**DISTINCT** - eliminiranje duplih vrijednosti. Nema efekta na funkcije MIN i MAX, ali ima na SUM i AVG.

Iskaz **GROUP BY** koristi se za grupisnje redaka u manje skupine. Često se koristi za dobivanje zbrojnih informacija za svaku grupu.

## Bilo koji stupac ili izraz u SELECT listi koji nije agregatna funkcija mora biti i u GROUP BY iskazu!

Iskaz **HAVING** koristi se ako se želi specifirati koje grupe treba prikazati, odnosno, koristi se za restrikciju gurpa koje se prikazuju, tj. za ispitivanje vrijednosti agregatnih funkcija.

Ne može se koristiti **WHERE** za restrikciju grupa, **WHERE** je samo za restrikciju pojedinačnih redaka.

**Osnovni načini spajanja su:**

* EQUI JOIN – spajanje po jednakosti se koristi FROM ili WHERE iskaz za kreiranje i spajanje. Upoređuju se vrijednosti stupaca spajanja iz obje tablice i moraju biti jednake da bi se izvršilo spajanje.
* NON-EQUI JOIN – spajanje po nejednakosti. Koristi se kada se vrijednosti nalaze u okviru nekog raspona i ne postoji izravna veza između stupaca u tablicama. Koristi se operator različit od =
* OUTER JOINS – koristi se za prikaz redaka koji inače ne zadovoljavaju uvjet spajanja.
* SELF JOINS- rekurzivna veza tablica se veže sama na sebe

Podupit je **SELECT** iskaz ugnježden u drugom **SELECT** iskazu.

Podupit se može koristi na bilo kojem dijelu **SELECT** naredbe, kao i u **INSERT**, **UPDATE**, **DELETE** iskazima.

**3 osnovna načina rada s podupitima:**

* kao dio uvjeta – postaje dio kriterija za odabir, podupit kao dio WHERE ili HAVING izraza
* kao dio listaTablica- specifizira izvor podataka, podupit dio FROM izraza
* kao dio listaStupca - kreira jedan izračunati stupac, podupit kao dio SELECT, GROUP BY ili ORDER BY izraza

**Pogled (view)** je objekat u bazi podataka koji predstavlja virtualnu tablicu. Pogled se stvara kreiranjem upita, a podaci koje taj upit vraća su podaci koji će biti dostupni putem pogleda.

**Osnovna pravila vezana za promjenu podatka kroz pogled:**

**Brisanje je zabranjeno ako pogled sadrži:**

* joins
* grupne funckije i GROUP BY

**Izmjena je zabranjena ako pogled sadrži:**

* sve navedeno i stupce definisane kao izraze

**Dodavanje je zabranjeno ako pogled sadrži:**

* sve navedeno i bilo koji nedostajući NOT NULL stupac ovisno o CHECK opciji

**UNION operator**

Ima 3 preduvjeta:

* svi SELECT izrazi kombinovani sa UNION moraju imati isti broj stupaca u SELECT listi
* svi stupci u savkoj SELECT listi moraju imati isti redoslijed
* svi odgivarajući stupci u svim SELECT listama moraju imati iste ili kompatibilne tipove podataka

**UNION vs JOIN**

* UNION spaja rezultat dva set upita, ali ne kreira individualne retke od stupaca odabranih tablica
* JOIN upoređuje stupce iz dvije tablice kako bi kreirao rezultat retke koje čine stupci iz obje tabele
* UNION ALL uključuje duplikate
* UNION isključuje duplikate

**INTERSECT** vraća podatke oba SET-a

**EXCEPT** vraća podatke koji su u jednom setu (lijevo), a nisu u drugom (desno)

**SQL Procedure mogu:**

* Prihvatiti input parametre i vratiti višestruke vrijednosti u obliku output parametra proceduri ili batch obradi
* Sadržavati programske izraze koji izvršavaju operacije nad bazom podataka i pozivati druge procedure
* Vraćati statusne vrijednosti proceduri koja ih je pozvala ili batch obradi kako bi ukazali na uspjeg i neuspjeh izvršenja

## OKIDAČI

Poseban oblik pohranjene procedure koja se izvršava automtski kada se dogodi odgovarajući događaj na serveru baze podataka.

**Vrste okidača:**

* **DDL TRIGERI** – odgovor na DDL događaje (CREATE, ALTER, DROP)
* **DML TRIGERI** –odgovor na DML događaje (INSERT, UPDATE, DELETE)
* **LOGON TRIGERI** – odgovor na LOGON događaje

Naredba **COMMIT** završava transakciju.

***Metode zaključavanja:***

Procedura koja se koristi za kontrolu konkurentnog pristupa podacima. Kada jedna transakcija pristupa bazi podataka, zaključivanje može spriječiti pristup drugim transakcijama kako bi izbjegli netačni rezultati.

**Shared lock** – transakcija može čitati podatak, ali ga ne može ažurirati

**Exclusive lock** – transakcija može i čitati i ažuritati podatke

**Deadlock** – mrtva tačka koja nastaje kada dvije (ili više) transakcija, obje (sve) čekaju da se otpusti zaključavanje, koje drži druga transakcija.

**TIMESTAP** – jedinstveni identifikator kreiran od strane DBMS-a koji pokazuje relativno početno vrijeme transkacije