



Indecsi



Indecsi

- ▶ Un index este o structură on-disk asociată unui tabel sau unui view
- ▶ Un index crește viteza de regasire a înregistrărilor care urmează să fie returnate
- ▶ Alegerea corectă a indecsilor afectează performanța
- ▶ Dacă nu este definit un index, SQL Server verifică fiecare înregistrare din tabel pentru a determina dacă ea conține sau nu informația necesară interogării (table scan)



Indecsi - Clasificare

1. Clustered versus nonclustered
2. Unique versus non-unique
3. Single column versus multi-column
4. Ordine crescătoare sau descrescătoare pe coloanele din index
5. Full-table versus filtered pentru indecsii nonclustered



Indecsi – clustered vs nonclustered

- ▶ **Clustered index** – definește ordinea fizică în care sunt stocate datele în tabel (dacă indexul clustered conține mai multe coloane, datele vor fi stocate în ordine secvențială în funcție de coloane: prima coloană, a doua coloană și așa mai departe)
- ▶ Fiecare tabel are asociat cel mult un index clustered

- ▶ **Sintaxa:**

```
CREATE CLUSTERED INDEX index_name ON  
table_name(column_name(s) [ASC|DESC])
```

- ▶ **Data pages** ale unui index clustered vor conține întotdeauna toate coloanele din tabel



Indecsi – clustered vs nonclustered

- ▶ Clustered index poate fi folosit pentru interogările care se execută în mod frecvent
- ▶ Nu este bine să definim un index clustered pe coloane care sunt actualizate des, deoarece SQL Server va trebui să modifice constant ordinea fizică a datelor
- ▶ Exemplu de definire a unui index clustered unique:

```
CREATE UNIQUE CLUSTERED INDEX  
Publicatie_id_publicatie_asc ON  
Publicatie(id_publicatie ASC)
```



Indecsi – clustered vs non-clustered

- ▶ **Nonclustered index** – spre deosebire de un index clustered, stochează pointeri la date din heap/clustered index ca parte din index key
- ▶ Putem avea mai multi indecsi nonclustered pe acelasi tabel
- ▶ Sintaxa:

```
CREATE INDEX Index_Name ON  
TableName(column_name(s) [ASC|DESC])
```
- ▶ (SQL Server suportă până la **999** indecsi nonclustered pe tabel)



Indecsi – clustered vs non-clustered

- ▶ Atunci când se definește o constrângere UNIQUE pe un tabel, se va crea un index nonclustered unique pe coloana sau coloanele pe care este definită constrângerea UNIQUE
- ▶ Exemplu de definire a unui index nonclustered non unique:

```
CREATE NONCLUSTERED INDEX  
Publicatie_titlu_asc ON Publicatie (titlu ASC)
```

Sau

```
CREATE INDEX Publicatie_titlu_asc ON  
Publicatie(titlu)
```



Indecsi – clustered vs non-clustered

- ▶ Când este creată o cheie primară pe un tabel, dacă nu este definit deja un index clustered si un index nonclustered nu este specificat, se creează un index clustered unique
- ▶ Dacă atunci când este creată o cheie primară pe un tabel este deja definit un index clustered, se va crea un index unique nonclustered
- ▶ Dacă toate coloanele returnate de catre o interogare se afla în index, indexul se numește *covering index*, iar interogarea se numeste *covered query*



Indecsi – key vs non-key index columns

- ▶ Key columns – coloanele specificate la crearea unui index
- ▶ Non-key columns – coloanele specificate în clauza INCLUDE a unui index nonclustered

- ▶ Sintaxa:

```
CREATE INDEX index_name  
ON table_name(key_column_name(s) [ASC|DESC])  
INCLUDE nonkey_column_name(s))
```

- ▶ Exemplu:

```
CREATE INDEX Publicatie_titlu_asc_autorp_asc  
ON Publicatie(titlu ASC, autorp ASC) INCLUDE  
(abstract)
```



Indecsi – key vs non-key index columns

- ▶ **Beneficii ale utilizării coloanelor non-key:**
 - ▶ Coloanele pot fi accesate cu un index scan
 - ▶ Tipurile de date care nu sunt permise în coloanele care fac parte din index (index key columns) sunt permise în coloanele non-key (exceptând tipurile de date text, ntext și image)
 - ▶ Pot fi incluse coloane calculate, dar valorile trebuie să fie deterministe
 - ▶ Coloanele care apar în clauza INCLUDE nu se iau în calcul în cazul limitei de 900 bytes a unui index impusă de SQL Server



Indecși - unique versus indecși non-unique

- ▶ Un index unique (unic) definit pe una sau mai multe coloane asigură unicitatea valorilor la nivelul coloanei sau combinației de coloane pe care este definit
- ▶ De exemplu, dacă vom crea un index nonclustered unic în tabelul *Categorie* pe coloana *nume*, nu vom putea avea două înregistrări cu aceeași valoare pentru coloana *nume* în tabel
- ▶ Dacă vom crea un index nonclustered unic în tabelul *Publicatie* pe coloanele *titlu* și *autor*, nu vom putea avea două înregistrări cu aceleași valori pentru coloanele *titlu* și *autor* în tabel
- ▶ Dacă vom crea un index unic după ce am introdus date în tabel și avem valori duplicate în coloana sau coloanele pe care am definit indexul unic, operațiunea de creare a indexului va esua



Indecși - unique versus indecși non-unique

- ▶ Pentru a putea crea un index unic pe un tabel, va trebui să eliminăm înainte toate valorile duplicate din coloana sau coloanele pe care definim indexul unic
- ▶ Un index unic garantează că fiecare valoare (inclusiv NULL) pentru coloana pe care a fost definit apare o singură dată în tabel
- ▶ Exemplu de index nonclustered unique definit pe o singură coloană:

```
CREATE UNIQUE INDEX Categorie_num_desc_uq  
ON Categorie nume DESC)
```
- ▶ Exemplu de index nonclustered unique definit pe mai multe coloane:

```
CREATE UNIQUE INDEX  
Publicatie_titlu_asc_autor_asc_uq  
ON Publicatie(titlu ASC, autor ASC)
```



Indecși single column versus indecși multi-column

- ▶ Un index *single column* este un index definit pe o singură coloană (care conține o singură coloană key în index key)
- ▶ Un index *multi-column* este un index definit pe mai multe coloane (care conține mai multe coloane key în index key)
- ▶ Dacă dorim să folosim indexul și pentru sortarea înregistrărilor care sunt returnate, va trebui să ținem cont de ordinea crescătoare sau descrescătoare a coloanelor care fac parte din index key
- ▶ În cazul unui index single-column, ordinea specificată pentru coloana key nu este atât de importantă deoarece se poate folosi indexul pentru a sorta după coloana respectivă atât în ordine crescătoare cât și în ordine descrescătoare



Indecși single column versus indecși multi-column

- ▶ Exemplu de definire a unui index nonclustered non unique single column:

```
CREATE INDEX Publicatie_titlu_desc ON  
Publicatie(titlu DESC)
```

- ▶ Indexul definit mai sus va putea fi folosit pentru ambele interogări de mai jos:

```
SELECT titlu FROM Publicatie ORDER BY titlu DESC  
SELECT titlu FROM Publicatie ORDER BY titlu ASC
```

- ▶ În cazul indecsilor multi-column, ordinea coloanelor key este importantă dacă dorim să se utilizeze indexul pentru sortarea înregistrărilor după coloanele care fac parte din index key



Indecși single column versus indecși multi-column

- ▶ Exemplu de definire a unui index nonclustered non unique multi-column:

```
CREATE INDEX Publicatie_titlu_asc_id_cat_asc ON Publicatie  
(titlu ASC, id_cat ASC)
```

- ▶ Indexul definit mai sus va putea fi folosit pentru sortare în cazul interogărilor de mai jos:

```
SELECT titlu, id_cat FROM Publicatie ORDER BY titlu ASC, id_cat ASC  
SELECT titlu, id_cat FROM Publicatie ORDER BY titlu DESC, id_cat DESC  
SELECT titlu, id_cat FROM Publicatie ORDER BY titlu ASC  
SELECT titlu, id_cat FROM Publicatie ORDER BY titlu DESC
```



Indecși single column versus indecși multi-column

- ▶ **Indexul** `Publicatie_titlu_asc_id_cat_asc` **definit anterior nu va putea fi folosit pentru sortare în cazul interogărilor de mai jos:**

```
SELECT titlu, id_cat FROM Publicatie ORDER BY titlu ASC, id_cat DESC  
SELECT titlu, id_cat FROM Publicatie ORDER BY titlu DESC, id_cat ASC
```



Full-table versus filtered pentru indecșii nonclustered

- ▶ Indecșii nonclustered *full-table* conțin toate valorile coloanei sau coloanelor pe care au fost definiți
- ▶ Indecșii nonclustered *filtered* conțin doar acele valori pentru care evaluarea condiției specificate la crearea indexului returnează true
- ▶ Exemplu de definire a unui index nonclustered non unique single column filtered:

```
CREATE INDEX IX_Publicatie_titlu_asc_filtered  
ON Publicatie (titlu ASC) WHERE titlu > 'C'
```



Full-table versus filtered pentru indecșii nonclustered

- ▶ **Indexul** `Publicatie_titlu_asc_filtered` **va putea fi folosit pentru următoarele interogări:**

```
SELECT titlu FROM Publicatie WHERE titlu > 'C'
```

```
SELECT titlu FROM Publicatie WHERE titlu > 'E'
```

- ▶ **Indexul** `Publicatie_titlu_asc_filtered` **nu va putea fi folosit pentru următoarele interogări:**

```
SELECT titlu FROM Publicatie WHERE titlu > 'B'
```

```
SELECT titlu FROM Publicatie
```



Indecși – Modificarea unui index

- ▶ Dacă dorim să ștergem sau să adăugăm coloane într-un index va trebui să ștergem și să creăm din nou indexul
- ▶ Dacă dorim să dezactivăm un index sau să setăm anumite opțiuni, putem folosi instrucțiunea **ALTER INDEX**
- ▶ Exemplu de dezactivare a unui index:

```
ALTER INDEX Publicatie_titlu_desc ON Publicatie DISABLE
```

- ▶ Exemplu de reactivare a unui index dezactivat:

```
ALTER INDEX ublicatie_titlu_desc ON Publicatie REBUILD
```



Indecși - ștergerea unui index

- ▶ În anumite situații, un index devine inutil și trebuie eliminat
- ▶ Sintaxa pentru ștergerea unui index:

```
DROP INDEX index_name ON table_name
```

- ▶ Exemplu de ștergere a unui index:

```
DROP INDEX Publicatie_titlu_desc ON Publicatie
```



Indecsi - Concluzii

- ▶ Indecsi sunt utili pentru mărirea performanței operațiilor de citire, dar scad performanța operațiilor de scriere
- ▶ Tipuri de coloane recomandate ca index key columns:
 - foreign keys
 - coloane care apar în clauza WHERE a interogărilor
 - coloane care apar în clauza ORDER BY a interogărilor
 - coloane pe baza cărora se fac JOIN-uri
 - coloane care apar în clauza GROUP BY a interogărilor
 - coloane cu grad mare de varietate a valorilor



Indecsi - Concluzii

- ▶ Înțelegerea caracteristicilor celor mai frecvente interogări
- ▶ Înțelegerea caracteristicilor coloanelor care sunt folosite în interogări
- ▶ Determinarea locației de stocare optimă pentru index



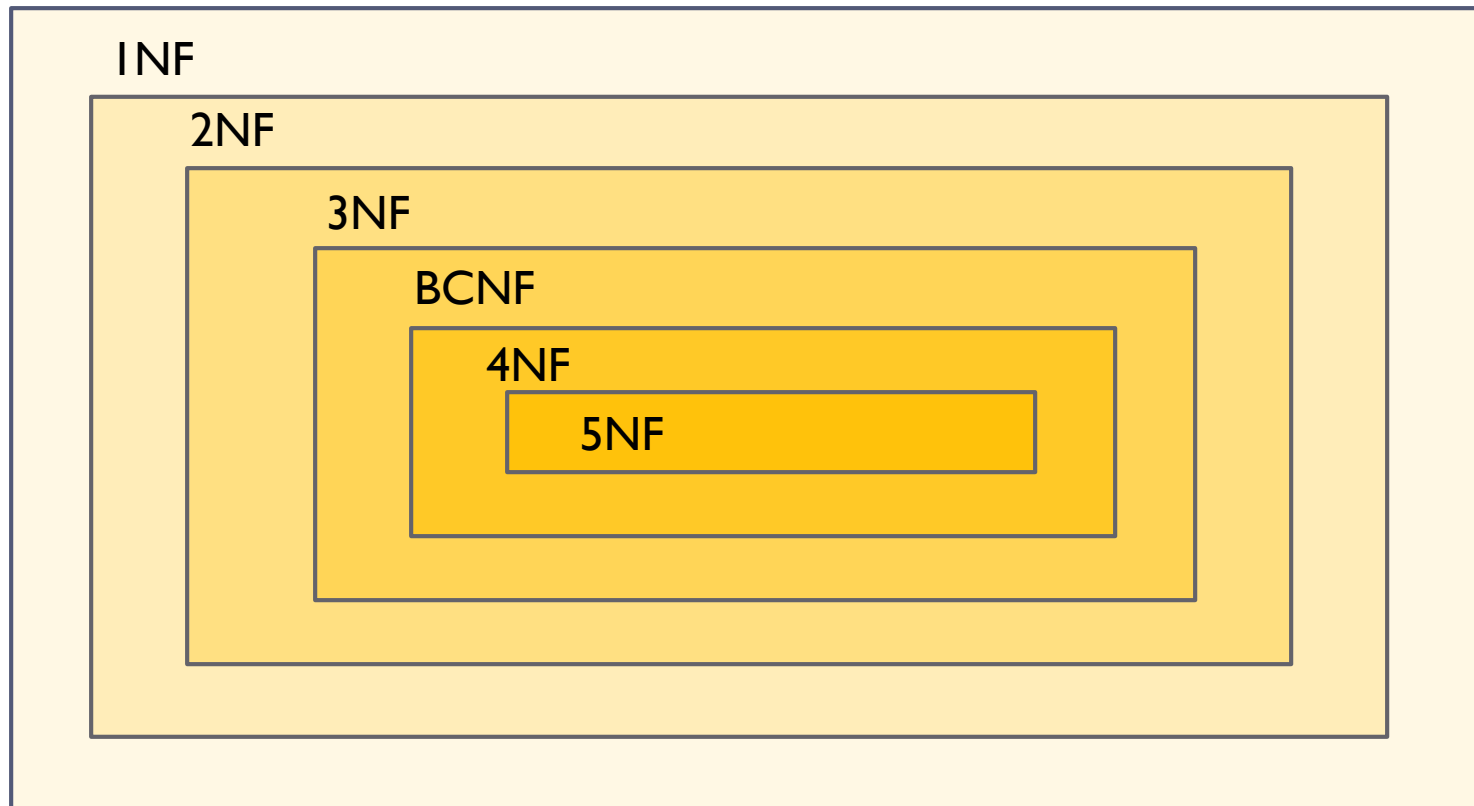


Formele normale

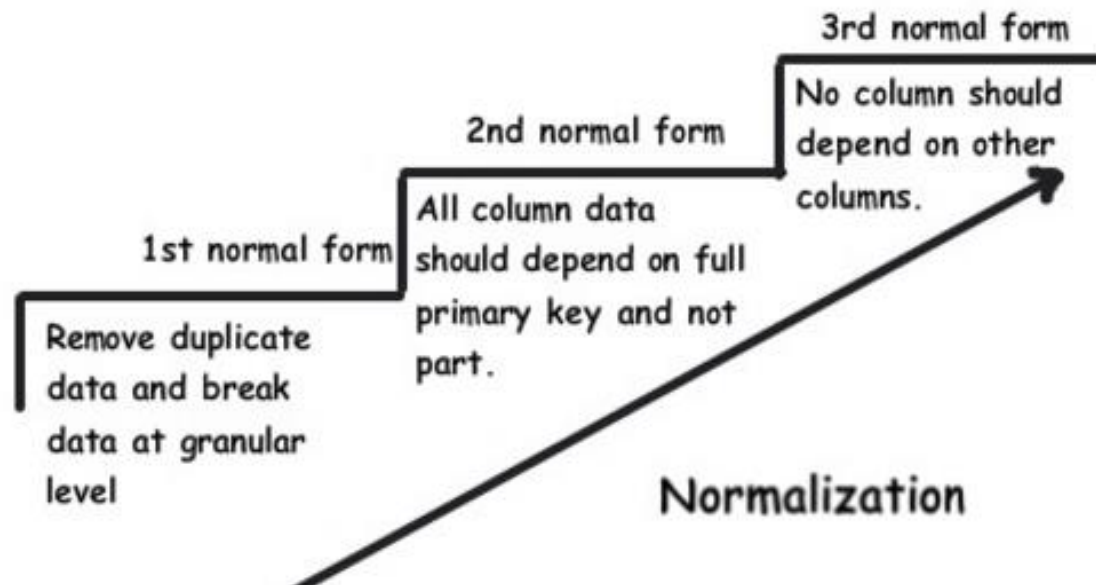


Recapitulare

- Incluziuni pentru relatiile aflate in diferite forme normale

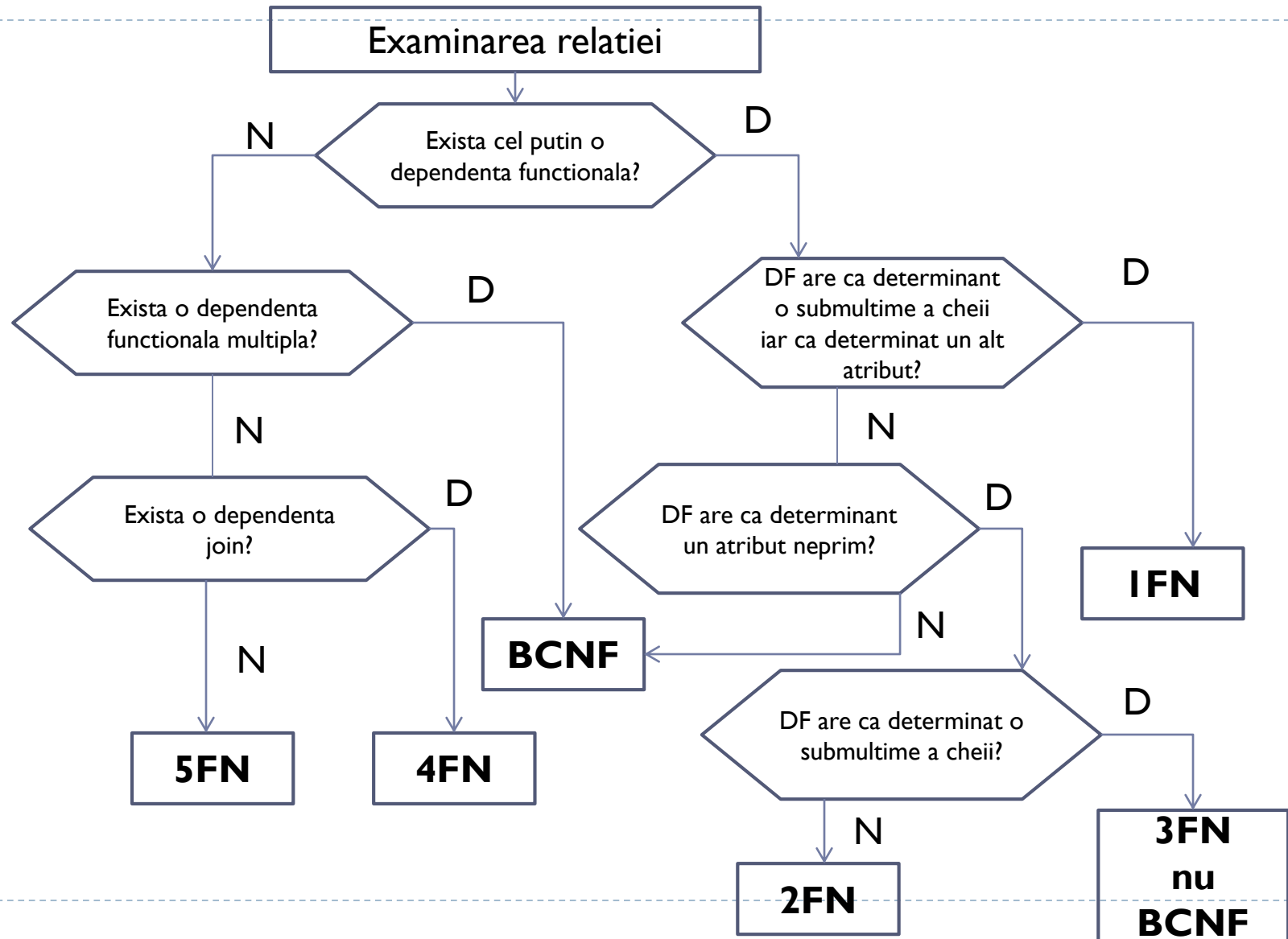


Recapitulare



Sursa: <http://www.xpode.com/ShowArticle.aspx?ArticleId=578>

Recapitulare - Algoritm de identificare a formeii normale a unei relatii



Exercitii

Gasiti formele normale ale relatiilor urmatoare

- ▶ R1 (n°carte, nume carte, n°colectie, nume colectie)
- ▶ R2 (CNP, nume autor, n°autor, adresa autor)
- ▶ R3 (n°carte, n°colectie, nb ex. vandute)
- ▶ R4 (n°carte, nume colectie, cantitate drepturi autor)
- ▶ R5 (n°autor, n°colectie, nb ex. vandute)

