

# Proiectarea Bazelor de Date

**Cornelia TUDORIE**



# Proiectarea Bazelor de Date - Cuprins

---

- A. Modelul Relațional.**
- B. Limbajul SQL. Lucrul cu tabele.**
- C. Bazele teoretice ale limbajelor relaționale**
- D. Sisteme de baze de date. Performanța.**
- E. Obiecte ale sistemului de baze de date. Performanța în utilizare**
- F. Programe pentru baze de date. Performanța în programare.**
- G. Protecția datelor.**
- H. Sisteme Informatice. Proiectarea sistemelor de baze de date.**

## **E. Obiecte ale sistemului de baze de date. Performanța în utilizare.**

---

- 1. Stocarea informațiilor în baza de date**
- 2. Obiectele bazei de date:**
  - 1. Tabele**
  - 2. Indecși**
  - 3. Tabele virtuale**
  - 4. Sinonime**
  - 5. Secvențe**
- 3. Gestiunea eficientă a spațiului de stocare**
- 4. Optimizarea cererilor**

## E. 2. 2. Obiectele bazei de date: Indecși

***Index*** – structură ordonată care permite:

- creșterea performanțelor ( localizarea directă, rapidă, a unui rând al unei tablele)
- asigurarea integrității datelor (unicitatea valorilor)

**Operații:**

- creare
- ștergere
- ( actualizarea automată la operații de manipulare a datelor )



# Clasificare indecși

## Clasificare din punct de vedere logic:

- după o coloană (*single column*)  
sau mai multe coloane (*concatenated index*)
- unic (*Unique*) sau neunic (*nonUnique*)

## Clasificare din punct de vedere fizic:

- *partiționat* sau *nepartiționat*
- de tip arbore binar (*B-tree*) sau *bitmap*

# Crearea indecșilor

- **Automat**

**Un index unique este creat automat când se definește o restricție de integritate de tip PRIMARY KEY sau UNIQUE.**

- **Manual**

**Utilizatorul poate crea indecși nonunique pentru anumite coloane pentru a micșora timpul de acces la rânduri.**

# Utilitatea indecșilor

**Indecșii se creează atunci când:**

- **Coloana (coloanele) este (sunt) frecvent folosită(e) în clauze WHERE sau în condiții de join.**
- **Coloana conține o diversitate mare de valori.**
- **Coloana conține un număr mare de null-uri.**
- **Tabela este mare și cererile întorc uzual cel mult 2-4% dintre rânduri.**

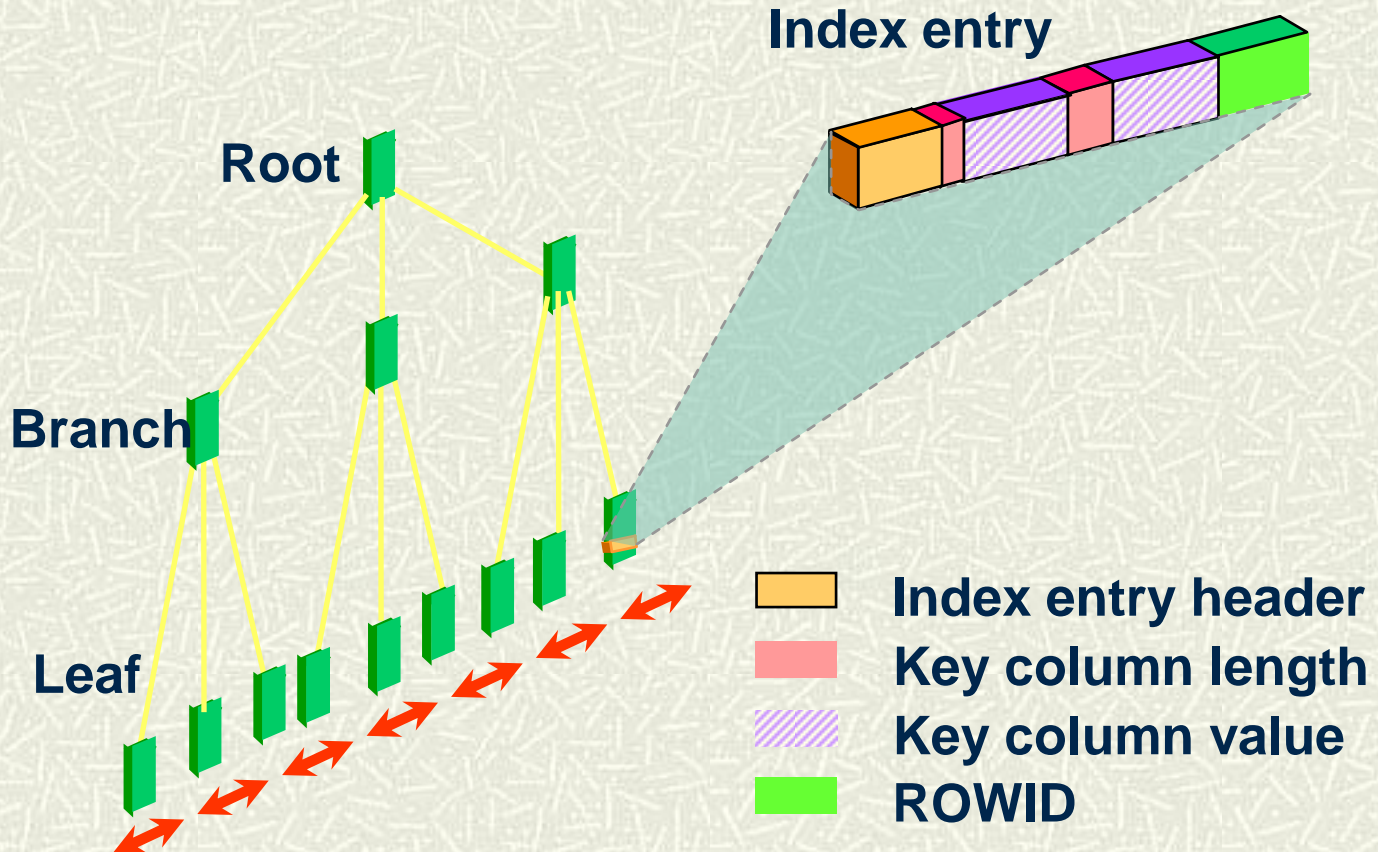
# Utilitatea indecșilor

**Indecșii NU se creează atunci când:**

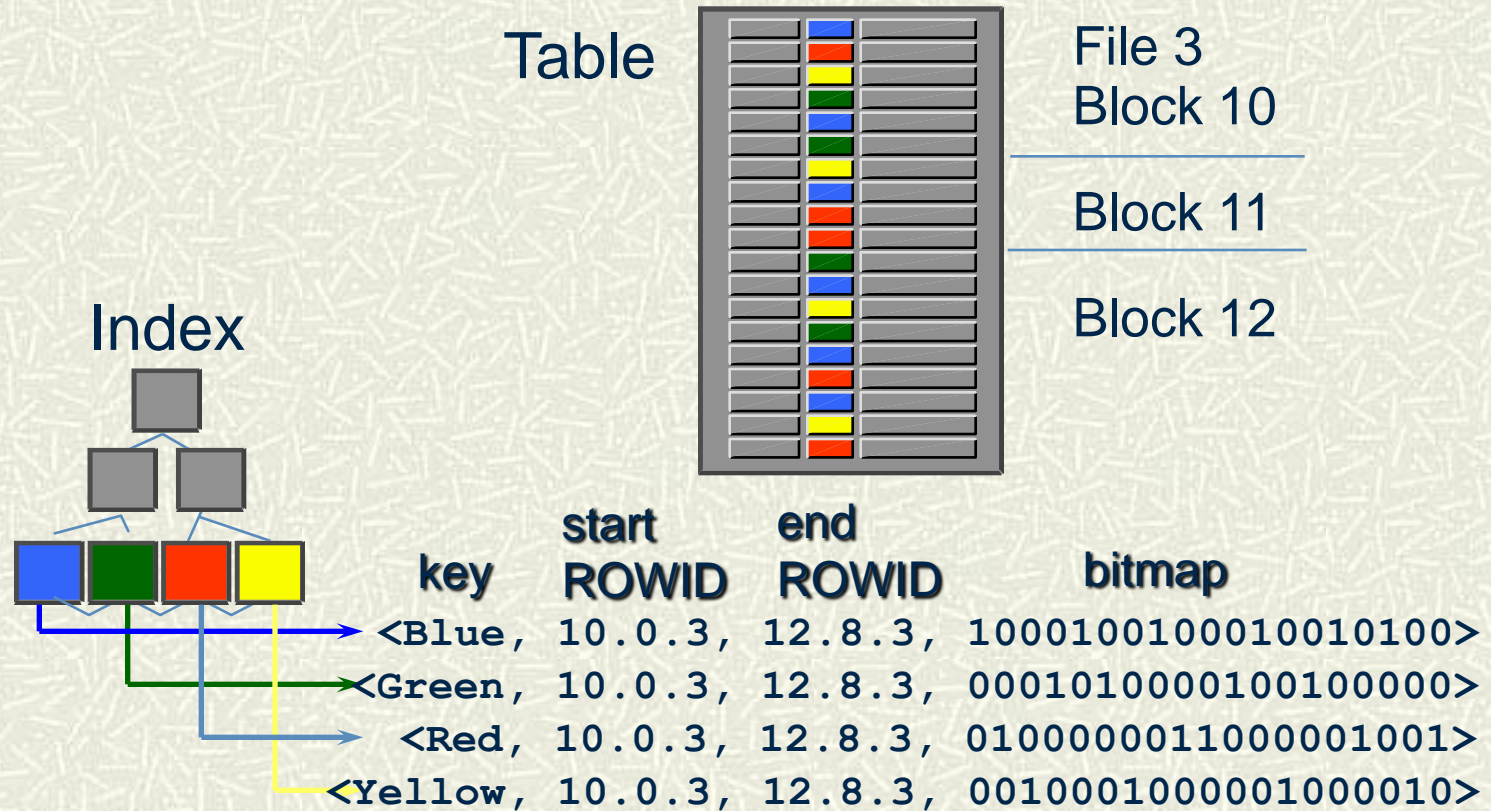
- **Tabela este mică**
- **Coloanele nu sunt des folosite în interogări.**
- **Cele mai multe cereri întorc mai mult de 2–4% dintre rânduri.**
- **Tabela este actualizată frecvent.**



# Index B-Tree



# Index Bitmap



# Indecși B-Tree și Bitmap (comparație)

<b>B-tree</b>	<b>Bitmap</b>
<b>Pentru coloanele cu cardinalitate mare</b>	<b>Pentru coloanele cu cardinalitate mică</b>
<b>Actualizarea cheilor este relativ costisitoare</b>	<b>Actualizarea cheilor este foarte costisitoare</b>
<b>Ineficient pentru operații logice</b>	<b>Eficient pentru operații logice</b>

# Crearea unui index B-Tree

```
CREATE [ UNIQUE ] INDEX [schema.] index
  ON [schema.] tabela
    ( coloană [ ASC | DESC ] [, coloană [ ASC | DESC ] , ... ] )
[TABLESPACE tablespace ]
[PCTFREE integer ]
[STORAGE ( INITIAL integer [ { K | M } ]
          NEXT integer [ { K | M } ]
          PCTINCREASE integer
          MINEXTENTS integer
          MAXEXTENTS integer ) ] ;
```



# Crearea unui index B-Tree

## Exemplu

```
CREATE INDEX emp_lname_idx  
ON employees(last_name)  
PCTFREE 30  
STORAGE (INITIAL 200K NEXT 200K  
PCTINCREASE 0 MAXEXTENTS 50)  
TABLESPACE indx01;
```

# Crearea unui index Bitmap

```
CREATE BITMAP INDEX [schema.] index
  ON [schema.] tabela
    ( coloană [ ASC | DESC ] [, coloană [ ASC | DESC ] , ... ] )
[TABLESPACE tablespace ]
[PCTFREE integer ]
[STORAGE ( INITIAL integer [ { K | M } ]
             NEXT integer [ { K | M } ]
             PCTINCREASE integer
             MINEXTENTS integer
             MAXEXTENTS integer ) ] ;
```

# Crearea unui index Bitmap

## Exemplu

```
CREATE BITMAP INDEX ord_region_id_idx  
ON ord(region_id)  
PCTFREE 30  
STORAGE (INITIAL 200K NEXT 200K  
PCTINCREASE 0 MAXEXTENTS 50)  
TABLESPACE indx01;
```

# Modificarea parametrilor de stocare

```
ALTER INDEX [schema.] index  
[STORAGE ( INITIAL integer [ K | M ]  
             NEXT integer [ K | M ]  
             PCTINCREASE integer  
             MINEXTENTS integer  
             MAXEXTENTS integer ) ] ;
```

```
ALTER INDEX emp_lname_idx  
STORAGE (NEXT 400K  
MAXEXTENTS 100) ;
```



# Refacerea indecșilor

```
ALTER INDEX [schema.]index REBUILD  
  [TABLESPACE tablespace ]  
  [PCTFREE integer ]  
  [STORAGE ( INITIAL integer [ { K | M } ]  
              NEXT integer [ { K | M } ]  
              PCTINCREASE integer  
              MINEXTENTS integer  
              MAXEXTENTS integer ) ] ;
```

# Obținerea de informații asupra indecșilor

---

## Dicționarul de date

**Dicționarul de date furnizează informații despre :  
indecși și coloanele ce compun indecșii**

**Disponibile pentru utilizator :  
user\_indexes  
user\_ind\_columns**

# Obținerea de informații asupra indecșilor

## Exemplu

```
SQL> describe user_ind_columns
```

Name	Null?	Type
-----	-----	----
INDEX_NAME	NOT NULL	VARCHAR2(30)
TABLE_NAME	NOT NULL	VARCHAR2(30)
COLUMN_NAME		VARCHAR2(4000)
COLUMN_POSITION	NOT NULL	NUMBER
COLUMN_LENGTH	NOT NULL	NUMBER

```
SQL> select * from user_ind_columns where table_name='PROD';
```

INDEX_NAME	TABLE_NAME	COLUMN_NAME	COLUMN_POSITION	COLUMN_LENGTH
-----	-----	-----	-----	-----
PROD_PK	PROD	CODP	1	22

# Obținerea de informații asupra indecșilor

## Comanda ANALYZE

```
ANALYZE INDEX [schema.] tabela  
VALIDATE STRUCTURE ;
```

**Rezultatul:            în INDEX\_STATS**



# Ștergerea unui index

---

```
DROP INDEX [schema.]index ;
```

## **E. Obiecte ale sistemului de baze de date. Performanța în utilizare.**

---

- 1. Stocarea informațiilor în baza de date**
- 2. Obiectele bazei de date:**
  - 1. Tabele**
  - 2. Indecși**
  - 3. Tabele virtuale**
  - 4. Sinonime**
  - 5. Secvențe**
- 3. Gestiunea eficientă a spațiului de stocare**
- 4. Optimizarea cererilor**

## E. 2. 3. Obiectele bazei de date: Tabele virtuale

***Tabelă virtuală*** – structură logică cu date conținute în baza de date care permite:

- restricționarea accesului la baza de date
- simplificarea interogărilor
- realizarea independenței logice a datelor (schema externă)
- prezentarea acelorași date sub forme diferite

### **Operații:**

- creare
- interogare
- actualizare ( numai în anumite situații )
- ștergere

## **E. Obiecte ale sistemului de baze de date. Performanța în utilizare.**

---

- 1. Stocarea informațiilor în baza de date**
- 2. Obiectele bazei de date:**
  - 1. Tabele**
  - 2. Indecși**
  - 3. Tabele virtuale**
  - 4. Sinonime**
  - 5. Secvențe**
- 3. Gestiunea eficientă a spațiului de stocare**
- 4. Optimizarea cererilor**



## E. 2. 4. Obiectele bazei de date: Sinonime

---

***Sinonim*** – Un alt nume dat unui obiect pentru simplificarea accesului, mai ales:

- pentru referirea la tabelele din alte scheme
- pentru scurtarea numelor lungi

### **Operații:**

- creare
- ștergere

## **E. Obiecte ale sistemului de baze de date. Performanța în utilizare.**

---

- 1. Stocarea informațiilor în baza de date**
- 2. Obiectele bazei de date:**
  - 1. Tabele**
  - 2. Indecși**
  - 3. Tabele virtuale**
  - 4. Sinonime**
  - 5. Secvențe**
- 3. Gestiunea eficientă a spațiului de stocare**
- 4. Optimizarea cererilor**

## E. 2. 5. Obiectele bazei de date: Secvențe

***Secvență*** – Generator automat de valori numerice unice

- Utilizat mai ales pentru generarea de valori pentru chei primare
- Obiect partajabil
- Înlocuiește secvențe de cod
- Mărește performanțele atunci când valori generate sunt deja memorate în cache

### **Operații:**

- creare
- invocare
- ștergere

## **E. Obiecte ale sistemului de baze de date. Performanța în utilizare.**

---

- 1. Stocarea informațiilor în baza de date**
- 2. Obiectele bazei de date:**
  - 1. Tabele**
  - 2. Indecși**
  - 3. Tabele virtuale**
  - 4. Sinonime**
  - 5. Secvențe**
- 3. Gestiunea eficientă a spațiului de stocare**
- 4. Optimizarea cererilor**