

Prelucrarea valorilor NULL

Valori NULL

NULL indică o valoare inexistentă(necunoscută). Valoarea *null* este diferită de șirul vid sau 0. Două valori *null* nu sunt egale între ele. Comparația dintre două valori *NULL* sau dintre *NULL* și o altă valoare returnează o valoare necunoscută (*UNKNOWN*). Operatorii *is null* și *is not null* se folosesc pentru a testa în clauza *where* dacă o valoare este sau nu *null*.

```
Select count(*) from tStudenti where telefon is null
```

Funcția ISNULL

Funcția *ISNULL* înlocuiește o valoare *null* cu o valoare de înlocuire specificată.

Sintaxa

```
ISNULL (expresie, valoare_de_inlocuire)
```

Exemplu fie tabelul *tTest* ce conține coloana *Coloana1* cu următoarele valori:

Coloana1

5
7
NULL
3
NULL

atunci

```
Select AVG (Coloana1 ) from tTest
```

furnizează rezultatul 5(valorile null sunt ignorate), iar

```
Select AVG (ISNULL (Coloana1, 0) ) from tTest
```

furnizează rezultatul 3 (valorile null sunt înlocuite cu 0).

Funcția NULLIF

Sintaxa

```
NULLIF( expresie1, expresie2)
```

returnează *expresie1* dacă *expresie1* \neq *expresie2* și *NULL* dacă *expresie1* = *expresie2*

Funcția CASE

Funcția **CASE** evaluează o listă de expresii și corespunzător unui criteriu specificat returnează valoarea uneia dintre ele.

Există două variante de utilizare a funcției CASE

Sintaxa 1 :

```
CASE expresie_intrare
  WHEN expresie1 THEN rezultat1
  WHEN expresie2 THEN rezultat2
...
  WHEN expresien THEN rezultatn
  [ELSE rezultat]
END
```

Expresie_intrare se compara cu *expresie₁*, *expresie₂*, .. *expresie_n*. Dacă *expresie_intrare* este egală cu *expresie_k* atunci se returnează *rezultat_k*.

Dacă *expresie_intrare* este diferită de *expresie₁*, *expresie₂*, .. *expresie_n* se returnează *rezultat*. Clauza *else* este opțională. În cazul în care *else* lipsește, funcția *case* returnează valoarea *null* dacă expresia de intrare este diferită de *expresie₁*, *expresie₂*, .. *expresie_n*.

Sintaxa 2 :

```
CASE
  WHEN expresieLogical1 THEN rezultat1
  WHEN expresieLogica2 THEN rezultat2
...
  WHEN expresieLogica n THEN rezultatn
  [ELSE rezultat]
END
```

Funcția CASE returnează *rezultat_k* dacă *expresieLogica_k* este prima dintre expresiile adevărate.

Exemple:

```
SELECT
  Sum (CASE UM WHEN 'Kg' THEN CANTITATE ELSE 0 END) AS [TOTAL Kg],
  Sum (CASE WHEN UM='L' THEN CANTITATE END) AS [TOTAL LITRI]
FROM tDetaliiFact as A
  inner join
  tProduse as B on A.CodProdus = B.CodProdus
```

```
SELECT
  Sum (CASE WHEN um= 'Kg' THEN CANTITATE ELSE 0 END) AS [TOTAL Kg],
  Sum (CASE WHEN UM='L' THEN CANTITATE END) AS [TOTAL LITRI]
FROM tDetaliiFact as A
  inner join
```

```
tProduse as B on A.CodProdus = B.CodProdus
```

```
select
  count(case Month(data) when 1 then 1 end) as ianuarie,
  count(case when Month(data) = 2 then 1 end) as februarie,
  count(case Month(data) when 3 then 1 end) as martie,
  count(case Month(data) when 4 then 1 else null end) as aprilie
from tFacturi
```