

Klauzule GROUP BY i HAVING

Jedan specijalan način upotrebe operatora GROUP BY je povezivanje neke agregatne funkcije sa grupom redova (ovo se naročito odnosi na funkciju COUNT koja broji redove u svakoj grupi).

Na primer imamo bazu podataka o svim pošiljkama i vrsti prevezenog tereta. Želimo da saznamo kolika je masa prevezena po pojedinim vrstama robe.

The screenshot shows the MySQL Query Browser interface. The query executed is `SELECT * FROM tranzit t;`. The result set contains 48 rows of data. The columns are: `spremnik_posilke`, `sifra_posilke`, `sifra_preuzeca`, `vrst`, `broj_kola`, `Masa_kg`, `sifra_tereta`, and `prihod`. The data shows various shipments with their respective weights and revenue.

spremnik_posilke	sifra_posilke	sifra_preuzeca	vrst	broj_kola	Masa_kg	sifra_tereta	prihod			
11028	23499	5529555	98	1	4	I	1	8000	870400	32430
11028	23499	5510323	98	1	4	D	17	704520	100590	13500
11028	23499	5519182	98	1	4	D	16	733060	100590	13500
11028	23499	5502246	98	1	4	D	16	784000	441000	9199,3
11028	23499	5502246	98	1	4	D	16	784000	720400	9199,3
11028	23499	5502246	98	1	4	D	16	784000	441000	9199,3
12498	23499	5512567	99	1	4	I	1	51572	380240	8563,5
12498	23499	5512567	99	1	4	I	1	51572	380210	8563,5
11028	23499	5519182	98	1	4	D	17	734570	100590	8364
11028	23499	5519182	98	1	4	D	17	710420	100590	8364
11028	23499	5518089	98	1	4	D	16	689000	100590	8364
11028	23499	5519182	98	1	4	D	15	756790	100590	8364
11028	23499	5518036	98	1	4	D	15	753290	100590	8364
11028	23499	5518655	98	1	4	D	15	718690	100590	8364
11028	23499	5519182	98	1	4	D	15	609160	100590	8364
11028	23499	5519182	98	1	4	D	14	586030	100590	8364
11028	23499	5506502	99	1	4	D	19	703080	110700	8363
11028	23499	5506502	99	1	4	D	17	665250	110710	8363
11028	23499	5502246	98	1	4	D	16	784000	441000	8363
11028	23499	5502246	98	1	4	D	14	686000	441000	8363
11028	23499	5506502	99	1	4	D	10	400140	110700	8363
11028	23499	5516568	98	1	4	I	1	48400	100100	8363
11028	12498	5272206	98	1	4	I	1	42430	270400	8200
11028	12498	5272206	98	1	4	I	1	40700	270400	8200
11028	12498	5272206	98	1	4	I	1	39250	270400	8200
11028	12498	5272206	98	1	4	I	1	37880	270400	8200

`SELECT SUM(masa_kg_) , sifra_tereta FROM transport GROUP BY sifra_tereta`

Dobicemo kao rezultat sledecu tabelu

SUM(masa_kg_)	sifra_tereta
4855900	40100
9141961	40120
45300	40130
49190	40390
92200	70110
41000	71000
50000	71310
82350	71400
40200	90300
2792300	100100
723420	100110
5654870	100190
203960	100300
3656040	100500
50870	100580
10452060	100590
80000	110290
50000	110300

Medjutim, neka nas npr. interesuje ukupan prihod po vrsti robe za prevezene pošiljke mase veće od 50000 kg koristićemo klauzulu HAVING

SELECT SUM(prihod) , masa_kg_ FROM transport GROUP BY sifra_tereta having masa_kg_>50000

The screenshot shows the MySQL Query Browser interface. The title bar indicates the connection is 'root@localhost:3306 / test'. The menu bar includes File, Edit, View, Query, Script, Tools, Window, and Help. The toolbar contains various icons for navigation and execution. The SQL Query Area contains the following query:

```
1 SELECT SUM(prihod) , masa_kg_ FROM baza GROUP BY sifra_tereta having masa_kg_>50000
```

The Results window displays the following data:

SUM(prihod)	masa_kg_
72620	51000
0	50200
5877	52660
62667	51020
12928	51200
46682	734570
0	50870
136026	704520
76205	703080
30259	665250
53573	53100
55680	51162
0	51162
15850	50230
12464	50150

PODUPITI

Još jedna česta upotreba podupita odnosi se na primenu operatora u uslovima klauzule WHERE koji sadrže rezultat upita SELECT nekog podupita. Na primer, prikažimo transportna kola po kategoriji kod kojih je prihod od prevoza je za 100 veći od prosečnog prihoda ostvarenog od ukupnog prevoza:

```
SELECT vrsta_prevoza
FROM ANTIKVITETI
WHERE prihod >
      (SELECT AVG(prihod) + 100
      FROM transport);
```

Podupit izračunava prosečnu cenu plus 100, a zatim se prikazuju kategorije transportnih kola za svaki prevoz kod koga je prihod bio veći od prosečnog za 100 din. Da bismo eliminisali duplikate, mogli smo da navedemo DISTINCT vrsta_prevoza.

Napomena za podupite: kada je podupit deo nekog WHERE uslova, klauzula SELECT u tom podupitu mora imati kolone koje se poklapaju po broju i tipu sa onima koje se nalaze u klauzuli WHERE spoljašnjeg upita. Drugim rečima, ako imate "WHERE ImeKolone = (SELECT...);", klauzula SELECT mora sadržati samo jednu kolonu koja odgovara koloni ImeKolone u spoljašnjoj klauzuli WHERE i njihov tip mora biti odgovarajući (oba tipa su celi brojevi, znakovne niske itd.).

PRIMER

Potrebno je da se izračuna ukupan prihod po svim tranzitnim pravcima:

```
SELECT SUM(prihod), ulazna_stanica FROM transport GROUP BY ulazna_stanica
```

MySQL Query Browser - Connection: root@localhost:3306 / test

File Edit View Query Script Tools Window Help

Transaction Explain Compare

SELECT FROM WHERE GROUP HAVING ORDER SET CREATE

Resultset 1

SQL Query Area

```
1 SELECT SUM(prihod), ulazna_stanica, granicni_prelaz
2 FROM tranzit GROUP BY ulazna_stanica
```

	SUM(prihod)	ulazna_stanica	granicni_prelaz
▶	63528	11028	12498
	1337707	12498	11028
	1239568	16201	11028
	62093	16319	11028
	1056399	16517	12498
	461738	21099	11028
	2019321	23499	11028

7 rows fetched in 0.0019s (0.1394s)

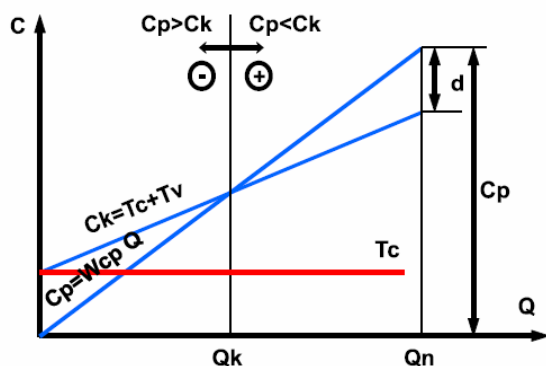
Edit Apply Changes Discard

Description

Na osnovu ovakvog prikaza lako je odrediti na kom pravcu je ostvaren maksimalni prihod.

ANALIZA PROFITABILNOST & MySQL

Podsetnik: Q-C dijagram za određivanje kritične tačke



Produktivnost = količnik broja proizvedenih proizvoda i broja radnika

Ekonomičnost = odnos prodajne cene i cene koštanja

Rentabilnost = odnos dobiti i angažovanih sredstava

Po analogiji sa proizvodnjom za preduzeće čija je osnovna delatnost transport potrebno je odrediti kritični obim 'rada' odnosno potrebno je proračunati profitabilne pravce, vrste robe i sl.

U tabeli transport postoji kolona 'prihod' iz koje se može izračunati ukupni prihod po vrsti transportnog sredstva ili transportnom pravcu ili sifri robe, međutim u datoj tabeli ne postoje podaci o troškovima. U zasebnim tabelama nalaze se podaci o utrošenom gorivu, materijalnim troškovima, troškovima održavanja i troškovima radne snage (lični dohodak + dodatna sredstva) na mesečnom nivou.

```
SELECT sum(prihod), vrsta_prevoza FROM transport GROUP BY vrsta_prevoza
```

naredba računa ukupno ostvaren prihod po vrsti prevoza odnosno kategoriji prevoznog sredstva

```
SELECT SUM(trosak1), vrsta_prevoza FROM gorivo GROUP BY vrsta_prevoza
```

naredba računa ukupan trosak za gorivo po vrsti prevoza odnosno kategoriji prevoznog sredstva

na sličan način može se izračunati ukupan trošak za održavanje, trošak radne snage, itd. iz odgovarajućih tabela.

```
SELECT sum(transport.prihod)-sum(tabela1.trosak1)+sum(tabela2.trosak2) +... FROM  
transport, tabela1, tabela2
```

može se izračunati poslovni dobitak, a grupisanjem po kategoriji transportnog sredstva (ili drugom parametru od interesa) dobiti detaljniji uvid u strukturu prihoda i troškova.

Problem u praksi nastaje kada se podaci u tabelama vode po različitim kategorijama..npr. u tabeli radnici mogu se nalaziti podaci o mesečnim primanjima radnika pri čemu se onda ne zna pojedinačan trošak radne snage po prevoznom sredstvu ili npr. punom pravcu i sl. a ukoliko ne postoji prihod i trošak po sredstvu rada ne može da se odredi njegova ekonomičnost.