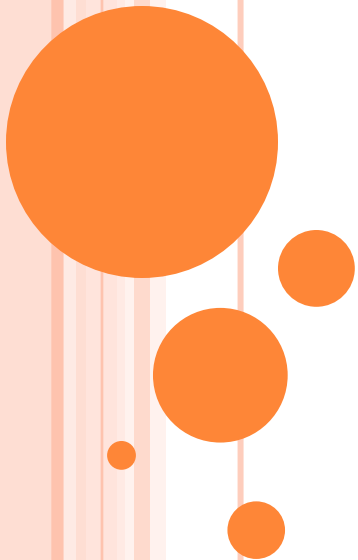


INTRODUCEREA SI ACTUALIZAREA DATELOR IN TABELE

- Cum se adaugă o linie nouă într-o tabelă
- Cum se sterge o linie dintr-o tabelă
- Cum se modifică valorile dintr-o tabelă



ADĂUGAREA DATELOR ÎN TABELE

- următoarea:

INSERT INTO *nume_tabela* (*lista_coloane*)
VALUES (*lista_valori*);

- unde ***nume_tabela*** este numele tabelii în care vom insera noua linie,
- ***lista_coloane*** precizează exact coloanele pe care dorim să le populăm. Această listă este opțională (ea poate lipsi).
- ***lista_valori*** specifică valorile pe care le va lua, pe rand, coloanele din lista de coloane.
- Lista de coloane și lista de valori trebuie să aibă același număr de elemente, și în plus coloanele și valorile din cele două liste trebuie să corespundă ca ordine și tip.
- Valorile specificate în listă (sau cele implicite) într-o comandă **INSERT**, trebuie să satisfacă toate constrângerile aplicabile coloanelor respective (ca de exemplu **PRIMARY KEY**, **CHECK**, **NOT NULL**).



- Pentru a adăuga linii într-o tabelă se utilizează comanda **INSERT**. Forma generală a acestei comenzi este
- Dacă la rularea unei comenzi **INSERT** este generată o eroare de sintaxă, sau a fost încălcată o constrângere, linia nu este adăugată la tabelă ci se va genera un mesaj de eroare.
- Atunci când din lista de coloane este omisă o coloană, Oracle va completa valoarea acelei coloane cu **NULL**, cu excepția situației când a fost definită o valoare implicită pentru coloana respectivă. În acest caz, Oracle completează coloana cu valoarea implicită. Dacă omiteți din lista de coloane o coloană care nu poate avea valoarea **NULL** (s-a definit o restricție **NOT NULL** sau **PRIMARY KEY**), și nu este definită o valoare implicită pentru acea coloană, se va genera o eroare.



- Pentru a exemplifica modul de funcționare a comenzii **INSERT** vom crea tabela **jucători**:

```
create table jucatori(  
  id NUMBER(5) PRIMARY KEY,  
  nume VARCHAR2(30) NOT NULL,  
  prenume VARCHAR2(30),  
  rating NUMBER(1) CHECK (rating between 1 and 5),  
  varsta NUMBER(2),  
  localitatea VARCHAR2(30) DEFAULT 'Timisoara',  
  email VARCHAR2(30) UNIQUE
```



- O comandă completă de inserare a unei linii în această tabelă se poate scrie:

```
insert into jucatori (id, nume, prenume, rating, varsta, localitatea, email)
```

```
values (18, 'Ionescu', NULL, 3, 30, 'Sibiu', 'user18@games.ro')
```

- Fără a mai specifica coloanele putem scrie următoarea comandă, în care am ținut cont de ordinea coloanelor în tabelă:

```
insert into jucatori values (11, 'Georgescu', 'Valeriu', 1, 18, 'Bucuresti', user11@games.ro')
```



- Comanda următoare are ca efect completarea coloanelor **id**, **nume**, **prenume** cu valorile specificate în lista de valori iar coloanele **rating**, **varsta**, **localitatea**, **email** cu valorile implicite pentru aceste coloane, adică **'Timisoara'** pentru **localitate** și respectiv **NULL** pentru **rating**, **varsta**, **email**:

**insert into jucatori (id, nume, prenume)
values (22, 'Vasilescu', 'Anca')**

Autocommit Display 10

```
select * from jucatori
```

Results Explain Describe Saved SQL History

ID	NUME	PRENUME	RATING	VARSTA	LOCALITATEA	EMAIL
11	Georgescu	Valeriu	1	18	Bucuresti	user11@games.ro
22	Vasilescu	Anca	-	-	Timisoara	-
18	Ionescu	-	3	30	Sibiu	user18@games.ro

3 rows returned in 0.01 seconds [Download](#)



- Deși câmpul email are definită o restricție **UNIQUE**, putem insera încă o valoare **NULL** în această coloană, doar valorile nenule trebuind să fie unice. Observați în comanda următoare cum s-a precizat că dorim setarea valorii implicite și a valorii NULL pentru câmpurile **localitate**, **rating** și **email**.

insert into jucatori (id, nume, prenume, rating, varsta, localitatea, email)

values (37, 'Enescu', 'Monica', NULL, 26, DEFAULT, NULL)

Autocommit Display 10

```
select * from jucatori
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

ID	NUME	PRENUME	RATING	VARSTA	LOCALITATEA	EMAIL
11	Georgescu	Valeriu	1	18	Bucuresti	user11@games.ro
22	Vasilescu	Anca	-	-	Timisoara	-
18	Ionescu	-	3	30	Sibiu	user18@games.ro
37	Enescu	Monica	-	26	Timisoara	-

4 rows returned in 0.01 seconds [Download](#)



- **Nu** putem însă inițializa coloanele **id** sau **nume** cu o valoare implicită, această valoare implicită fiind în acest caz valoare **NULL**, care nu este permisă pentru cheia primară sau pentru o coloană având restricția **NOT NULL**:

```
Autocommit Display 10
insert into jucatori (prenume, varsta) values('Maria', 29)
```

Results Explain Describe Saved SQL History

```
ORA-01400: cannot insert NULL into ("RO_GLS2_SQL01_T01"."JUCATORI"."ID")
```

```
Autocommit Display 10
insert into jucatori (id, prenume, rating, varsta,
                    localitatea, email)
values (91, 'Veronica', 4, 10,
        'Iasi', 'user91@games.ro')
```

Results Explain Describe Saved SQL History

```
ORA-01400: cannot insert NULL into ("RO_GLS2_SQL01_T01"."JUCATORI"."NUME")
```



- Pentru completarea unui câmp putem folosi o subinterogare ca în următorul exemplu:

**insert into jucatori (id, nume, prenume)
values ((select max(id)+1 from jucatori),
'Plesca','Ovidiu')**

Autocommit Display 10

```
select * from jucatori
```

Results Explain Describe Saved SQL History

ID	NUME	PRENUME	RATING	VARSTA	LOCALITATEA	EMAIL
11	Georgescu	Valeriu	1	18	Bucuresti	user11@games.ro
22	Vasilescu	Anca	-	-	Timisoara	-
43	Marin	Adriana	1	32	Brasov	user43@yahoo.com
18	Ionescu	-	3	30	Sibiu	user18@games.ro
37	Enescu	Monica	-	26	Timisoara	-
44	Plesca	Ovidiu	-	-	Timisoara	-

6 rows returned in 0.01 seconds [Download](#)



- În Oracle este permisă adăugarea mai multor linii simultan prin preluarea datelor din alte tabele, cu ajutorul unei subinterogări. Comanda următoare, de exemplu, preia toți angajații din tabela **employees** care au **job_id**-ul egal cu **'IT_PROG'** și îi inserează în tabela **jucători**:

insert into jucatori (id, nume)

select employee_id, last_name from employees

where job_id='IT_PROG'

Autocommit Display 30

```
select * from jucatori
```

Results Explain Describe Saved SQL History

ID	HUME	PRENUME	RATING	VARSTA	LOCALITATEA	EMAIL
11	Georgescu	Valeriu	1	18	Bucuresti	user11@games.ro
22	Vasilescu	Anca	-	-	Timisoara	-
43	Marin	Adriana	1	32	Brasov	user43@yahoo.com
103	Hunold	-	-	-	Timisoara	-
104	Ernst	-	-	-	Timisoara	-
107	Lorentz	-	-	-	Timisoara	-
18	Ionescu	-	3	30	Sibiu	user18@games.ro
37	Enescu	Monica	-	26	Timisoara	-
44	Plesca	Ovidiu	-	-	Timisoara	-

3 rows returned in 0.01 seconds [Download](#)



ȘTERGEREA DATELOR DINTR-O TABELĂ

- Ștergerea uneia sau mai multor linii dintr-o tabelă se face utilizând comanda **DELETE** a cărei sintaxă este:

DELETE FROM *nume_tabela* WHERE *conditie*

- Liniile care se vor șterge sunt selectate folosind clauza **WHERE**:

DELETE FROM jucatori WHERE id>100

- Ștergerea liniilor se poate face și pe baza valorilor returnate de către o subinterogare:

DELETE FROM jucatori WHERE id <

(SELECT id FROM jucatori WHERE nume='Ionescu')

- Dacă este omisă clauza **WHERE**, se vor șterge toate liniile din tabelă, însă structura tabelii rămâne (se șterge doar conținutul tabelii, nu și tabela propriu-zisă). Deci comanda:

DELETE FROM jucatori

- șterge toate liniile din tabela **jucatori**. **Atenție!** Aceste linii nu vor mai putea fi recuperate.



MODIFICAREA DATELOR DINTR-O TABELĂ

- Modificarea uneia sau mai multor înregistrări (linii) dintr-o tabelă se realizează cu comanda **UPDATE** care are sintaxa:

UPDATE *nume_tabela*

SET *coloana1 = valoare1,*

coloana2 = valoare2,

...

WHERE *conditie*

- ca în următorul exemplu:

update jucatori

SET prenume='Emilian' **WHERE** id=18

- care modifică (completează) prenumele jucătorului cu id-ul **18**.



- Modificarea valorilor unei linii se poate face pe baza valorilor returnate de către o subinterogare. Astfel, dacă dorim să îi atribuim jucătorului cu **id-ul 44** același rating ca cel al jucătorului cu codul **18**, iar vârsta să fie cu **5** mai mare decât vârsta jucătorului cu codul **43**, vom scrie:

UPDATE jucatori

SET rating=(SELECT rating FROM jucatori WHERE id=18),

varsta=(SELECT varsta+5 FROM jucatori WHERE id=43)

WHERE id=44

- Dacă o subinterogare utilizată la actualizarea valorilor dintr-o coloană nu returnează nici o valoare, atunci câmpul respectiv va fi inițializat cu **NULL**:

UPDATE jucatori

SET rating = (SELECT rating FROM jucatori WHERE id=200)

WHERE id=44



Înainte rulării acestei comenzi conținutul tabeli **jucatori** era cea din figura din stânga, iar după rularea sa conținutul este cel din figura din dreapta. Se observă că inițial ratingul jucătorului **44** era **3**, iar după rularea comenzii acesta a devenit **null**.

Autocommit Display 30

```
select * from jucatori
```

Results Explain Describe Saved SQL History

ID	NUME	PRENUME	RATING	VARSTA	LOCALITATEA	EMAIL
11	Georgescu	Valeriu	1	18	Bucuresti	user11@games.ro
22	Vasilescu	Anca	-	-	Timisoara	-
43	Marin	Adriana	1	32	Brasov	user43@yahoo
103	Hunold	-	-	-	Timisoara	-
104	Ernst	-	-	-	Timisoara	-
107	Lorentz	-	-	-	Timisoara	-
18	Ionescu	Emilian	3	30	Sibiu	user18@game
37	Enescu	Monica	-	26	Timisoara	-
44	Plesca	Ovidiu	3	37	Timisoara	-

Autocommit Display 30

```
select * from jucatori
```

Results Explain Describe Saved SQL History

ID	NUME	PRENUME	RATING	VARSTA	LOCALITATEA	EMAIL
11	Georgescu	Valeriu	1	18	Bucuresti	user11@games.ro
22	Vasilescu	Anca	-	-	Timisoara	-
43	Marin	Adriana	1	32	Brasov	user43@yahoo.com
103	Hunold	-	-	-	Timisoara	-
104	Ernst	-	-	-	Timisoara	-
107	Lorentz	-	-	-	Timisoara	-
18	Ionescu	Emilian	3	30	Sibiu	user18@games.ro
37	Enescu	Monica	-	26	Timisoara	-
44	Plesca	Ovidiu	-	37	Timisoara	-

- Interesant este că o comandă de forma:

UPDATE jucatori

**SET rating = (SELECT rating FROM jucatori
WHERE id=18),**

**varsta = (SELECT varsta+5 FROM jucatori
WHERE id=18)**

WHERE id=44

- se poate scrie și astfel:

UPDATE jucatori

SET (rating, varsta) =

**(SELECT rating, varsta FROM jucatori WHERE
id=18)**

WHERE id=44



In această lecție am învățat:

- Cum se adaugă o linie nouă într-o tabelă
- Cum se șterge o linie dintr-o tabelă
- Cum se modifică valorile dintr-o tabelă

