

# **Baze de Date II**

## **(Proiectarea Sistemelor cu Baze de Date )**

**Cornelia TUDORIE**



# Proiectarea Sistemelor cu Baze de Date - Cuprins

---

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

## C. Sisteme cu baze de date. Performanța.

---

### *Sistem informatic :*

- model al unui sistem real
- conține reprezentări informatice ale obiectelor și proceselor reale
- pentru a urmări evoluția sistemului real, a lua decizii,..
- facilitează controlul și coordonarea sistemului real

## C. Sisteme cu baze de date. Performanța.

---

*Sistem cu baze de date :*

- sistem integrat de date memorate și organizate sub un control centralizat

*integrare:* structuri de date intercorelate

*control centralizat :* SGBD, cu ajutorul schemei



## C. Sisteme cu baze de date. Performanța.

---

### *Schema :*

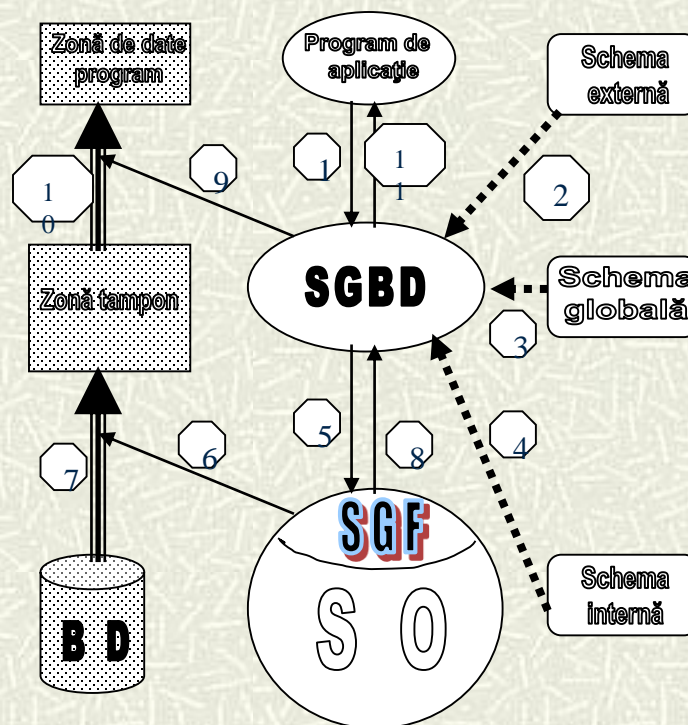
- descrierea datelor, obiectelor și a accesului la ele

### *SGBD :*

- ansamblu coordonat de programe ce permite descrierea, memorarea, manipularea, prelucrarea ansamblurilor de date ce constituie baze de date

# Rolul SGBD

Date      Programe      Model



## Legendă

- informații de control
- ⇒ date
- .....→ consultare scheme

# Avantaje

---

- ❑ *coerență* (se reduce redundanța sau este strict controlată + restricții de integritate)
- ❑ *independența datelor față de programe*
- ❑ *accesibilitate* (limbaje de interogare de nivel înalt)
- ❑ *partajarea datelor*
- ❑ *protecția datelor*

# Dezavantaje

- ❑ *costuri* (hardware, software, instruire, personal)
- ❑ *probleme de securitate și integritate* (din partajarea datelor)
- ❑ *dificultatea modelării datelor*



# Obiective

## *Performanța la nivel de:*

- ☐ **utilizare**
- ☐ **programare**
- ☐ **proiectare**
- ☐ **administrare**

## *Se urmărește:*

- ☐ **minimizarea timpului de răspuns**
- ☐ **gestiunea optimă a spațiului de stocare**
- ☐ **asigurarea integrității datelor**

# Proiectarea Sistemelor cu Baze de Date - Cuprins

---

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

## **D. 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**

# D. 1. Stocarea informațiilor în baza de date

---

## Structura Bazei de Date

**Nivel conceptual**       $\Longrightarrow$       **Diagrama entitate-legătură**

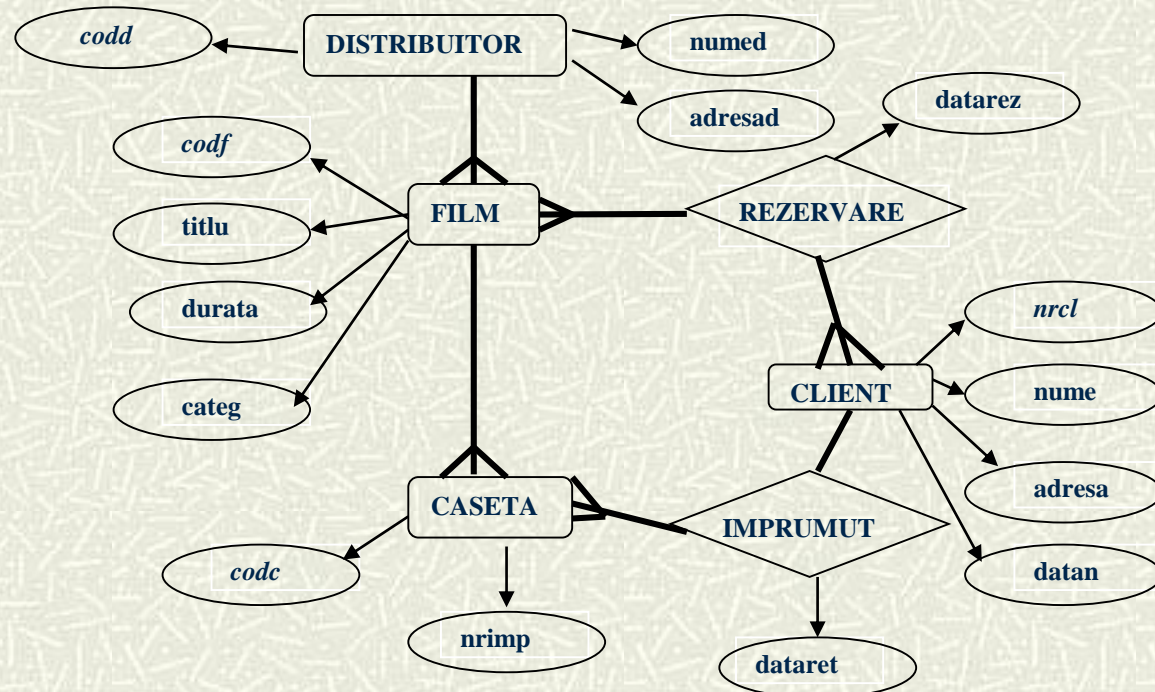
**Nivel logic**       $\Longrightarrow$       **Tabele, ... și alte obiecte logice**

**Nivel fizic**       $\Longrightarrow$       **Obiecte fizice (fișiere)**



# D. 1. Stocarea informațiilor în baza de date

## Structura conceptuală a Bazei de Date



# D. 1. Stocarea informațiilor în baza de date

## Structura logică a Bazei de Date

**DISTRIBUTOR** [ codd, numed, adresad ]

**FILM** [ codf, titlu, durata, categ, codd ]

**CASETA** [ codc, codf, nrimp ]

**CLIENT** [ nrcl, nume, adresa, datan ]

**IMPRUMUT** [ codc, nrcl, dataret ]

**REZERVARE** [ nrcl, codf, datarez ]  
- - - - -

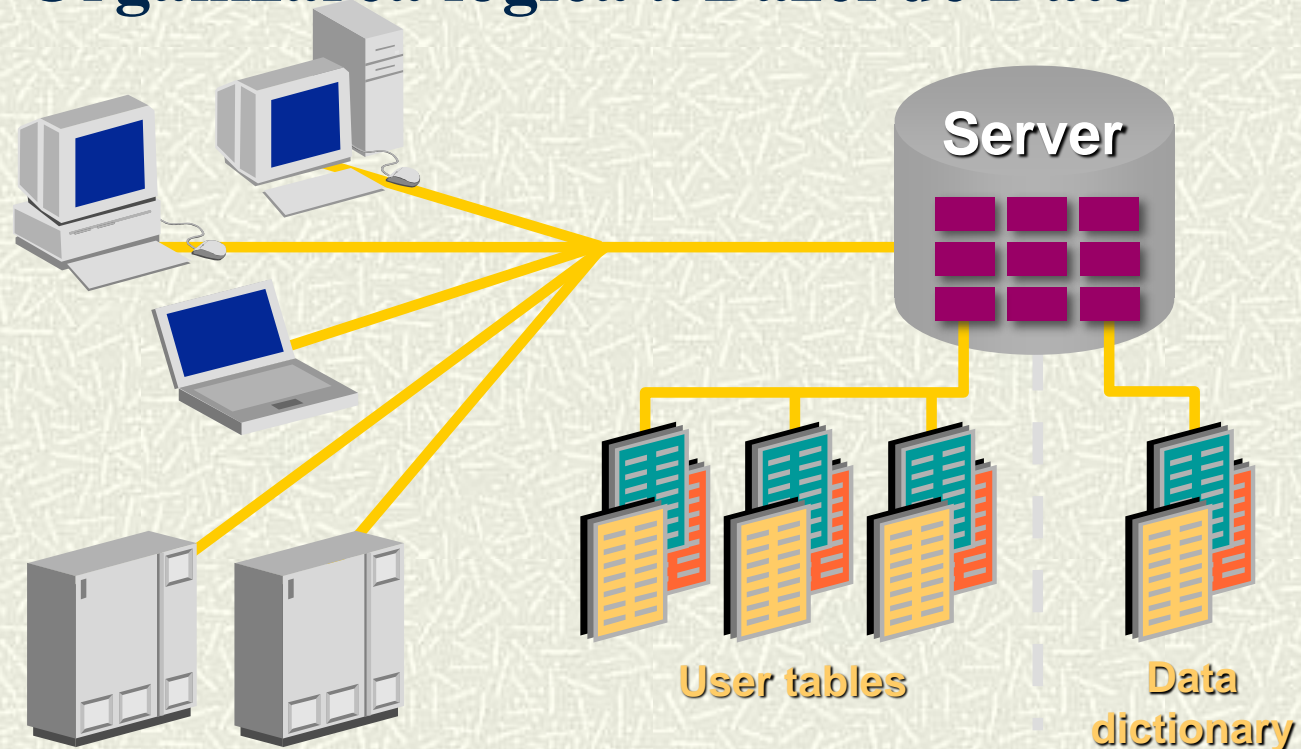
+ restricții de integritate

indecși,

tabele virtuale, ...

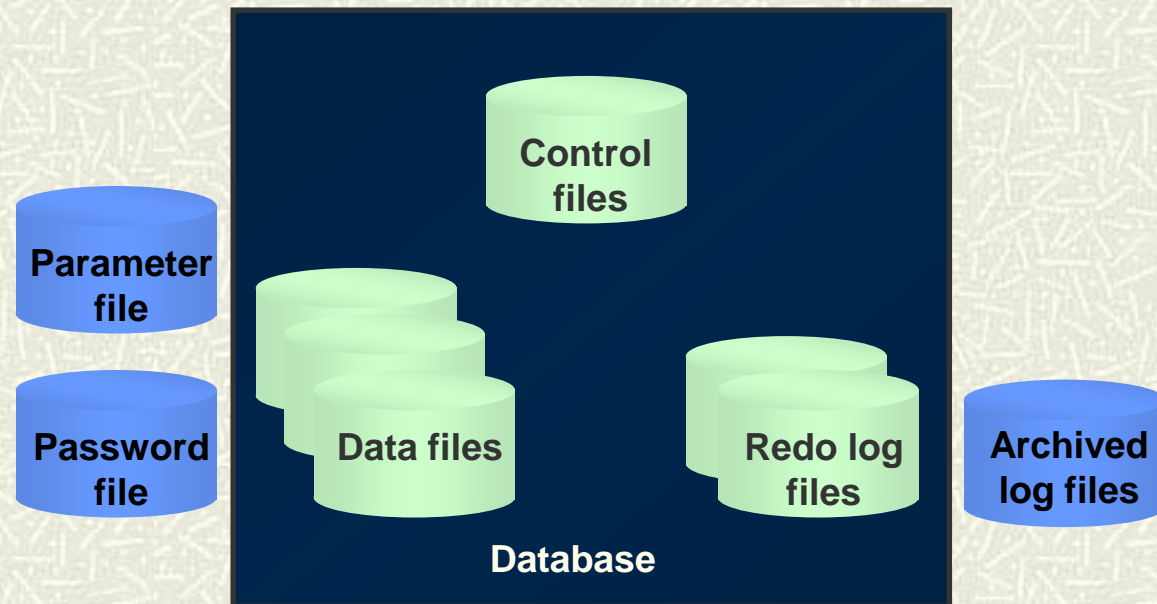
# D. 1. Stocarea informațiilor în baza de date

## Organizarea logică a Bazei de Date



# D. 1. Stocarea informațiilor în baza de date

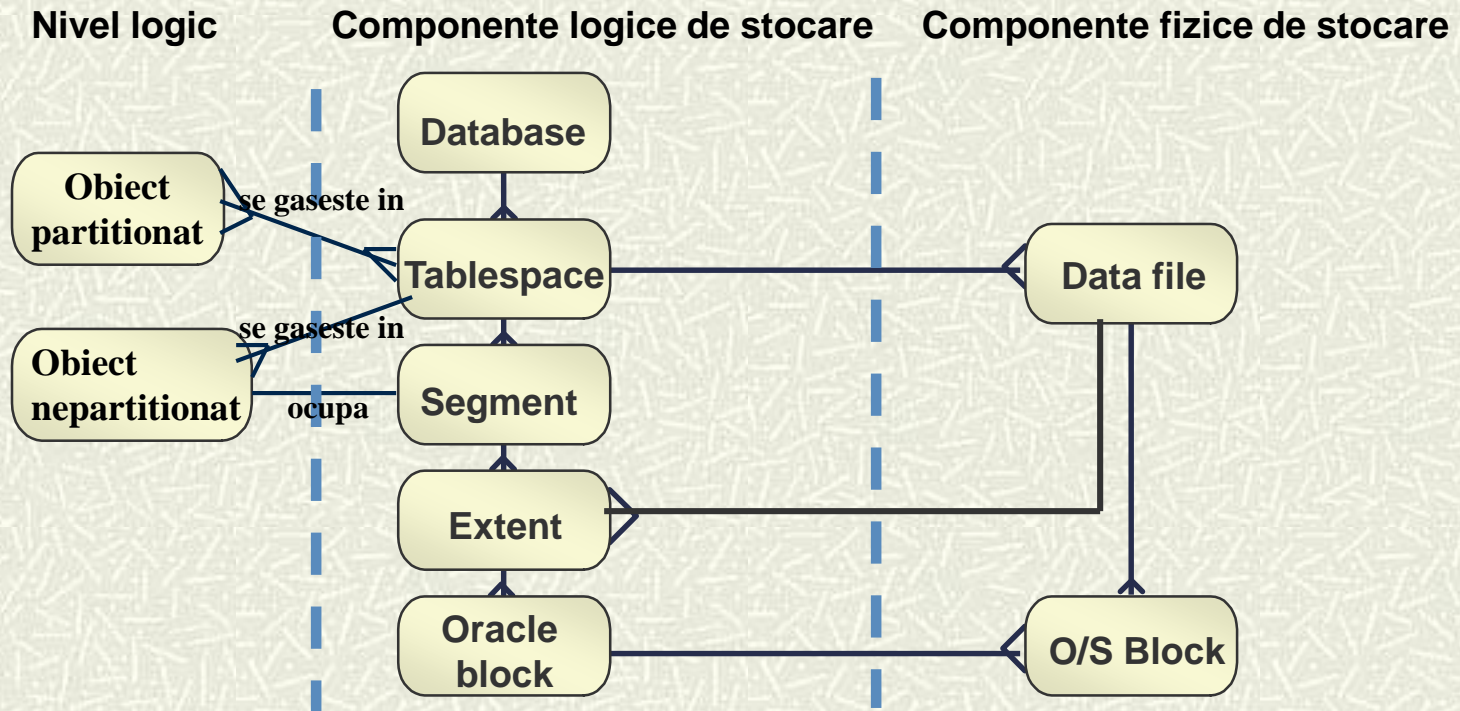
## Organizarea fizică a Bazei de Date





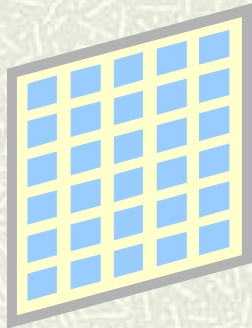
# D. 1. Stocarea informațiilor în baza de date

## Componente de date Oracle

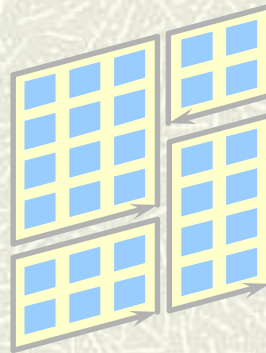


# D. 1. Stocarea informațiilor în baza de date

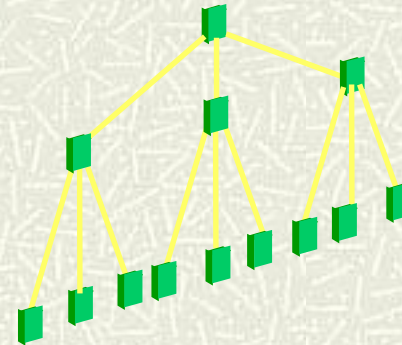
## Tipuri de segmente



**Table**



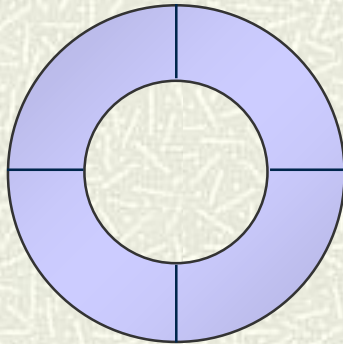
**Table  
partition**



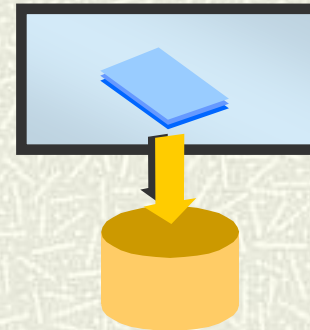
**Index**

# D. 1. Stocarea informațiilor în baza de date

## Tipuri de segmente



**Rollback  
segment**



**Temporary  
segment**

## **D. 1. Stocarea informațiilor în baza de date**

---

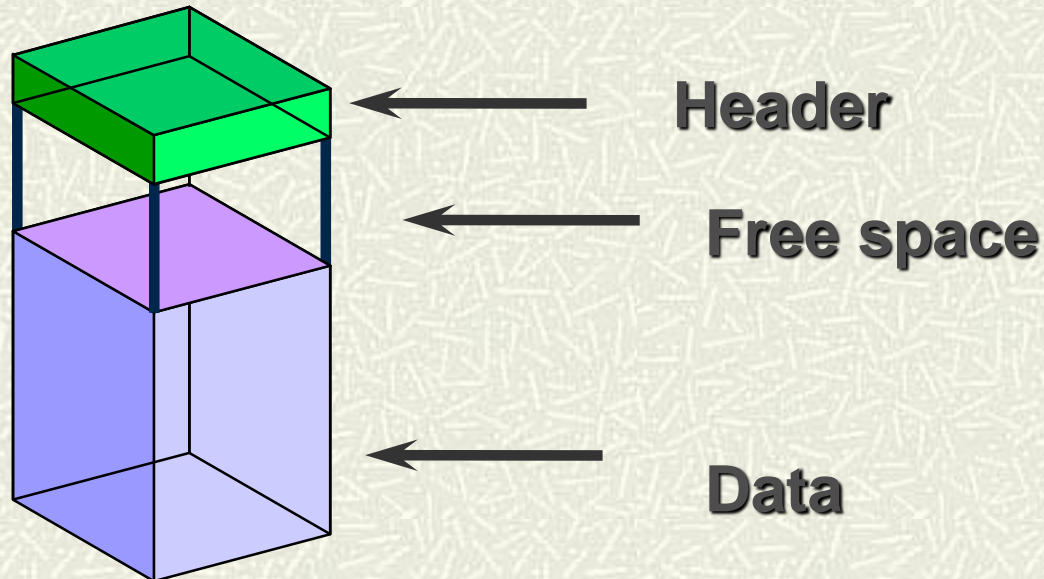
### **Blocul de baza de date (DataBase Block)**

- **Cea mai mică unitate de stocare**
- **Constă în unul sau mai multe blocuri ale SO**
- **Stabilită la crearea bazei de date prin parametrul DB\_BLOCK\_SIZE**



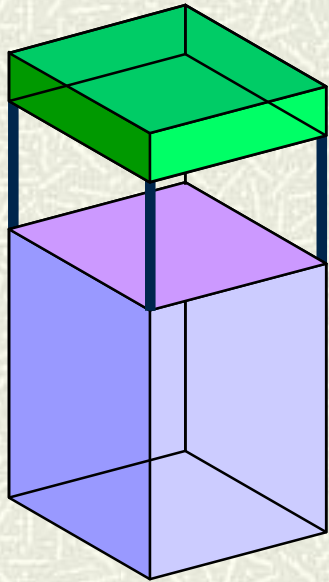
## D. 1. Stocarea informațiilor în baza de date

### Continutul unui bloc de baza de date



## D. 1. Stocarea informațiilor în baza de date

**Parametrii pentru utilizarea spațiului  
în blocul de bază de date**



**PCTFREE**

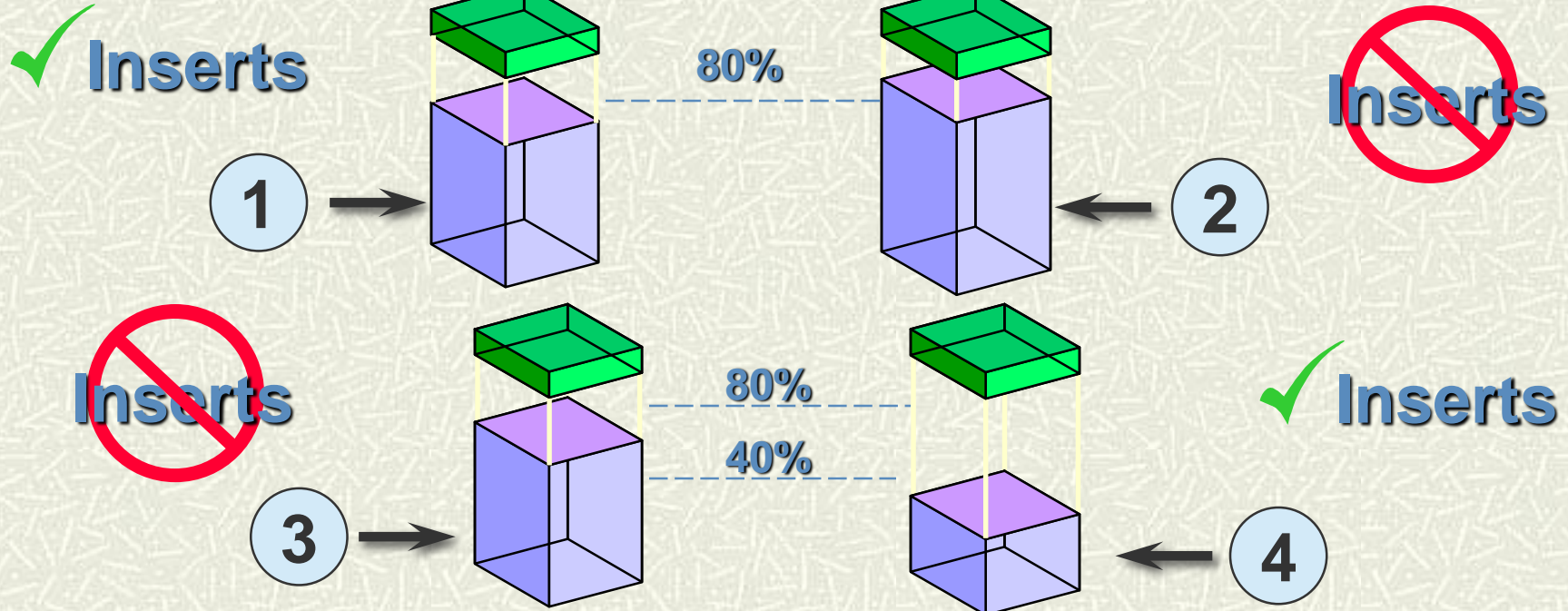
**PCTUSED**

# D. 1. Stocarea informațiilor în baza de date

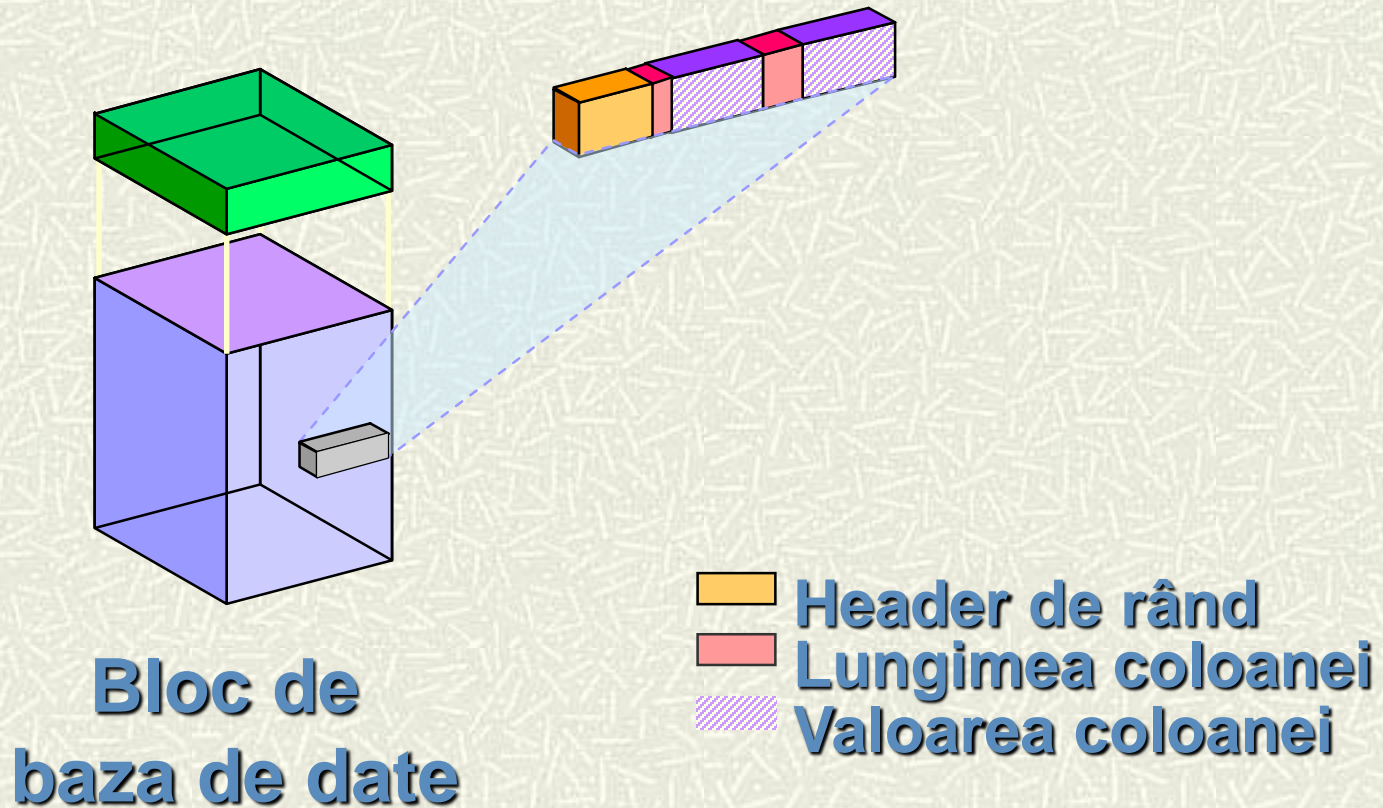
## Utilizarea spațiului în blocul de baza de date

PCTFREE=20

PCTUSED=40

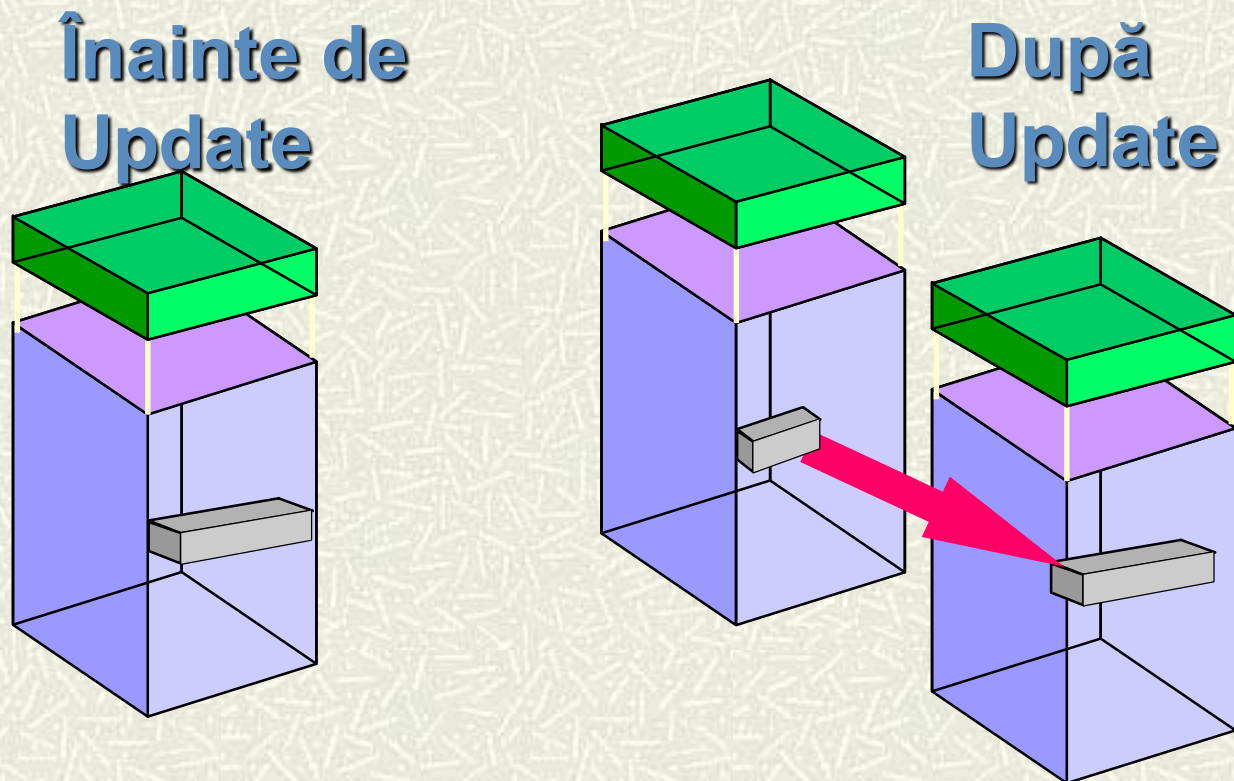


# Structura unui bloc





# Migrarea rândurilor



# Obținerea de informații

---

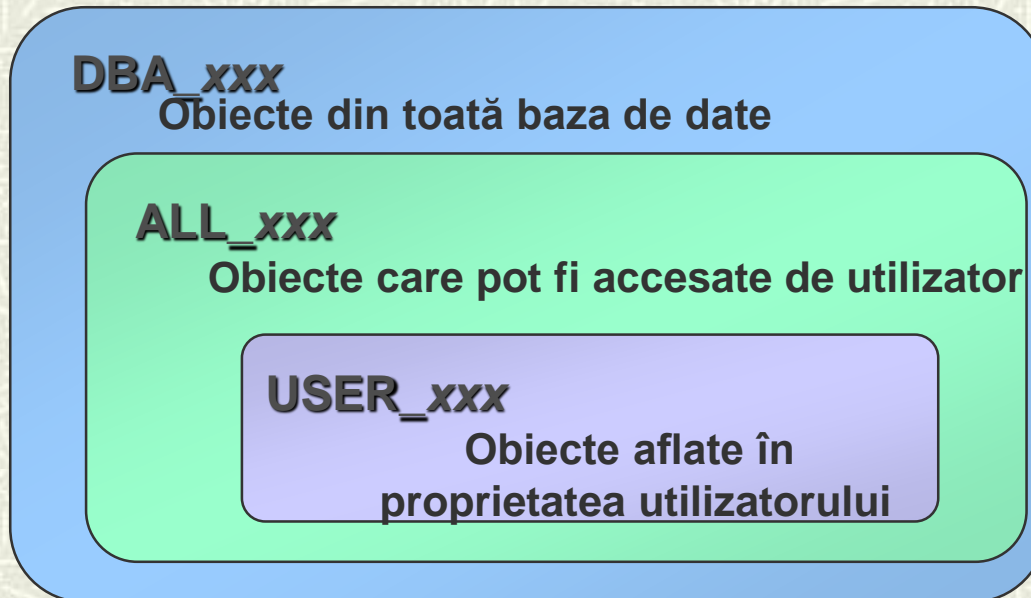
## Dicționarul de date

Dicționarul de date ( *Data dictionary* ) furnizează informații despre :

- structuri logice sau fizice
- nume, definiții, spații alocate obiectelor
- restricții de integritate
- monitorizare

# Obținerea de informații

## Dicționarul de date



# Obținerea de informații

## Dicționarul de date : exemple

Views	Description
<b>dictionary</b> <b>dict_columns</b>	<b>Informații generale</b>
<b>user_tables</b> <b>user_objects</b> <b>user_lobs</b> <b>user_tab_columns</b> <b>user_constraints</b>	<b>Informații relativ la obiectele utilizatorului: tabele, coloane, restricții, alte obiecte</b>
<b>dba_users</b> <b>dba_sys_privs</b> <b>dba_roles</b>	<b>Informații de administrare</b>



# Obținerea de informații

**Disponibile pentru utilizator :**

- **DICTIONARY**
- **USER\_USERS**
- **USER\_TABLES**
- **ALL\_TABLES**
- **USER\_VIEWS**
- **USER\_TAB\_COLUMNS**
- **SESSION\_PRIVS**
- **SESSION\_ROLES**
- . . . . .

# Obținerea de informații asupra tabelelor

## Exemplu

```
SQL> SELECT * FROM dictionary WHERE table_name LIKE '%TABLE%';
```

TABLE_NAME	COMMENTS
-----	-----
ALL_ALL_TABLES	Description of all object and relational tables accessible to the user
ALL_NESTED_TABLES	Description of nested tables in tables accessible to the user
ALL_OBJECT_TABLES	Description of all object tables accessible to the user
ALL_TABLES	Description of relational tables accessible to the user
.....	

15 rows selected.

# Obținerea de informații asupra tabelelor

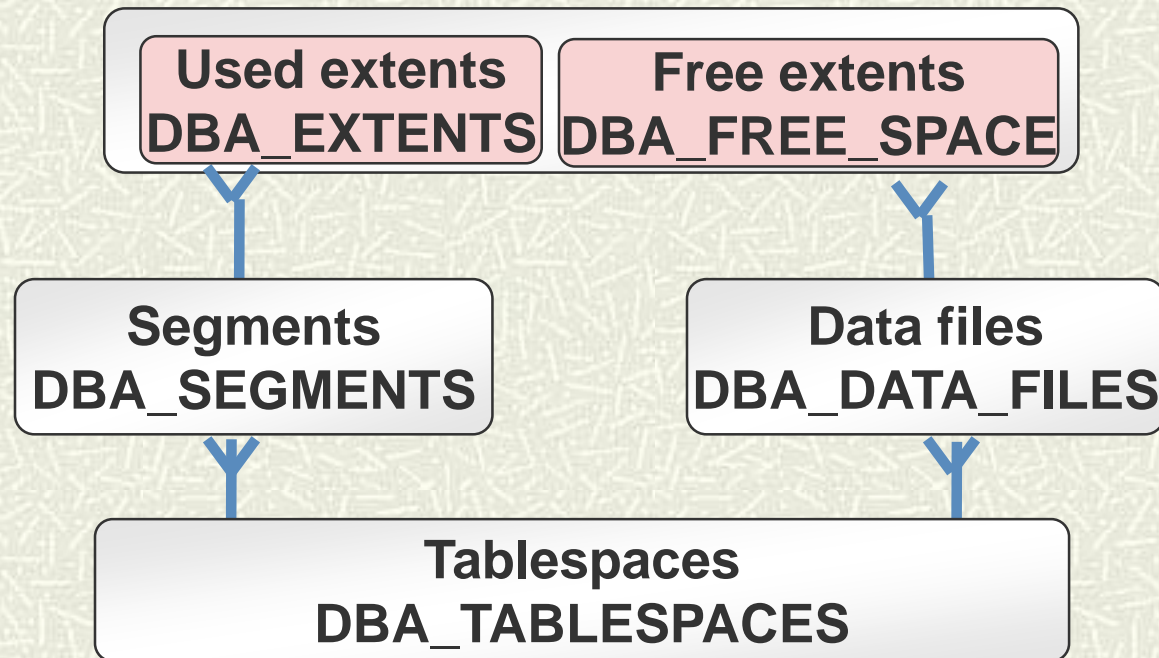
## Exemplu

```
SQL> describe user_tables
```

Name	Null?	Type
-----	-----	----
TABLE_NAME	NOT NULL	VARCHAR2(30)
TABLESPACE_NAME		VARCHAR2(30)
PCT_FREE		NUMBER
PCT_USED		NUMBER
INI_TRANS		NUMBER
MAX_TRANS		NUMBER
INITIAL_EXTENT		NUMBER
NEXT_EXTENT		NUMBER
MIN_EXTENTS		NUMBER
.....		

## D. 1. Stocarea informațiilor în baza de date

### Informații în dicționarul de date (pentru administrator)





## **D. 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**

# D. 2. 1. Obiectele bazei de date: Tabele

***Tabela (relația)*** - componenta principală a bazei de date relaționale

- structură pe *linii* (***rânduri*** sau *înregistrări*)  
și ***coloane*** (*attribute* sau *câmpuri*)

**Elemente importante:**

- cheia primară
- cheie străină
- cheie secundară

**Operații:**

- creare
- interogare
- actualizare : adăugare, modificare, ștergere de rânduri
- ștergere

# Creare de tabele

---

1. **Crearea unei tabele de sine stătătoare**
2. **Crearea unei tabele pe baza altora, deja existente.**

# Crearea unei tabele de sine stătătoare

```
CREATE TABLE [schema.] tabela  
(coloană tip [, coloană tip, ...] )  
[TABLESPACE tablespace ]  
[PCTFREE integer ] [PCTUSED integer ]  
[STORAGE ( INITIAL integer [ { K | M } ]  
NEXT integer [ { K | M } ]  
PCTINCREASE integer  
MINEXTENTS integer  
MAXEXTENTS integer )] ;
```

*schema* numele proprietarului tabelei  
TABLESPACE numele spațiului de stocare  
PCTFREE procentul de spațiu liber a blocului rezervat pentru actualizări  
PCTUSED procentul minim de ocupare permanentă a unui bloc  
STORAGE indicații de alocare a spațiului (dimensiunea spațiului inițial, a celei de a doua extensii, procentul de creștere începând de la a treia extensie, numărul minim și maxim de extensii)



# Crearea unei tabele de sine stătătoare

## Exemplu

```
CREATE TABLE employees (  
    empno NUMBER(4) ,  
    last_name VARCHAR2(30)  
    deptno NUMBER(2) )  
TABLESPACE data01  
PCTFREE 20 PCTUSED 50  
STORAGE (INITIAL 200K NEXT 200K  
PCTINCREASE 0 MAXEXTENTS 50) ;
```

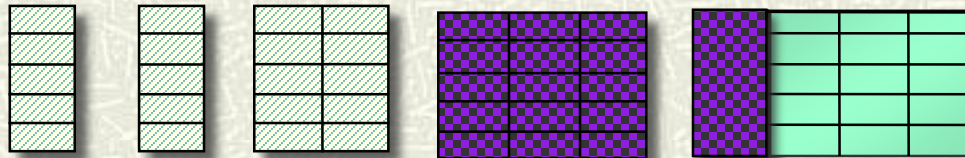
# Modificarea parametrilor de stocare

```
ALTER TABLE [schema.] tabela
[PCTFREE integer] [PCTUSED integer]
[STORAGE ( INITIAL integer [ K | M ]
           NEXT integer [ K | M ]
           PCTINCREASE integer
           MINEXTENTS integer
           MAXEXTENTS integer ) ] ;
```

```
ALTER TABLE scott.employees
PCTFREE 30
PCTUSED 50
STORAGE ( NEXT 200K
          MINEXTENTS 2
          MAXEXTENTS 100) ;
```

# Dealocarea spațiului neutilizat

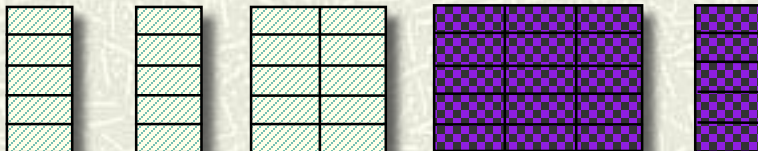
**Before  
deallocation**



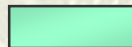
```
ALTER TABLE scott.employees  
DEALLOCATE UNUSED;
```

High water  
mark

**After  
deallocation**



Used block



Unused block



Free space after delete

# Crearea unei tabele de sine stătătoare

```
CREATE TABLE [schema.] tabela  
(coloană tip [, coloană tip, ...] )  
[TABLESPACE tablespace ]  
[PCTFREE integer ] [PCTUSED integer ]  
[STORAGE ( INITIAL integer [ K | M ]  
            NEXT integer [ K | M ]  
            PCTINCREASE integer  
            MINEXTENTS integer  
            MAXEXTENTS integer ) ] ;
```

În același timp, se pot defini și  
restricții de integritate  
(CONSTRAINTS)



# Creare de tabele

---

1. **Crearea unei tabele de sine stătătoare**
2. **Crearea unei tabele pe baza altora, deja existente**

# Crearea unei tabele pe baza altora

```
CREATE TABLE [schema.] tabela
    [ (coloană [, coloană, ...] ) ]
[TABLESPACE tablespace ]
[PCTFREE integer ] [PCTUSED integer ]
[STORAGE ( INITIAL integer [ { K | M } ]
            NEXT integer [ { K | M } ]
            PCTINCREASE integer
            MINEXTENTS integer
            MAXEXTENTS integer ) ]
AS subinterogare;
```

# Obținerea de informații asupra tabelelor

**Disponibile pentru utilizator :**

- **DICTIONARY**
- **USER\_USERS**
- **USER\_TABLES**
- **ALL\_TABLES**
- **USER\_VIEWS**
- **USER\_TAB\_COLUMNS**
- **SESSION\_PRIVS**
- **SESSION\_ROLES**
- . . . . .

# Obținerea de informații asupra tabelelor

## Exemplu

```
SQL> SELECT * FROM dictionary WHERE table_name LIKE '%TABLE%';
```

TABLE_NAME	COMMENTS
-----	-----
ALL_ALL_TABLES	Description of all object and relational tables accessible to the user
ALL_NESTED_TABLES	Description of nested tables in tables accessible to the user
ALL_OBJECT_TABLES	Description of all object tables accessible to the user
ALL_TABLES	Description of relational tables accessible to the user
.....	

15 rows selected.



# Obținerea de informații asupra tabelelor

## Exemplu

```
SQL> describe user_tables
```

Name	Null?	Type
-----	-----	----
TABLE_NAME	NOT NULL	VARCHAR2(30)
TABLESPACE_NAME		VARCHAR2(30)
PCT_FREE		NUMBER
PCT_USED		NUMBER
INI_TRANS		NUMBER
MAX_TRANS		NUMBER
INITIAL_EXTENT		NUMBER
NEXT_EXTENT		NUMBER
MIN_EXTENTS		NUMBER
.....		

# Obținerea de informații asupra tabelelor

## Exemplu

```
SQL> select table_name,tablespace_name,pct_free,pct_used,  
2  initial_extent,next_extent from user_tables  
3  where table_name='PROD';
```

TABLE_NAME	TABLESPACE_NAME	PCT_FREE	PCT_USED	INITIAL_EXTENT	NEXT_EXTENT
PROD	DATA01	10	40	10240	10240

# Obținerea de informații asupra tabelelor

## Comanda ANALYZE

Comanda ANALYZE generează statistici referitoare la stocare, pe baza întregii tabele, sau a unei porțiuni

```
ANALYZE TABLE [schema.] tabela  
COMPUTE STATISTICS ;
```

sau

```
ANALYZE TABLE [schema.] tabela  
ESTIMATE STATISTICS  
[ SAMPLE integer { ROWS | PERCENT } ] ;
```

# Obținerea de informații asupra tabelelor

## Exemplu

```
SQL> ANALYZE TABLE prod ESTIMATE STATISTICS;
```

Table analyzed.

```
SQL> SELECT num_rows, avg_space, sample_size,  
2      avg_row_len, chain_cnt  
3      FROM user_tables WHERE table_name='PROD';
```

NUM_ROWS	AVG_SPACE	SAMPLE_SIZE	AVG_ROW_LEN	CHAIN_CNT
9	1718	1064	24	0



# Comentarii la tabele și coloane

```
COMMENT ON TABLE tabela  
IS 'text' ;
```

sau

```
COMMENT ON COLUMN tabela.coloana  
IS 'text' ;
```

# Comentarii la tabele și coloane

## Exemplu

```
SQL> COMMENT ON TABLE prod IS 'Nomenclatorul de produse';
```

Comment created.

```
SQL> SELECT table_name, comments FROM user_tab_comments  
2 WHERE table_name='PROD';
```

TABLE_NAME	COMMENTS
-----	-----
PROD	Nomenclatorul de produse

# Comentarii la tabele și coloane

## Exemplu

```
SQL> COMMENT ON COLUMN prod.den IS 'Denumirea produsului';
```

Comment created.

```
SQL> SELECT table_name, column_name, comments  
2 FROM user_col_comments WHERE table_name='PROD';
```

TABLE_NAME	COLUMN_NAME	COMMENTS
------------	-------------	----------

PROD	CODP	
PROD	DEN	Denumirea produsului
PROD	CULOARE	
PROD	STOC	
PROD	PRET	

# Golirea unei tabele

```
TRUNCATE TABLE [schema.] tabela  
[ { DROP | REUSE } STORAGE ] ;
```



# Ștergerea unei tabele

```
DROP TABLE [schema.] tabela  
[ CASCADE CONSTRAINTS ] ;
```

## **D. 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**