

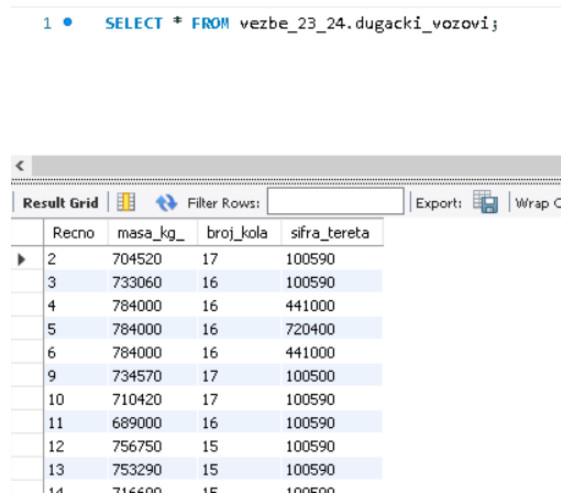
# Baze podataka – V5

## Kreiranje pogleda:

Kako bi pregled podataka u spojenim tabelama bio lakši, u sql kodu postoji mogućnost kreiranja fiktivne tabele koja će sadržati sve zadate podatke, ali neće zauzimati prostor na serveru. Korišćenjem naredbe CREATE VIEW ime\_pogleda AS i potom korišćenja klasične naredbe SELECT u stablu sheme se kreira nova tabela koja sadrži sve podatke obuhvaćene naredbom select, a nalazi se u podkartici Views označene sheme, sa imenom koje mu je definisano u naredbi. Svi podaci koji se nalaze u određenom pogledu su u direktnoj vezi sa podacima tabela iz kojih su uzeti i pri izmeni podataka u tabelama, takođe se i podaci u pogledima menjaju automatski. Primer kreiranja pogleda se izvršava naredbom:

```
CREATE VIEW Dugacki_vozovi AS
```

```
SELECT Recno, masa_kg_, broj_kola, sifra_tereta FROM tranzit WHERE broj_kola >= 10 ;
```



1 • SELECT \* FROM vezbe\_23\_24.dugacki\_vozovi ;

Recno	masa_kg_	broj_kola	sifra_tereta
2	704520	17	100590
3	733060	16	100590
4	784000	16	441000
5	784000	16	720400
6	784000	16	441000
9	734570	17	100500
10	710420	17	100590
11	689000	16	100590
12	756750	15	100590
13	753290	15	100590
14	716600	15	100590

Slika 1.

Pogled se može izmeniti naredbom CREATE OR REPLACE VIEW ime\_pogleda AS ili se mogu obrisati naredbom DROP VIEW. Pogledi se najčešće koriste pri obradi podataka u spojenim tabelama, kako bi se sintaksa koda značajno uprostila.

## Naredba HAVING:

Pri radu sa agregatnim funkcijama (SUM, COUNT, AVG..) ako je potrebno dodatno filtriranje podataka, a uslov za to se odnosi na kolonu dobijenom agregacijom, ne može se koristiti naredba WHERE već je neophodno nakon segmenta GROUP BY staviti naredbu HAVING i u njoj definisati novi uslov. Primer koda je:

```
SELECT SUM(prihod) AS Ukupni_prihod, vrsta_transporta FROM tranzit GROUP BY vrsta_transporta HAVING SUM(prihod) > 45000;
```

	Ukupni_prihod	vrsta_transporta
▶	3711365.1	I
	968747.9	D
	178787.0	G
	846670.2	0
	613393.4	R

Slika 2.

### Naredba EXIST:

U slučaju kada je potrebno izlistati upise iz jedne tabele koji ispunjavaju uslov iz druge, povezane, tabele, umesto izvršavanja u nekoliko koraka, postupak je moguće izvršiti atributom EXISTS u naredbi WHERE. Primer koda je:

```
SELECT Naziv FROM Sifrerobe WHERE EXISTS (SELECT sum(masa_kg_) FROM tranzit WHERE
tranzit.sifra_tereta = sifrerobe.sifra group by sifra_tereta HAVING SUM(masa_kg_) > 90000);
```

	Naziv
▶	MLEKO, PAVLAKA, NEKONCENTROVANI, NEZAS...
	KROMPIR, SVEZ
	POVRCE MAHUNASTO, SUSENO
	PSENICA, NAPOLICA
	JECAM
	KUKURUZ
	GRIZ, PREKRUPA (JARMA), PELETE, OD ZITARI...
	SLAD
	ULJE OD SUNCOKRETA, ULJE OD SAFRANIKE, ...
	ULJE OD PERICE, ULJE OD PEPE, ULJE OD SLA

Slika 3.

### Naredba UNION:

Za razliku od naredbe JOIN koja u zavisnosti on varijante spaja upise iz dve tabele i zadržavajući kolone, naredba UNION uzima sve upise iz jedne i druge tabele i postavlja ih u istu kolonu. Ako se određeni podatak ponavlja, u spojenoj tabeli se upisuje samo jednom. Primer primene ove naredbe može biti kreiranje imenika svih kupaca i dobavljača iz zasebnih tabela:

```
SELECT Name, Phone_number FROM Customers UNION SELECT Name, Phone_ number FROM
Suppliers ORDER BY Name;
```

Treba voditi računa o tipu podataka u kolonama, kao i o redosledu pisanja, kako ne bi došlo do pogrečnog spajanja.