

2.7a. Bendrumo ir egzistavimo kvantoriai užklausoje

Pateiksime ankstesnio uždavinio „vykdytojų, dalyvaujančių projekte Nr. 1, pavardės“ sprendimo dar vieną formuluotę, pavartojant predikatą – egzistavimo kvantorių:

```
SELECT Pavardė FROM Vykdytojai
WHERE EXISTS ( SELECT * FROM Vykdymas
               WHERE Vykdytojas = Nr AND Projektas = 1 ).
```

Šiame sakinyje išorinės užklauskos paieškos sąlygoje yra pavartotas predikatas, kurio sintaksė yra `EXISTS (SELECT * FROM...)`. Šio predikato reikšmė yra „tiesa“ tuomet ir tik tuomet, kai vidinės užklauskos rezultatas yra netuščioji aibė.

Pastaroji užklausa yra priklausomoji. Vidinės, predikate pavartotos užklauskos rezultatas priklauso nuo parametro *Nr*, kuris įgyja reikšmę išorinėje užklausoje. Šioje užklausoje projekto Nr. 1 vykdytoju yra laikomas kiekvienas vykdytojas, kuriam egzistuoja bent vienas vykdymas, kuriame jis dalyvauja vykdant tą projektą.

Paieškos sąlyga

```
EXISTS (SELECT * FROM...)
```

yra ekvivalentiška sąlygai

```
(SELECT COUNT (*) FROM...) > 0.
```

Šiame reiškinyje palyginimo operacijos vienas iš operandų yra užklausa. Kadangi užklauskos, kurioje nėra duomenų grupavimo ir `SELECT` frazėje yra tik jungtinė funkcija, rezultatas visuomet yra vienintelė reikšmė, tai tokią užklausą galima naudoti palyginimo operacijoje.

Predikatų logikos kvantorių atitikmenys naudojami ir tokiuose predikatuose

```
<reikšins> <palyginimo operacija> <ALL | ANY | SOME> ( <užklausa> )
```

Apskaičiuojant predikato `ALL` reikšmę lyginamojo reiškinio reikšmė yra lyginama su visomis užklauskos rezultata sudarančiomis reikšmėmis. Predikato `ALL` reikšmė yra „tiesa“ tuomet ir tik tuomet, kai užklauskos rezultatas yra tuščioji aibė arba palyginimo reikšmė yra „tiesa“ visoms užklauskos rezultato reikšmėms.

Predikato `ANY` reikšmė yra „tiesa“ tuomet ir tik tuomet, kai palyginimo reikšmė yra „tiesa“ bent vienai užklauskos rezultato reikšmei. Predikatas `SOME` yra predikato `ANY` sinonimas.

Darbuotojų kvalifikacijas, kuriose visi darbuotojai yra ne mažesnės negu 2-os kategorijos, galima sužinoti užklausa

```
SELECT DISTINCT A.Kvalifikacija FROM Vykdytojai A
WHERE 2 <= ALL (SELECT B.Kategorija FROM Vykdytojai B
               WHERE A.Kvalifikacija = B.Kvalifikacija).
```

Predikatas `ALL` labai palengvina tokio uždavinio sprendimą. Kad užklauskos rezultata sudarytų tik skirtingos kvalifikacijos, panaudojome frazę `DISTINCT`. Pagrindinės (išorinės) užklauskos sąlyga yra tikrina kiekvienai lentelės *Vykdytojai* eilutei. Kai šioje lentelėje yra labai daug eilučių užklauskos vykdymui gali prireikti nemažai laiko, nes vidinė užklausa yra priklausoma nuo parametro *A.Kvalifikacija*. Tarus, kad skirtingų kvalifikacijų yra žymiai mažiau negu darbuotojų, galima sudaryti efektyvesnį uždavinio sprendinį. Užklausoje galima pasinaudoti laikinąja lentele, kad vidinė užklausa būtų vykdoma tik skirtingoms kvalifikacijoms,

```
SELECT A.Kvalifikacija
FROM (SELECT DISTINCT Kvalifikacija FROM Vykdytojai) AS A
WHERE 2 <= ALL (SELECT B.Kategorija FROM Vykdytojai B
               WHERE A.Kvalifikacija = B.Kvalifikacija).
```

Jau žinomas vykdytojų, dalyvaujančius projekte Nr. 1, pavardės galima sužinoti dar ir taip:

```
SELECT Pavardė FROM Vykdytojai
WHERE Nr = ANY (SELECT Vykdytojas FROM Vykdymas WHERE Projektas = 1).
```

Nesunku pastebėti, kad ši užklausa struktūriškai yra labai panaši į užklausa, pateiktą ankstesniajame skyrelyje, panaudojant predikatą IN. Predikatą IN galima išreikšti per predikatą ANY, o predikatą NOT IN per predikatą ALL. Užrašas

<reiškinys> IN (<užklausa>)

yra ekvivalentiškas užrašui

<reiškinys> = ANY (<užklausa>),

o užrašas

<reiškinys> NOT IN (<užklausa>)

yra ekvivalentiškas užrašui

<reiškinys> <> ALL (<užklausa>).

Darbuotojų kvalifikacijas, kuriose visi darbuotojai yra ne mažesnės negu 2-os kategorijos, galima išreikšti be predikato ALL

```
SELECT DISTINCT A.Kvalifikacija FROM Vykdytojai A
WHERE (SELECT COUNT (*) FROM Vykdytojai B
      WHERE A.Kvalifikacija = B.Kvalifikacija AND A.Kategorija < 2) = 0.
```

Šioje užklausoje kvalifikacijos, kuriose visi darbuotojai yra ne mažesnės negu 2-os kategorijos, yra apibrėžtos kaip kvalifikacijos, kuriose nėra mažesnės negu antros kategorijos darbuotojų.

2.9a. Konstantinės lentelės

Standarte SQL2 yra numatyta galimybė apibrėžti lentelę betarpiškai SQL sakinyje. Tai - **konstantinės lentelės**. Tokioje lentelėje visos eilutės išvardijamos betarpiškai atskiriant jas tarpusavyje kableliais. Kiekvienos eilutės stulpelių reikšmės yra rašomos tarp rištinių skliaustų, atskiriant reikšmes kableliais. Konkretaus stulpelio reikšmė eilutėje pateikiama reiškiniu, paprasčiausiu atveju, konstanta. Konstantinės lentelės apibrėžiamos sakiniu VALUES.

Kadangi sakiniu VALUES iš reiškinių galima sudaryti lentelę, tai jis vadinamas lentelės konstruktoriumi. Šis sakinyss taip pat yra ypatinga užklausa. Sakinio rezultatas yra lentelė, kurios turinys yra nurodytas sakinyje. Pavyzdžiui, sistemos datą galima sužinoti tokia užklausa

```
VALUES (CURRENT DATE).
```

Lentelė (užklausa), sudaryta iš vienos eilutės, kurioje yra 3 datos: vakar, šiandien ir rytoj, sudaroma sakiniu:

```
VALUES (CURRENT DATE - 1 DAY, CURRENT DATE, CURRENT DATE + 1 DAY).
```

Lentelę su tais pačiais duomenimis, tačiau išdėstytais vieno stulpelio trijose eilutėse, galima išreikšti labai panašiai

```
VALUES (CURRENT DATE - 1 DAY), (CURRENT DATE), (CURRENT DATE + 1 DAY).
```

Jei sakinyje VALUES yra išvardyta n eilučių: $e_1, e_2, \dots, e_n, n > 1$, t.y.

```
VALUES (e1), (e2), ..., (en),
```

tai šis sakinyss yra ekvivalentiškas sakiniui

```
VALUES(e1) UNION ALL VALUES(e2) ... UNION ALL VALUES(en).
```

Todėl visos sakinyje VALUES išvardytos eilutės turi būti tarpusavyje suderintos – sudarytos iš vienodo skaičiaus reikšmių (stulpelių) ir atitinkamų reikšmių tipai neprieštarinti.

Sakiniu VALUES sudarytą lentelę, kaip kiekvienos užklauskos rezultatą, galima rūšiuoti ir grupuoti, pvz.,

```
VALUES (CURRENT DATE - 1 DAY), (CURRENT DATE), (CURRENT DATE + 1 DAY)
ORDER BY 1 DESC.
```

Sakinyje VALUES negalima suteikti stulpeliams pavadinimų. Tačiau, tai galima padaryti apibrėžiant papildomą laikinąją lentelę, pvz.,

```
SELECT Vakar, Šiandien, Rytoj FROM
  (VALUES (CURRENT DATE - 1 DAY), (CURRENT DATE), (CURRENT DATE + 1 DAY))
  AS Dienos(Vakar, Šiandien, Rytoj).
```

Sakinyss VALUES dažniausiai naudojamas vardinių konstantų (sistemos laiko, datos, vartotojo vardo ir kt.) reikšmėms sužinoti.