

## 6. Virtualiosios lentelės ir duomenų nepriklausomumo lygiai

1-21

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

Virtualiosios turi laikinųjų savybes, bet virtualiosios taikomos žymiai plačiau.

### 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>
```

2-21

Virtualioji lentelė, su informacija apie informatikus:

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

3-21

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

*Informatikai* yra „langas“ į firmos 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

4-21

Užklausoje virtualiosioms formuluojamos įprastai:  
**SELECT \* FROM Informatikai WHERE Kategorija=2**  
Sisteminame kataloge yra *Informatikai* apibrėžimas:

5-21

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

DBVS užklausa **performuluoja** į užklausa lentelėi:  
**SELECT Pavardė, Kategorija, Išsilavinimas**  
**FROM Vykdytojai WHERE Kvalifikacija='Informatikas'**  
**AND Kategorija = 2**

Yra keletas ribojimų apibrėžiančios užklausoje.  
Pvz., užklausoje negalima naudoti **ORDER BY**  
Tačiau užklausoje virtualiajai galima:

6-21

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

### 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')
```

7-21

**Vertikali** virtualioji – realiosios stulpelių poaibis:

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

**Mišrieji rodiniai** – lentelės eilučių ir stulpelių poaibis.

Rodinyje *Informatikai* – mišrusis.

8-21

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

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

SELECT AVG(Visos_Valandos)
FROM Apie_Vykdyma
```

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
```

Apibrėžiančioje užklausoje galima kreiptis ir į virtualiąją,

```
CREATE VIEW Projektas1_Vykdymas
AS SELECT *
FROM Projektai_Vykdymas
WHERE Nr = 1
```

### 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'
```

Virtualiąją lentelę **galima atnaujinti**, jei apibrėžiančioji užklausa tenkina reikalavimus:

- nėra frazės **DISTINCT**;
- frazėje **FROM** – tik viena lentelė, kurią galima atnaujinti;
- kiekvienas stulpelis – paprastas lentelės stulpelis;
- paieškos sąlygoje nėra kitos užklauso;
- nėra duomenų grupavimo.

Tiksleni reikalavimai – DBVS dokumentacijoje

### 6.4. Virtualiųjų lentelių atnaujinimo valdymas

Virtuali lentelė – vykdytojai, kurių kategorijos > 2:

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

#### 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



#### 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

**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
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“

**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	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 Vykdytojai**  
**WHERE Kategorija > 2**  
**WITH CHECK OPTION**

– DBVS užtikrina, kad neatsirastų „nematomų“ eilučių.

**INSERT INTO Gudručiai**  
**VALUES (8, 'Juozaitis', 'Informatikas', 2, 'VDU')**  
 – **SQL klaida:** negalima įvesti duomenų

**WITH CHECK OPTION** neigiamai atsiliepia efektyvumui.

Virtualiųjų lentelių ypatumai **PostgreSQL**

- PostgreSQL (kol kas) neleidžia atnaujinti virtualiųjų lentelių, t.y. **INSERT, UPDATE, DELETE** – negalimi.
- Duomenų atnaujinimą galima modeliuoti nestandartiniu SQL sakiniu **CREATE RULE** apibrėžiant, kaip rodinio atnaujinimas keičiamas tikrų lentelių atnaujinimu.