



- Seminar 5 -

Indecsi

- este o structură on-disk asociată unui tabel sau unui view care crește viteza de returnare a înreg.
- sunt def. pt. a localiza mai rapid înregistrările care urmează să fie returnate
- sunt organizate ca **B-trees**

Caracteristici:

- clustered / nonclustered
- unique / non-unique
- single-column / multi-column
- ordine descrescătoare / crescătoare
- full table / filtered pt. indecși non-clustered

Clustered

- def. ordinea fizică în care sunt stocate date
- se poate defini doar un singur index clustered pe fiecare tabel, deoarece nu putem stoca fizic date decât într-o singură ordine
- paginile date de index vor conține întotdeauna toate coloanele din tabel
- sintaxă:

```
CREATE CLUSTERED INDEX index_nume  
ON table_nume (column_nume(s) [ASC | DESC]);
```
- un index key poate cuprinde maxim 900 de bytes

Non-clustered

- stochează pointeri la date din heap / indexul clustered ca parte din index key
- putem avea mai mulți indexi nonclustered
- sintaxă:

```
CREATE INDEX index_nume ON table_nume  
(column_nume(s) [ASC | DESC]);
```
- SQL Server suportă până la 999 indexi nonclustered pe tabel
- un index key poate fi format din maxim 32 de coloane și un index poate cuprinde maxim 1700 bytes

- când este creată o cheie primară pe tabel, dacă nu există deja un index clustered și un index non-clustered nu este specificat \Rightarrow index clustered unique

\hookrightarrow dacă există deja index clustered \Rightarrow index nonclustered unique

\hookrightarrow dacă toate coloanele returnate de către o interogare se află în index, indexul se numește **covering index**, iar interogarea se numește **covered query**

Index clustered

- poate fi folosit pt. interogările care se execută în mod frecvent
- poate fi folosit în **range queries**
- nu este bine să-l definim pe coloane actualizate des, deoarece va trebui să modifice constant ordinea fizică a datelor

Exemplu:

```
CREATE UNIQUE CLUSTERED INDEX ix_produse_cod_produs_asc ON Produse(cod_produs ASC);
```

Index nonclustered

Exemplu:

```
CREATE NONCLUSTERED INDEX ix_atracții_nume_asc ON Atracții(nume ASC);  
SAU  
CREATE INDEX ix_atracții_nume_asc ON Atracții(nume);
```

- atunci când este creată o constrângere **UNIQUE** pe un tabel, se va crea un index nonclustered unique pe coloana sau coloane pe care este definită constrângerea **UNIQUE**

Indecii: coloane *key* și coloane *nonkey*.

o coloane *key* - coloanele specificate la crearea unui index

o coloane *nonkey* - coloanele specificate în *clausa INCLUDE* a unui index *nonclustered*

- sintaxa:

```
CREATE INDEX index_nume ON table_nume (key_column_nume(s) (ASC|DESC))  
INCLUDE (nonkey_column_nume(s));
```

Exemplu:

```
CREATE INDEX ix_Persoane - nume - desc - prenume - asc ON Persoane (nume ASC, prenume ASC)  
INCLUDE (pseudonim, localitate);
```

Beneficii coloane *nonkey*:

o coloanele sunt asociate cu un index *scan*

o tipurile de date care nu sunt permise în coloanele care fac parte din index (index *key columns*) sunt permise în coloanele *nonkey* (exceptând tipurile de date *text*, *ntext* și *image*)

o coloanele care apar în *clausa INCLUDE* nu se iau în calcul în cazul limitei de 1700 bytes a unui index *key*

Indecii *unique* vs. indecii *non-unique*

- dacă vom crea un index *nonclustered* unic în tabelul *Categorii* pe coloanele *nume*, nu vom putea avea două înregistrări cu aceeași valoare pt. coloana *nume* în tabel

- operațiunea de a crea un index unic eșuează, dacă în tabel avem deja valori duplicate în coloane sau coloanele pe care am definit indexul unic.

Exemplu:

```
CREATE UNIQUE INDEX ix_Categorii - nume - desc - asc ON Categorii  
(nume DESC);
```

```
CREATE UNIQUE INDEX ix_Persoane - nume - desc - prenume - asc - asc ON  
Persoane (nume ASC, prenume ASC);
```

- putem seta opțiunea *IGNORE_DUP_KEY* în cazul indexilor unici

↳ în cazul unui batch *INSERT*, se vor insera toate cele care nu au valori duplicate → și cele care conțin astfel de valori, vor fi ignorate și nu vor fi inserate

Exemplu:

```
CREATE UNIQUE INDEX ix_Vizitatori - email - asc - asc  
ON Vizitatori (email ASC) WITH (IGNORE_DUP_KEY = ON);
```

Indecii *single column* vs. indecii *multi-column*

o index *single column* este un *key* def. pe o singură coloană (care conține o singură coloană *key* în index *key*)

Exemplu:

```
CREATE INDEX ix_Lecturi - nume - desc ON Lecturi (nume DESC); (index nonc. non-unique  
single column)
```

```
SELECT nume FROM Lecturi ORDER BY nume DESC;  
SELECT nume FROM Lecturi ORDER BY nume ASC;
```

```
CREATE INDEX ix_Attractii - vîrto - min - asc - nume - asc ON Attractii  
(vîrto - min ASC, nume ASC);
```

```
SELECT vîrto - min, nume FROM Attractii ORDER BY vîrto - min ASC, nume ASC;  
SELECT vîrto - min, nume FROM Attractii ORDER BY vîrto - min DESC, nume DESC;  
SELECT vîrto - min, nume FROM Attractii ORDER BY vîrto - min ASC;  
SELECT vîrto - min, nume FROM Attractii ORDER BY vîrto - min DESC;
```

ASC, DESC → nu poate fi folosit pt. sortare
DESC, ASC

Full table vs. filtered pt. indecizii nonclustered

- indecizii nonclustered full table conțin toate valorile coloanei sau coloanelor pe care au fost def.
- indecizii nonclustered filtered conțin doar acele valori pt. care evaluează condiția specificată la crearea indecizului returnează True

Exemplu: `CREATE INDEX IX_Atracții-nume_asc-filtered ON Atracții
(nume ASC) WHERE nume > 'C';`

`SELECT nume, descriere FROM Atracții WHERE nume > 'C';`

`SELECT nume, descriere FROM Atracții WHERE nume > 'E';`

nu putem folosi pt. < 'C' sau fără clauza WHERE!

Indecizii pt. DELETE

- un indice definit pe foreign key va putea fi folosit pt. a șterge înregistrările dependente mult mai rapid

Modificarea unui indice

- dacă dorim să ștergem sau să adăugăm coloane într-un indice ne trebuie să ștergem și să creștem din nou indicele

- dacă dorim să deactivăm un indice sau să setăm anumite opțiuni, putem folosi instrucțiunea

ALTER INDEX

Exemplu:

`ALTER INDEX IX_Atracții-vârsto_min_asc_nume_asc ON Atracții DISABLE; (de deactivare)`

`ALTER INDEX IX_Atracții-vârsto_min_asc_nume_asc ON Atracții REBUILD; (de reactivare)`

Ștergerea unui indice

`DROP INDEX IX_Atracții-vârsto_min_asc_nume_asc ON Atracții;`

Recomandări

- tipuri de coloane recomandate ca indecizii key columns:

- constrângere foreign key
- care apar în clauza WHERE
- care apar în clauza ORDER BY
- pe baza cărora se fac join-uri
- care apar în clauza GROUP BY
- cu grad mare de varietate a valorilor