

View

4. apríla 2015

View (Pohľad)

View (Pohľad)

Je to predpripravený select, ktorý sa vykoná keď sa odkáže na view.

Syntax

```
CREATE [OR REPLACE] [ FORCE | NOFORCE ]  
VIEW [schema.] pohľad [(alias_stlpca [...])]  
AS Select-prikaz  
[WITH [ READ ONLY | CHECK OPTION [CONSTRAINT obmedzenie] ] ]  
|  
CREATE VIEW [schema.] pohľad [(alias_stlpca [...])]  
AS Select-prikaz  
[WITH [ CASCADED | LOCAL] CHECK OPTION ]
```

- schema – názov schémy, v ktorej sa má pohľad nachádzať
- OR REPLACE – predefinovanie pohľadu, ak už bol definovaný
- FORCE – tento druh pohľadu je možné vytvoriť aj vtedy, ak tabuľky (objekty), z ktorých má byť pohľad odvodený neexistujú, alebo užívateľ, ktorý vytvára pohľad nemá na ne práva
- NOFORCE – implicitne – pohľad je možné vytvoriť len vtedy, ak základné tabuľky (objekty) existujú a užívateľ má na ne práva.
- READ ONLY – nedovolí operácie INSERT, UPDATE ani DELETE nad pohľadom.
- CHECK OPTION – kontroluje dodržiavanie podmienky WHERE pri operáciách INSERT, UPDATE, DELETE do pohľadu
- CONSTRAINT – pomenovanie obmedzenia
- CASCADED - kontrola podmienok v odvodených pohľadoch
- LOCAL – obmedzenie kontroly podmienok len na podmienku definovanú v danom pohľade

```
CREATE OR REPLACE VIEW poh11  
AS  
SELECT meno, priezvisko  
FROM os_udaje;
```

- vytvorenie, alebo predefinovanie pohľadu
- nie je možný INSERT

```
CREATE OR REPLACE VIEW poh11a  
AS  
SELECT rod_cislo, meno, priezvisko  
FROM os_udaje;
```

- vytvorenie, alebo predefinovanie pohľadu
- už je možný INSERT, ak všetky atribúty sú NOT NULL

Pohľad s predefinovaním stĺpcov

```
CREATE OR REPLACE VIEW zoznam(meno, priezvisko, pohlavie) AS
select ou.meno, ou.priezvisko,
       decode(substr(ou.rod_cislo,3,1),'5', 'zena', '6', 'zena','muz')
from os_udaje ou;
```

ALEBO

```
CREATE OR REPLACE VIEW zoznam AS
select ou.meno, ou.priezvisko,
       decode(substr(ou.rod_cislo,3,1),'5', 'zena', '6', 'zena','muz') as pohlavie
from os_udaje ou;
```

```
CREATE OR REPLACE VIEW  pohl2 AS
  select ou.meno, ou.priezvisko, ou.rod_cislo
  from os_udaje ou
  where ou.meno like 'K%'

WITH CHECK OPTION
```

Pozor!

Je možné vložiť do pohľadu pohl2 aj údaje, ktoré pri selecte nebudete vidieť, ale dáta budú vložené do zdrojovej tabuľky

```
INSERT INTO pohl2 (meno, priezvisko,rod_cislo)
VALUES ('Martinko','Klingacik','0512224/0000');
```


Zakázanie deštruktívnych operácií

```
CREATE OR REPLACE VIEW poh12 AS  
  select ou.meno, ou.priezvisko, ou.rod_cislo  
  from os_udaje ou  
  where ou.meno like 'K%'  
  WITH READ ONLY;
```

Pohľad s použitím funkcií

```
CREATE OR REPLACE VIEW pohl3 (meno, priezvisko, priemer)
AS
SELECT ou.meno, ou.priezvisko,
       avg(decode (zp.vysledok,'A',1,'B',1.5,'C',2,'D',2.5,'E',3, 4) )
FROM os_udaje ou JOIN student st USING(rod_cislo)
   JOIN zap_predmety zp USING(os_cislo)
GROUP BY ou.meno, ou.priezvisko, st.os_cislo
WITH READ ONLY;
```

Pohľad z viacerých tabuliek

```
CREATE OR REPLACE VIEW pohl4
AS
SELECT ou.meno, ou.priezvisko, st.rocnik,
       st.st_skupina, rod_cislo, st.os_cislo
FROM os_udaje ou JOIN student st USING(rod_cislo);

INSERT INTO pohl4 (meno, priezvisko, rocnik, skupina, rod_cislo, os_cislo)
VALUES ('Peter','Novy',1,'5ZI011','841231/1212',55);
```

Tento insert nefunguje, aby fungoval je potrebné definovať trigger namiesto Insertu.

Pohľad z viacerých tabuliek

```
CREATE OR REPLACE VIEW pohl4
AS
SELECT ou.meno, ou.priezvisko, st.rocnik,
       st.st_skupina, rod_cislo, st.os_cislo
FROM os_udaje ou JOIN student st USING(rod_cislo);

INSERT INTO pohl4 (meno, priezvisko, rocnik, skupina, rod_cislo, os_cislo)
VALUES ('Peter','Novy',1,'5ZI011','841231/1212',55);
```

Tento insert nefunguje, aby fungoval je potrebné definovať trigger namiesto Insertu.

```
CREATE OR REPLACE TRIGGER pohl4_ins
INSTEAD OF INSERT ON pohl4
referencing new as novy
Begin
    insert into os_udaje (meno, priezvisko, rod_cislo)
    values (:novy.meno,:novy.priezvisko, :novy.rod_cislo);

    insert into student
    ( rod_cislo, os_cislo, st_skupina, rocnik, st_odbor, st_zameranie)
    values (:novy.rod_cislo,:novy.os_cislo,:novy.skupina,:novy.rocnik, 100, 0);

-- 100, 0 je to potrebne, aby boli dodržané pravidlá referenčnej integrity, Informatika bez zamerania
END;
/
```

Delete z pohľadu z viacerých tabuliek

```
DELETE FROM poh14  
WHERE os_cislo = 55;
```

POZOR!!!

Tento delete funguje 'záhadne' - z pohľadu síce riadok zmizne, ale v tabuľke os_udaje zostanú údaje o študentovi - Peter Novy

Delete z pohľadu z viacerých tabuliek

```
DELETE FROM poh14  
WHERE os_cislo = 55;
```

POZOR!!!

Tento delete funguje 'záhadne' - z pohľadu síce riadok zmizne, ale v tabuľke os_udaje zostanú údaje o študentovi - Peter Novy

```
CREATE OR REPLACE TRIGGER poh14_del  
  INSTEAD OF DELETE ON poh14  
  referencing old as stary  
begin  
  delete from student  
  where os_cislo = :stary.os_cislo;  
  
  delete from os_udaje  
  where rod_cislo = :stary.rod_cislo;  
end;  
/
```

Pohľad z pohľadu - DELETE

```
CREATE OR REPLACE VIEW poh15  
AS  
SELECT meno, priezvisko, rod_cislo  
FROM poh14;
```

Pozor!!!

Insert nebude fungovať, lebo nemáte všetky potrebné údaje. Ale nasledovný DELETE vymaže nielen z tabuľky os_udaje, ako by sa zdalo, ale aj z tabuľky študent.

```
DELETE FROM poh15  
WHERE rod_cislo = '0512224/0000';
```

Pohľad z pohľadu - CHECK OPTION

```
CREATE OR REPLACE VIEW pohl6 (meno, priezvisko, rod_cislo)
AS
SELECT ou.meno, ou.priezvisko, ou.rod_cislo
FROM os_udaje ou
where ou.meno like 'S%';

CREATE OR REPLACE VIEW pohl7
AS
SELECT * FROM pohl6
where rod_cislo like '79%'

WITH CHECK OPTION;
```


Pohľad z pohľadu - CHECK OPTION

```
CREATE OR REPLACE VIEW poh16 (meno, priezvisko, rod_cislo)
AS
SELECT ou.meno, ou.priezvisko, ou.rod_cislo
FROM os_udaje ou
where ou.meno like 'S%';
```

```
CREATE OR REPLACE VIEW poh17
AS
SELECT * FROM poh16
where rod_cislo like '79%'
WITH CHECK OPTION;
```

- Nasledovný insert funguje

```
INSERT INTO poh16
VALUES ( 'Karol', 'Novy', '790502/1212');
```

Pohľad z pohľadu - CHECK OPTION

```
CREATE OR REPLACE VIEW poh16 (meno, priezvisko, rod_cislo)
AS
SELECT ou.meno, ou.priezvisko, ou.rod_cislo
FROM os_udaje ou
where ou.meno like 'S%';
```

```
CREATE OR REPLACE VIEW poh17
AS
SELECT * FROM poh16
where rod_cislo like '79%'
WITH CHECK OPTION;
```

- Nasledovný insert funguje

```
INSERT INTO poh16
VALUES ( 'Karol', 'Novy', '790502/1212');
```

- Tento insert nefunguje. Klausula WITH CHECK OPTION kontroluje aj zdedené podmienky.

```
INSERT INTO poh17
VALUES ( 'K arol', 'Novy', '79 0502/1212');
```

- ak SELECT pohľadu obsahuje PK a všetky ostatné NOT NULL stĺpce tabuľky, potom je možné vykonať INSERT
- ak je pohľad definovaný ako READ ONLY, nie sú možné operácie INSERT, DELETE a UPDATE
- ak je pohľad definovaný ako WITH CHECK OPTION, potom do pohľadu je možné vložiť len riadky, ktoré zodpovedajú podmienkam SELECTu a zdedeným podmienkam
- ak je SELECT z viacerých tabuliek, pre operácie INSERT, DELETE a UPDATE je nutné definovať triggre INSTEAD OF