



**Code
Academy**



Dėstytojas

Vilmantas Neviera

DBMS

Data



Šiandien išmoksime

01

Platesnis WHERE panaudojimas

02

Group BY panaudojimas



Platenis WHERE paragrafo panaudojimas

Duomenų išrinkimui pagal užduotą sąlygą formuojama užklausa, naudojant paragrafą WHERE. Sąlyga gali būti pateikiama kaip aritmetinė išraiška ar kaip simbolių eilutė, surišta per lyginimo operatorius =, >, <, >=, <=, <>.

```
SELECT * FROM Emp WHERE Sal >= 1500;
```

```
SELECT Ename, Sal, Comm FROM Emp WHERE Sal <= Comm;
```

Sąlygoje dalyvauja du stulpeliai, bet nenurodyta konkreti reikšmė.

*Simbolinės eilutės arba data sąlygoje turi būti pateikta tarp viengubų kabučių.



Platenis WHERE paragrafo panaudojimas

Paragrafe **WHERE** gali būti formuojama sąlyga, naudojant sąlygos pateikimo būdus:

BETWEEN ... AND - kurie nurodo žemiausios ir aukščiausios reikšmių intervalą, išrenkant duomenis iš lentelės, t.y. rezultate pateikia duomenis tam tikram reikšmių intervalui.

SELECT Ename, Sal **FROM** EMP **WHERE** Sal **BETWEEN** 1000 **AND** 1500;

Išrenka darbuotojus, kurių atlyginimas yra tarp 1000 ir 1500.

BETWEEN ... AND operatorius apima ir kraštutines reikšmes. Mažesnioji reikšmė turi būti nurodyta pirma, didesnioji - antra.



Platenis WHERE paragrafo panaudojimas

IN (sąrašas reikšmių, kurios turi būti pateiktos užklausos rezultate, pateikiamos per kablelį)

SELECT Empno, Ename, Sal, Mgr **FROM** EMP **WHERE** Mgr IN(7902,7566);

Išrinks darbuotojus kurių laukas Mgr 7902 ir 7566.

Šią sąlygą galima būtų pateikti ir kitaip. **SELECT * FROM** EMP **WHERE** Deptno IN(10, 20); Išrinks darbuotojus, kurie dirba 10 ir 20 departamentuose.



Platenis WHERE paragrafo panaudojimas

LIKE (šablonas duomenų išrinkimui iš lentelės) Su LIKE galima naudoti šabloną, pagal kurį atrenkami duomenys iš lentelės. Sąlygoje gali būti pateikiami tik simboliai arba skaičiai. Galima naudoti du pagalbinius simbolius formuojant šabloną.

% - reiškia nulį arba keletą simbolių

_ - reiškia vieną simbolį

SELECT Ename **FROM** EMP **WHERE** Ename **LIKE** 'S%';

parodo visus darbuotojus, kurių vardas prasideda "S" raide, bet neparodys, kurių vardai prasidės "s" raide(mažąja).



Platenis WHERE paragrafo panaudojimas

SELECT Ename, Hiredate **FROM** EMP **WHERE** Hiredate **LIKE** '%81';

- visi darbuotojai kurie pradėjo dirbti 81 metais.

SELECT Ename **FROM** EMP **WHERE** Ename **LIKE** '_a%';

- visi darbuotojai, kurių varde bus antra raidė "A"



Platenis WHERE paragrafo panaudojimas

IS NULL - (nurodo, jog vartotoją domina tik eilutės lentelėje su neapibrėžta lauko reikšme)

NULL naudojamas norint pateikti duomenis su jokia reikšme.

SELECT Ename, Mgr **FROM** EMP **WHERE** Mgr IS NULL;

išrinks duomenis, kur laukas Mgr neužpildytas.

IS NULL tikrina nurodyto stulpelio reikšmę.

SELECT Ename, Job, Comm **FROM** EMP **WHERE** Comm **IS NULL**;

parodo visus darbuotojus, kurie negauna komisinio atlyginimo.



Platenis WHERE paragrafo panaudojimas

Sudėtinga sąlyga gali būti pateikta naudojant **AND** arba **OR**. Tokiu būdu galima pateikti kelias sąlygas duomenų išrinkimui.

SELECT Empno, Ename, Job, Sal **FROM** EMP **WHERE** Sal \geq 1100 **AND** Job = 'Clerk';

Rezultate pasirodys visi darbuotojai, kurie užima nurodytas pareigas ir jų alga viršija 1100.

SELECT Empno, Ename, Job, Sal **FROM** EMP **WHERE** Sal $>$ 1100 **OR** Job = 'Clerk';

Rezultate pasirodys visi darbuotojai, kurie užima nurodytas pareigas arba jų alga viršija 1100.



Platenis WHERE paragrafo panaudojimas

SELECT Ename, Job, Sal **FROM** EMP **WHERE** job = 'SALESMAN' **OR** job = 'PRESIDENT' **AND** sal > 1500;

Dvi sąlygos nurodant pareigas su atlyginimu, kai pareigos pateikiamos taip pat per sąlygą OR (arba) t.y. kai pareigos gali turėti skirtingas reikšmes, bet abiem atvejais atlyginimui keliami tie patys reikalavimai.

SELECT Ename, Job, Sal **FROM** EMP **WHERE** (job = 'SALESMAN' OR job = 'PRESIDENT') **AND** sal > 1500; Irgi dvi sąlygos, bet tikrinama pirma pareigybės ir jei yra viena iš nurodytų, toliau tikrinamas atlyginimas.



Platenis WHERE paragrafo panaudojimas

SELECT Ename, Job, Sal **FROM** EMP **WHERE** job = 'SALESMAN' **OR** job = 'PRESIDENT' **AND** sal > 1500;

Dvi sąlygos nurodant pareigas su atlyginimu, kai pareigos pateikiamos taip pat per sąlygą OR (arba) t.y. kai pareigos gali turėti skirtingas reikšmes, bet abiem atvejais atlyginimui keliami tie patys reikalavimai.

SELECT Ename, Job, Sal **FROM** EMP **WHERE** (job = 'SALESMAN' OR job = 'PRESIDENT') **AND** sal > 1500; Irgi dvi sąlygos, bet tikrinama pirma pareigybes ir jei yra viena iš nurodytų, toliau tikrinamas atlyginimas.



Pirmenybės, atliekant lyginimo operaciją

Eilės tvarka	Operatorius
1	Visi lyginimo operatoriai
2	NOT
3	AND
4	OR



Rūšiavimas ORDER BY

Išrenkant duomenis iš lentelės į ekraną, duomenų tvarka yra tokia, kokia jie buvo suvesti į lentelę. Norint pakeisti duomenų atvaizdavimo tvarką, operatoriuje SELECT galima naudoti paragrafą **ORDER BY**, kuriame nustatome, kaip turi būti pateikti duomenys užklausos rezultate. Standartiniu būdu, naudojant ORDER BY, duomenų vaizdavimas vyksta abėcėlės tvarka nuo A iki Z arba skaičių didėjimo tvarka.



Rūšiavimas ORDER BY

Norint pakeisti duomenų atvaizdavimo tvarką atvirkščiai, prie paragrafo ORDER BY naudojame papildomą žodį **DESC**.

Jei naudojame paragrafą ORDER BY, jis turi būti nurodomas operatoriaus SELECT gale.

SELECT * FROM EMP ORDER BY Hiredate;

Rūšiuoja pagal priėmimo į darbą datą

SELECT * FROM EMP ORDER BY Hiredate DESC;

Rūšiuoja taip pat pagal datą, bet atvirkštine tvarka.



Pagalbinės funkcijos

LOWER/UPPER(stulpelis|reikšmė) - bet koku formatu parašytus simbolius rezultate pateikia mažosiomis/didžiosiomis raidėmis. Labai naudinga nurodant sąlygą užklausoje kai reikia suvienodinti informacijos formatą duomenų atrinkimo metu. Ji naudinga aplikacijose pateikiant informaciją vienodam pavidale, jei ji yra suvesta skirtingam formate.

```
SELECT LOWER(VARDAS) FROM DARBUOTOJAS;
```

```
SELECT UPPER(VARDAS) FROM DARBUOTOJAS;
```



Pagalbinės funkcijos

CONCAT - apjungia dvi string reikšmes.

Pavyzdys:

```
SELECT CONCAT('vardas: ', vardas) FROM darbuotojas;
```

```
SELECT CONCAT(vardas, ' ', pavarde) FROM darbuotojas;
```




Grupinės funkcijos

Šios funkcijos veikia per visą lentelę arba per lentelės sritį ir tik tada pateikia bendrą rezultatą kiekvienai grupei atskirai arba visai lentelei. Pagal nutylėjimą visos lentelės eilutės skaitomos kaip viena grupė, jei nenurodyta kitaip. Paragrafas **GROUP BY** komandoje SELECT gali būti naudojamas padalinti lentelę į grupes. Grupinėse funkcijose NULL reikšmė ignoruojama. Grupinėms funkcijoms galima priskirti funkcijas:

AVG() - nustato vidurkį grupei

COUNT() - nustato eilučių, kurių išraiška yra apibrėžta, kiekį lentelėje

MAX() - nustato didžiausią reikšmę **MIN()** - nustato mažiausią reikšmę

SUM() - nustato bendrą sumą, ignoruojant neapibrėžtas reikšmes



Grupinės funkcijos

Funkcijos taikomos visai grupei, bet jei nurodyta `DISTINCT(MIN(DISTINCT SAL))`, tai funkcija taikoma tik skirtingoms reikšmėms, t.y. pasikartojančios reikšmės yra ignoruojamos. Visais atvejais, jei yra `NULL` reikšmė, tai įrašas ignoruojamas, išskyrus funkciją `COUNT`, kai naudojama `*`.

```
SELECT MAX(SAL), MIN(SAL), AVG(SAL) FROM Emp;
```

```
SELECT COUNT(*) FROM EMP WHERE DEPTNO = 20;
```

Norint sugrupuoti įrašus lentelėje, naudojame paragrafą **GROUP BY**

```
SELECT AVG(SAL) FROM Emp;
```

```
SELECT AVG(SAL) FROM Emp GROUP BY JOB;
```



Grupinės funkcijos

Grupės viduje galima grupuoti smulkiau. Tarkime skirtinguose departamentuose, norime grupuoti informaciją pagal pareigybes:

```
SELECT JOB, AVG(SAL) FROM Emp WHERE JOB = 'MANAGER' GROUP BY DEPTNO, JOB;
```

```
SELECT Deptno, Job, SUM(sal) FROM EMP GROUP BY Deptno, Job;
```

Pateikiant rezultatą pirmiausia duomenys grupuojami pagal departamento numerį, o po to jo viduje dar grupuojama pagal darbo pobūdį. Taip sutvarkius, sumuojamas bendras atlyginimas suformuotose grupėse.



Grupinės funkcijos

Grupinėse funkcijose yra svarbu išvedant duomenis, užklausą formuoti taip, kad nebūtų individualių laukų, kurie nėra įtraukti į paragrafą GROUP BY.

```
SELECT DEPTNO, MIN(SAL) FROM EMP;
```

- klaidinga užklausa, nes funkcija MIN yra grupinė, o laukas DEPTNO yra kiekvienam įrašui ir tai yra nesuderinama.

Teisingai būtų:

```
SELECT DEPTNO, MIN(SAL) FROM EMP GROUP BY DEPTNO;
```

Dabar DEPTNO jau yra grupės pavadinimas ir nebėra individualus laukas.



Grupinės funkcijos

Gali būti grupuojami duomenys, pateikiant juos kaip grupę grupėje. Naudojant grupines funkcijas yra galimybė išjungti arba įjungti iš rezultato atskiras duomenų grupes. Tam galime naudoti paragrafą **HAVING**, kuris naudojamas po GROUP BY. Turi tokią pačią prasmę kaip ir operatorius WHERE, tačiau kuris yra negalimas grupinėse funkcijose.



Grupinės funkcijos

Bendru atveju sintaksė būtų tokia:

SELECT column, group_function

FROM table

[**WHERE** condition]

[**GROUP BY** list_to_group]

[**HAVING** group_condition]

[**ORDER BY** sorting_condition]



Grupinės funkcijos

```
SELECT Job, Sum(Sal) TotalSum FROM EMP  
WHERE Job NOT LIKE 'SALES%'  
GROUP BY Job  
HAVING Sum(Sal) > 300  
ORDER BY Sum(Sal)
```

Rūšiuoja pagal gautą sumą grupėse. TotalSum sukuria naują stulpelio antraštę. Skaičiuoja visiems darbuotojams, išskyrus "SALES"



Grupinės funkcijos

Norint parodyti vidutinį atlyginimą darbuotojams, pagal sąlygą, jog mus domina tik departamentai, kuriuose dirba daugiau nei trys žmonės, formuosime užklausą:

```
SELECT DEPTNO, AVG(SAL) FROM EMP GROUP BY DEPTNO HAVING COUNT(*) > 3;
```

Norint parodyti tik darbus, kurių didžiausias atlyginimas yra ≥ 3000 , formuosime užklausą:

```
SELECT JOB, MAX(SAL) FROM EMP GROUP BY JOB HAVING MAX(SAL) >= 3000;
```




Grupinės funkcijos

Nesant grupavimo, geriau atrankos sąlygai naudoti paragrafą WHERE. Žemiau pateikiamas pavyzdys yra neteisingas, nes grupinė funkcija naudojama poroje su paragrafu WHERE

```
SELECT DEPTNO, AVG(SAL) FROM EMP GROUP BY DEPTNO WHERE AVG(SAL) > 2000;
```

Teisingai būtų taip:

```
SELECT DEPTNO, AVG(SAL) FROM EMP GROUP BY DEPTNO HAVING AVG(SAL) > 2000;
```



Grupinės funkcijos

```
SELECT Max(a.salary) FROM (Select Avg(BaseRate)salary FROM [AdventureWorksDW2019].[dbo].[DimEmployee]  
GROUP BY Title) a;;
```

Pirma suskaičiuoja atlyginimo vidurkį departamente, o po to išrenka didžiausią vidurkį tarp departamentų.

**Užduotis nr. 1**

- 1.Išrinkite duomenis apie darbuotoją (asmens kodą, vardą ir pavardę) iš lentelės DARBUOTOJAS kurie būtų gimę 1988m liepos 20d.
- 2.Išrinkite visus duomenis apie darbuotojus iš lentelės DARBUOTOJAS, kurie yra gimę iki 1988m liepos 29d
- 3.Išrinkite duomenis apie darbuotojus (dirba nuo kada ir gimimo metus) iš lentelės DARBUOTOJAS, kurie būtų įsidarbinę nuo 2009m spalio 30d iki 2012m lapkričio 11d.
- 4.Išrinkite duomenis apie darbuotojus (vardą, Skyrių ir Projekto ID) iš lentelės DARBUOTOJAS kurie dirba 2 ir 3 projektuose. (Panaudoti IN operatorių).
- 5.Išrinkite duomenis (vardą, pavardę ir asmens kodą) apie visas moteris iš lentelės DARBUOTOJAS (panaudojant operatorių LIKE).
- 6.Išrinkite visus duomenis apie visus darbuotojus iš lentelės DARBUOTOJAS, kurie yra gimę 12 diena (panaudojant operatorių LIKE).
- 7.išrinkite visus projektus iš lentelės PROJEKTAS kad projekto pavadinime 3 raidė būtų 'u'.

**Užduotis nr. 2**

- 8.Išrinkite visus darbuotojus iš lentelės DARBUOTOJAS, kuriems nepaskirtos jokios pareigos.
- 9.Išrinkite duomenis apie darbuotoją (vardą, pavardę, nuo kada dirba ir pareigas) kad tenkintų sąlygas: (dirba nuo 2011-02-12 ir jų pareigos yra Programuotojai).
- 10.Išrinkite duomenis apie darbuotojus (vardą, pavardę, skyriaus pavadinimą ir projekto ID) iš lentelės DARBUOTOJAS su sąlyga, kad jie būtų iš Java skyriaus arba 1 projekto.
- 11.Išrinkite visus darbuotojų vardus išskyrus tuos, kurių vardai prasideda raide 'S' .
- 12.Išrinkite duomenis (vardą, dirba nuo kada ir gimimo metus) iš lentelės "DARBUOTOJAS", apie visus darbuotojus tik ne tuos, kurie įsidarbino nuo 2009m spalio 30d iki 2012m lapkričio 11d.
- 13.Išrinkite duomenis apie darbuotojus (vardą, pavardę ir gimimo metus) iš lentelės DARBUOTOJAS ir išrikiuokite visus duomenis nuo seniausio žmogaus iki jauniausio.
- 14.Išrinkite duomenis apie darbuotojus (vardą, pavardę ir gimimo metus) iš lentelės DARBUOTOJAS ir išrikiuokite visus duomenis nuo jauniausio žmogaus iki seniausio.

**Užduotis nr. 3**

- 15.Išrinkite iš lentelės DARBUOTOJAS projekto id kuris būtų minimalus skaičius ir maksimalus skaičius.
- 16.Išrinkite duomenis apie projektą ir kiek tame projekte yra priskirta žmonių iš lentelės DARBUOTOJAS (projekto numeris ir skaičius kiek dalyvauja žmonių).
- 17.Išrinkite duomenis (projekto numeris, pareigos, skaičius) iš lentelės DARBUOTOJAS kiek dirba programuotojų kiekvienam projekte.
- 18.#17 punkto užklausa pataisykite taip, kad rodytų tik tuos projektus, kur dirba bent 2 darbuotojai.

Paskaitos pavadinimas



Headline

www.youtube.com

**Naudinga
informacija**