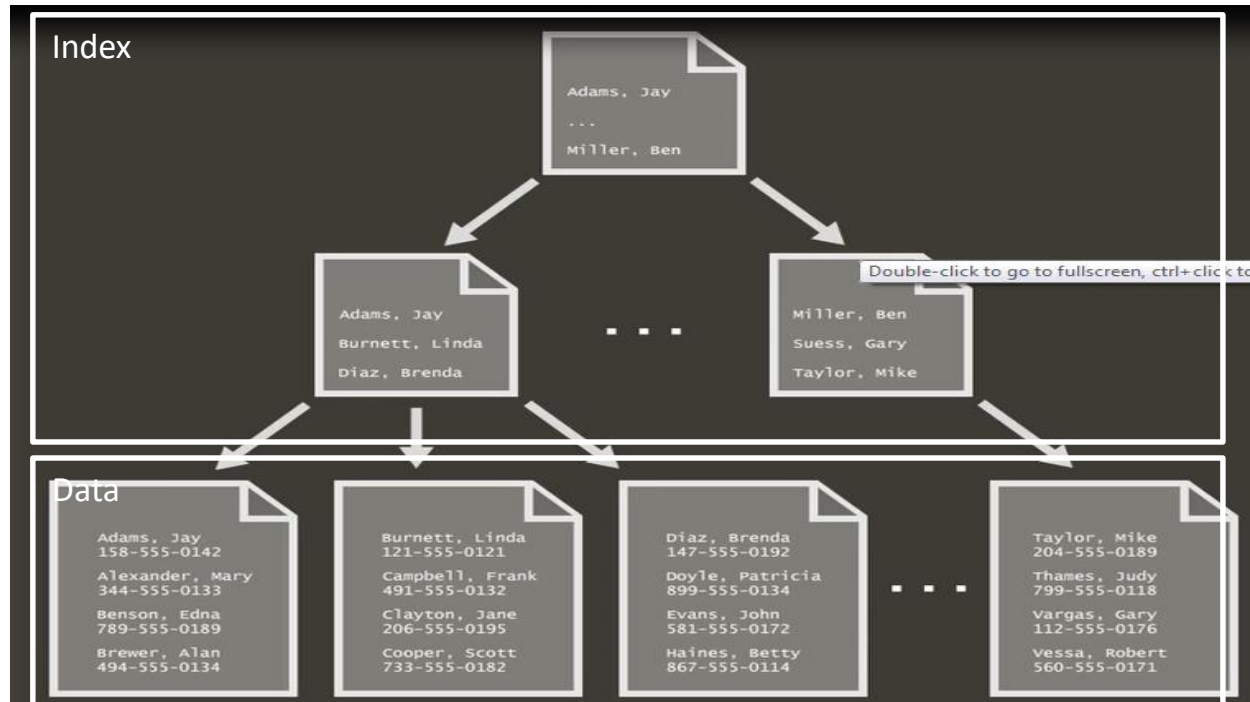
- Data is stored in the order on the clustered index
- Only 1 clustered index per table
- Usually the Primary Key
- Sort and store the data rows in the table based on their key value.

✓ ***Non-clustered***

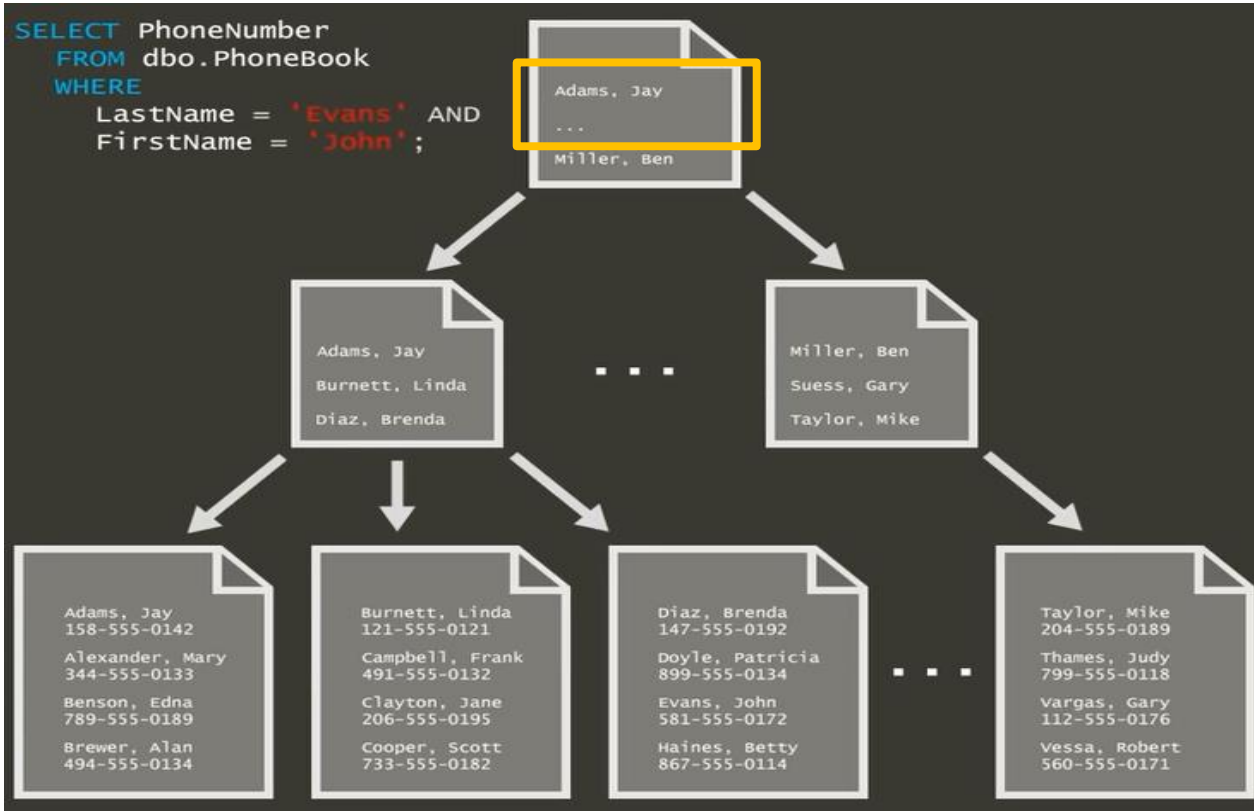
- Data is not stored in the order on the non clustered index
- Have a structure completely separate from the data rows.

Clustered Index

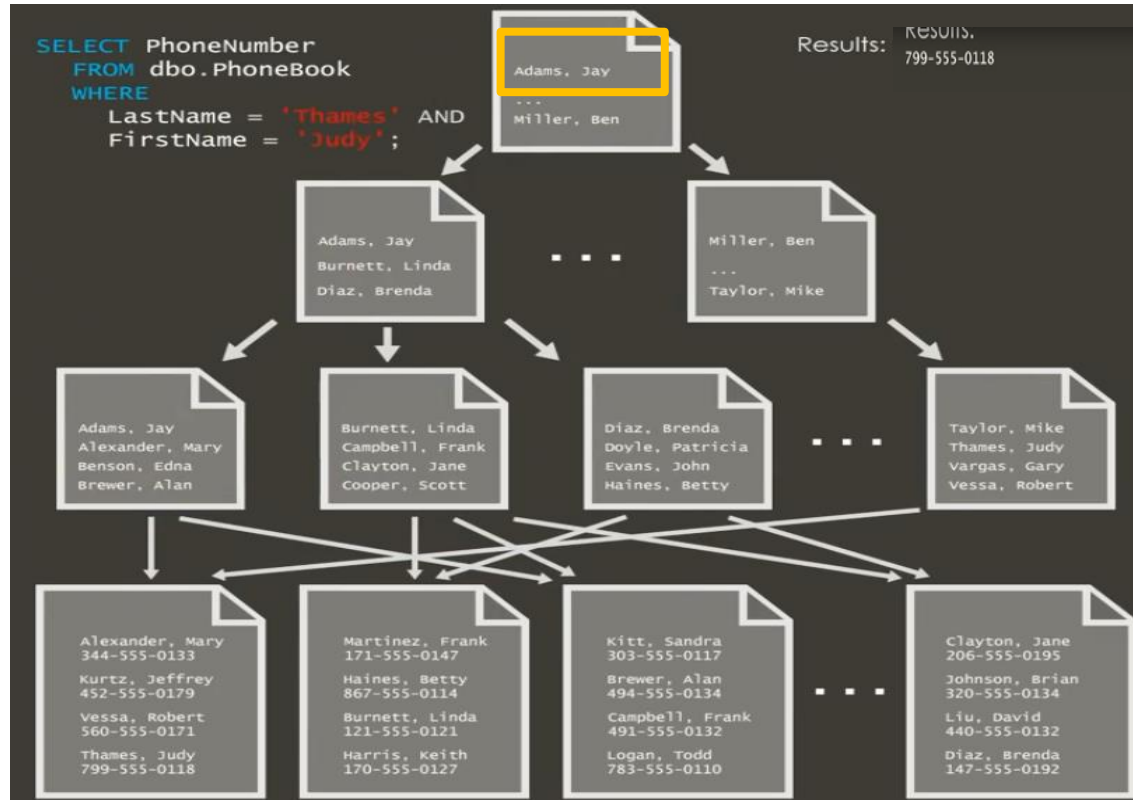
```
CREATE CLUSTERED INDEX IX_PhoneBook_CI  
ON dbo.PhoneBook (LastName, FirstName)
```



Clustered Index



Non - Clustered Index



Creating an Index

- Create a new index:

```
CREATE INDEX index_name  
ON table_name (column1_name, column2_name, ...)
```

- Deleting an Index

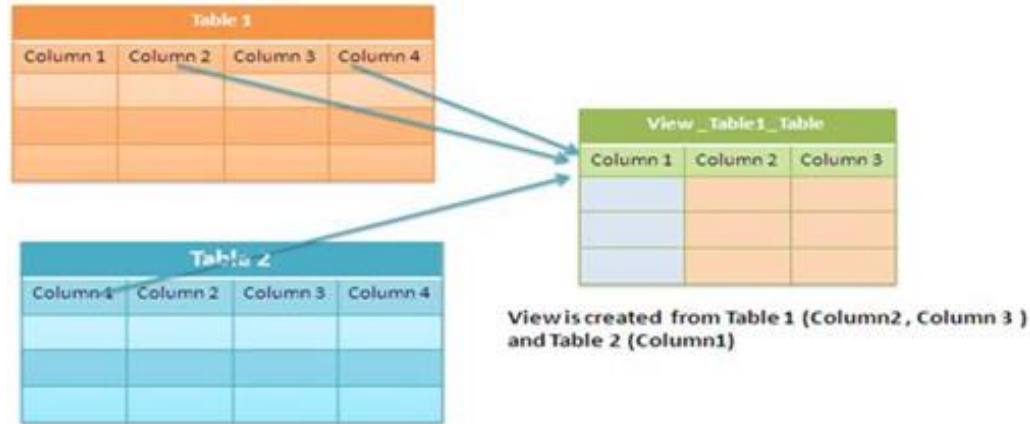
```
DROP INDEX table_name.index_name
```

Section2

VIEWS

What is a view?

- A **View** is a logical or virtual table. The fields in a view are fields from one or more real tables in the database.
- There are **two major reasons** you might want to use views:
 - ✓ Views allow you to limit the data users can access
 - ✓ Views reduce complexity for end users.



Creating a view

```
CREATE VIEW View_Name [list of column names]
AS
SELECT...
```

Example:

```
CREATE VIEW view_EmployeeByDpt
AS
SELECT ID, NAME, AGE, DEPT_NAME
FROM EMP, DEPARTMENT
WHERE EMP.DEPT_ID = DEPARTMENT.DEPT_ID
```

```
SELECT * FROM view_EmployeeByDpt
```

Table: EMP

ID	NAME	AGE	DEP_ID
1	John	25	3
2	Mike	30	2
3	Parm	25	1
4	Todd	23	4
5	Sara	35	1
6	Ben	40	3

Table: DEPARTMENT

DEPT_ID	DEPT_NAME
1	IT
2	Payroll
3	HR
4	Admin

view_EmployeeByDpt

ID	NAME	AGE	DEPT_NAME
1	John	25	HR
2	Mike	30	Payroll
3	Parm	25	IT
4	Todd	23	Admin
5	Sara	35	IT
6	Ben	40	HR

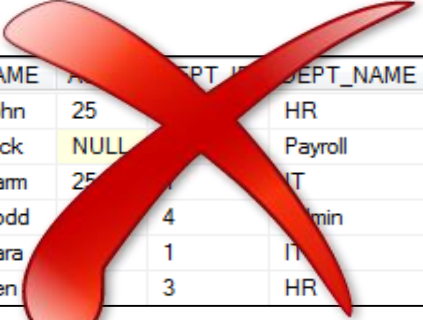
Deleting a view

- Syntax:

```
DROP VIEW View_Name
```

- Example:

```
DROP VIEW view_EmployeeByDpt
```



ID	NAME	AGE	DEPT_ID	DEPT_NAME
1	John	25	1	HR
2	Nick	NULL	2	Payroll
3	Pam	25	3	IT
4	Todd	4	4	Admin
5	Sara	1	5	IT
6	Ben	3	3	HR

view_EmployeeByDpt

Section3

NAMING CONVENTION AND STYLES

1. Use UPPER CASE for all T-SQL constructs, excepts Types

Correct:

```
SELECT MAX([Salary]) FROM dbo.[EmployeeSalary]
```

Incorrect:

```
SELECT max([Salary]) from dbo.[EmployeeSalary]
```

2. Use lower case for all T-SQL Types and Usernames

Correct:

```
DECLARE @MaxValue int
```

Incorrect:

```
DECLARE @MaxValue INT
```

3. Use Pascal casing for all UDO's

Correct:

```
CREATE TABLE dbo.EmployeeSalary  
(  
    EmployeeSalaryID    INT  
)
```

Incorrect:

```
CREATE TABLE dbo.EmployeeSalary  
(  
    EmployeesalaryID    int  
)
```

4. Avoid abbreviations and single character names

Correct:

```
DECLARE @RecordCount int
```

Incorrect:

```
DECLARE @Rc int
```

5. UDO naming must confer to the following regular expression ([a-zA-Z][a-zA-Z0-9]).

Do not use any special or language dependent characters to name objects. Constraints can use the underscore character.

Correct:

```
CREATE TABLE dbo.[EmployeeSalary]
```

Incorrect:

```
CREATE TABLE dbo.[Employee Salary]
```

6. Use the following prefixes when naming objects

usp_: User stored procedures

ufn_: User defined functions

view_: Views

IX_: Indexes

usp_: User stored procedures

DF_: Default constraints

PK_: Primary Key constraints

FK_: Foreign Key constraints

CHK_: Check constraints

UNI_: Unique constraints

Correct:

```
CREATE PROCEDURE dbo.usp_EmployeeSelectAll
```

Incorrect:

```
CREATE PROCEDURE dbo.EmployeeSelectRetired --without  
prefixed
```

7. Name tables in the singular form

Correct:

```
CREATE TABLE dbo.[Employee]
```

Incorrect:

```
CREATE TABLE dbo.[Employees]
```

8. Tables that map one-to many, many-to-many relationships should be named by concatenating the names of the tables in question, starting with the most central table's name.

Correct:

```
CREATE TABLE dbo.[EmployeeSalary]
```

✓ Table Indexes

- ✓ Why use indexes?
- ✓ Create, maintain and use index

✓ View

- ✓ Create, maintain and use view

✓ Naming convention

✓ Demo

- ✓ View



Thank you

