

6. Virtualiosios lentelės ir duomenų nepriklausomumo lygiai

Virtualiaja lentele (vaizdiniu, rodiniu) (angl. *view*) vadinama užklausa, kuriai suteikiamas vardas ir ji įsimenama DB-je (jos sisteminame kataloge).

DB vartotojas gali peržiūrėti (rašyti užklausas) virtualiajai priskirtus duomenis.

Virtualiosios turi laikinųjų lentelių savybių, bet virtualiosios taikymo sritis platesnė.

Realiose lentelėse laikomi (saugomi) duomenys:

Vykdytojai

Nr	Pavardė	Kvalifikacija	Kategorija	Išsilavinimas
1	Jonaitis	Informatikas	2	VU
2	Petraitis	Statistikas	3	VU
3	Gražulytė	Inžinierius	1	NULL
4	Onaitytė	Vadybininkas	5	VDU
5	Antanaitis	Informatikas	3	VU

Galima ieškoti lentelėse esančių duomenų:

```
SELECT Pavardė, Kategorija, Išsilavinimas
FROM Vykdytojai WHERE Kvalifikacija = 'Informatikas'
```

Pavardė	Kategorija
Jonaitis	2
Antanaitis	3

Nei užklausa, nei jos rezultatas DB-je nesaugomi.

6.1. Virtualiųjų lentelių sudarymas

Virtualioji lentelė **neturi savo duomenų!**

Realiai egzistuoja tik virtualios lentelės apibrėžimas:

```
CREATE VIEW <virtualios lentelės vardas>
[(<stulpelio vardas> {,< stulpelio vardas> })]
AS <užklausa> [WITH CHECK OPTION]
```

Virtualioji lentelė (jos apibrėžimas) išlieka tol, kol ji nesunaikinama,

```
DROP VIEW <virtualiosios lentelės vardas>
```

Virtualioji lentelė, su informacija apie informatikus:

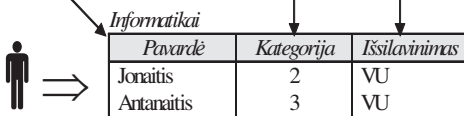
```
CREATE VIEW Informatikai
(Pavardė, Kategorija, Išsilavinimas)
AS SELECT Pavardė, Kategorija, Išsilavinimas
FROM Vykdytojai
WHERE Kvalifikacija = 'Informatikas'
```

Virtualioji lentelė yra „langas“ į tikrą (realią) lentelę.

Informatikai yra „langas“ į darbuotojus, per kurį matomi tik informatikai:

Vykdytojai

Nr	Pavardė	Kvalifikacija	Kategorija	Išsilavinimas
1	Jonaitis	Informatikas	2	VU
2	Petraitis	Statistikas	3	VU
3	Gražulytė	Inžinierius	1	NULL
4	Onaitytė	Vadybininkas	5	VDU
5	Antanaitis	Informatikas	3	VU



Informatikai		
Pavardė	Kategorija	Išsilavinimas
Jonaitis	2	VU
Antanaitis	3	VU

Užklausos virtualiosioms formuluojamos įprastai:

```
SELECT * FROM Informatikai WHERE Kategorija = 2
```

Sisteminame kataloge yra *Informatikai* apibrėžimas:

```
CREATE VIEW Informatikai
(Pavardė, Kategorija, Išsilavinimas)
AS SELECT Pavardė, Kategorija, Išsilavinimas
FROM Vykdytojai
WHERE Kvalifikacija = 'Informatikas'
```

DBVS užklausa **performuluoja** į užklausa lentelei:

```
SELECT Pavardė, Kategorija, Išsilavinimas
FROM Vykdytojai WHERE Kvalifikacija='Informatikas'
AND Kategorija = 2
```

Yra keletas ribojimų apibrėžiančiam užklausiai.

Pvz., SQL:2003 standartas **neleidžia** apibrėžiančioje užklausoje naudoti **ORDER BY**.

Užklausos virtualiajai lentelei rezultatas rikiuojamas įprastai:

```
SELECT * FROM Informatikai
ORDER BY Pavardė
```

Dalis DBVS, pvz. PostgreSQL, šio ribojimo nepaiso.

6.2. Virtualiųjų lentelių rūšys

Horizontali virtualioji-lentelės eilučių poaibis:

```
CREATE VIEW Vykdo_Jonaitis
AS SELECT * FROM Vykdymas
WHERE Vykdytojas =
(SELECT Nr FROM Vykdytojai
WHERE Pavardė = 'Jonaitis')
```

Vertikali virtualioji – realiosios stulpelių poaibis:

```
CREATE VIEW Vieši_Duomenys
AS SELECT Nr, Pavardė, Kvalifikacija
FROM Vykdytojai
```

Mišrieji rodiniai – lentelės eilučių ir stulpelių poaibis.

Rodinys *Informatikai* – mišrusis.

9-27

Rodinys vadinamas **grupiniu**, jei apibrėžime yra duomenų grupavimas:

```
CREATE VIEW Apie_Vykdyimą
(Projektas, Visos_Valandos, Vidurkis)
AS SELECT Projektas,
SUM(Valandos), AVG(Valandos)
FROM Vykdymas GROUP BY Projektas

SELECT AVG(Visos_Valandos)
FROM Apie_Vykdyimą
```

10-27

Rodinys yra **jungtinis**, kai užklausoje – kelios lentelės.

```
CREATE VIEW Projektai_Vykdymas
AS SELECT Projektai.*, Vykdytojas, Pavardė,
Kvalifikacija, Kategorija, Išsilavinimas,
Statusas, Valandos
FROM Projektai, Vykdytojai, Vykdymas
WHERE Projektas = Projektai.Nr AND
Vykdytojas = Vykdytojai.Nr
```

11-27

Apibrėžiančioje užklausoje galima kreiptis ir į virtualiąją,

```
CREATE VIEW ProjektasI_Vykdymas
AS SELECT *
FROM Projektai_Vykdymas
WHERE Nr = 1
```

12-27

6.3. Virtualiųjų lentelių atnaujinimas

```
UPDATE Informatikai
SET Kategorija = Kategorija + 1
WHERE Pavardė = 'Jonaitis'
```

DBVS performuluoja šį sakinį:

```
UPDATE Vykdytojai
SET Kategorija = Kategorija + 1
WHERE Pavardė = 'Jonaitis' AND
Kvalifikacija = 'Informatikas'
```

13-27

Virtualiosios lentelės duomenis **galima atnaujinti** (vykdyti **UPDATE** ir **DELETE**), jei apibrėžiančioji užklausa tenkina reikalavimus:

- nėra frazės **DISTINCT**;
- frazėje **FROM** – tik viena lentelė (gali būti ir virtuali), kurią galima atnaujinti;
- paieškos sąlygoje nėra kitos užklauskos;
- nėra duomenų grupavimo.

Tikslesni reikalavimai – DBVS dokumentacijoje.

14-27

Į virtualiąją lentelę **galima įvesti** duomenis (vykdyti **INSERT**), jei ją galima atnaujinti ir tenkina papildomus reikalavimus:

- į virtualiąją lentelę patenka visi bazinės lentelės stulpeliai neturintys numatytosios (**DEFAULT**) reikšmės;
- visi stulpeliai – paprasti bazinės lentelės stulpeliai, nėra reiškinių.

Tikslesni reikalavimai – DBVS dokumentacijoje.

15-27

6.4. Virtualiųjų lentelių atnaujinimo valdymas

Virtuali lentelė – vykdytojai, kurių kategorijos > 2:

```
CREATE VIEW Gudručiai
AS SELECT * FROM Vykdytojai
WHERE Kategorija > 2
```

16-27

Vykdotojai

Nr	Pavardė	Kvalifikacija	Kategorija	Išsilavinimas
1	Jonaitis	Informatikas	2	VU
2	Petraitis	Statistikas	3	VU
3	Gražulytė	Inžinierius	1	NULL
4	Onaitytė	Vadybininkas	5	VDU
5	Antanaitis	Informatikas	3	VU

Gudručiai

Nr	Pavardė	Kvalifikacija	Kategorija	Išsilavinimas
2	Petraitis	Statistikas	3	VU
4	Onaitytė	Vadybininkas	5	VDU
5	Antanaitis	Informatikas	3	VU

SELECT * FROM Gudručiai

INSERT INTO Gudručiai

VALUES (7, 'Uždavinys', 'Informatikas', 3, 'VU')

– įterptoji eilutė yra „matoma“.

INSERT INTO Gudručiai

VALUES (8, 'Juozaitis', 'Informatikas', 2, 'VDU')

– įterptoji eilutė yra „nematoma“

nors buvo įterpta į Gudručiai

SELECT * FROM Gudručiai

Nr	Pavardė	Kvalifikacija	Kategorija	Išsilavinimas
2	Petraitis	Statistikas	3	VU
4	Onaitytė	Vadybininkas	5	VDU
5	Antanaitis	Informatikas	3	VU
7	Uždavinys	Informatikas	3	VU

Vykdotojai

Nr	Pavardė	Kvalifikacija	Kategorija	Išsilavinimas
1	Jonaitis	Informatikas	2	VU
2	Petraitis	Statistikas	3	VU
3	Gražulytė	Inžinierius	1	NULL
4	Onaitytė	Vadybininkas	5	VDU
5	Antanaitis	Informatikas	3	VU
7	Uždavinys	Informatikas	3	VU
8	Juozaitis	Informatikas	2	VDU

Gudručiai

Nr	Pavardė	Kvalifikacija	Kategorija	Išsilavinimas
2	Petraitis	Statistikas	3	VU
4	Onaitytė	Vadybininkas	5	VDU
5	Antanaitis	Informatikas	3	VU
7	Uždavinys	Informatikas	3	VU

SELECT * FROM Gudručiai

– Antanaitis – „gudrutis“

UPDATE Gudručiai SET Kategorija = 2

WHERE Pavardė = 'Antanaitis'

– eilutė tampa „nematoma“

SELECT * FROM Gudručiai

– Antanaitis jau – ne „gudrutis“

Vykdotojai

Nr	Pavardė	Kvalifikacija	Kategorija	Išsilavinimas
1	Jonaitis	Informatikas	2	VU
2	Petraitis	Statistikas	3	VU
3	Gražulytė	Inžinierius	1	NULL
4	Onaitytė	Vadybininkas	5	VDU
5	Antanaitis	Informatikas	2	VU
7	Uždavinys	Informatikas	3	VU
8	Juozaitis	Informatikas	2	VDU

Gudručiai

Nr	Pavardė	Kvalifikacija	Kategorija	Išsilavinimas
2	Petraitis	Statistikas	3	VU
4	Onaitytė	Vadybininkas	5	VDU
7	Uždavinys	Informatikas	3	VU

Pasirinktis WITH CHECK OPTION

– daro rodinį „uždaru“

CREATE VIEW Gudručiai

AS SELECT * FROM Vykdotojai

WHERE Kategorija > 2

WITH CHECK OPTION – DBVS užtikrina, kad neatsirasų „nematomų“ eilučių.

INSERT INTO Gudručiai

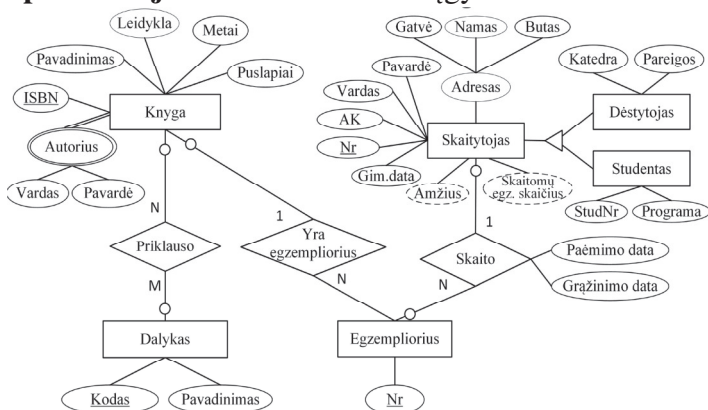
VALUES (8, 'Juozaitis', 'Informatikas', 2, 'VDU')

– SQL klaida: negalima įvesti duomenų

WITH CHECK OPTION atsiliepia efektyvumui.

Pasirinktis prasminga tik atnaujinamoms VIEW.

VIEW galima pasitelkti ER modelio esybių apskaičiuojamiems atributams įgyvendinti



Lentelės *Skaitytojas* stulpeliai su dviem papildomais stulpeliais, atitinkančiais apskaičiuojamus atributus: *Amžius* ir *Skaitomų egz. skaičius*:

```
CREATE VIEW Skaitytojas_viskas AS
SELECT A.*,
       EXTRACT(YEAR FROM
               AGE(Gimimas)) AS Amžius,
       (SELECT COUNT(*)
        FROM Egzemplierius AS B
        WHERE A.Nr= B.Skaitytojo_Nr
       ) AS Skaitomų_egz_skaičius
FROM Skaitytojas AS A
```

Pasitelkus tarpinę virtualiąją lentelę galima išvengti priklausomos vidinės užklaustos:

```
CREATE VIEW Skaitytojų_skaitymai AS
SELECT Skaitytojo_Nr,
       COUNT(*) AS Egz_skaičius
FROM Egzemplierius
WHERE Skaitytojo_Nr IS NOT NULL
GROUP BY Skaitytojo_Nr
```

```
CREATE VIEW Skaitytojas_viskas_gražiai AS
SELECT Skaitytojas.*,
       EXTRACT(YEAR FROM
               AGE(Gimimas)) AS Amžius,
       Egz_skaičius
FROM Skaitytojas LEFT OUTER JOIN
     Skaitytojų_skaitymai
ON Nr = Skaitytojo_Nr
```

Pastaba: daugelyje DBVS sakinyje **CREATE VIEW** negalima naudoti konstrukcijos **WITH**, todėl čia pasitelkta papildoma virtualioji lentelė.