**Predavanje za 27.3.2019.**

Poslednji čas smo radili u okviru naredbe SELECT definicije operatora IN, BETWEEN, LIKE, ORDER BY, DISTINCT i LIKE.

Da ponovimo:

**1. Operator IN**

- koristi se za generisanje upita sa uslovom da svi rezulati upita zadovoljavaju uslov pripadnosti skupu vrednosti -IN(skup vrednosti).

Primer:

SELECT ime, prezime, brindexa, godina\_upisa FROM student WHERE godina\_upisa IN(2018,2019)

daće prikaz imena, prezimena, brojeva idneska i godinu upisa iz baze podataka student pri ćemu su svi studenti upisani ili 2018 ili 2019 godine.

**2. Operator BETWEEN**

- koristi se za generisanje upita sa uslovom da svi rezultati upita zadovoljavaju uslo pripadnosti skupu vrednosti od - do za definisanu kolonu

Primer:

SELECT ime, prezime, brindexa, godina\_upisa FROM student WHERE godina\_upisa BETWEEN 2016 AND 2019

daće nam prikaz imena, prezimena, brojeva idneska i godinu upisa iz baze podataka student pri ćemu su svi studenti upisani u periodu od 2016 do 2019. (Računaju se i granične vrednosti, Znači sve slogove koji imaju u koloni godina upisa vrednost 2016, 2017, 2018 i 2019).

**3. KLAUZULA ORDER BY**

- koristi se za sortiranje rezulata upita po određenom redosledu

Primer:

SELECT ime, prezime, brindexa, godina\_upisa FROM student ORDER BY prezime

daće nam prikaz imena, prezimena, brojeva idneska i godinu upisa iz baze podataka student, soritirane po prezimenima studenata

\* ORDER BY ASC - rasutći niz

 ORDER BY DESC - opadajući niz

Primer:

SELECT ime, prezime, brindexa, godina\_upisa FROM student ORDER BY prezime DESC

daće nam prikaz imena, prezimena, brojeva idneska i godinu upisa iz baze podataka student, soritirane po prezimenima studenata u opadajućem nizu (od Z ka A)

**4. Klauzula DISTINCT**

- koristi se za eliminaciju duplikata. Odnosno prikazuje različite vrednosti u okviru određenog polja.

Primer:

SELECT DISTINCT godina\_upisa FROM student

 prikazaće vam samo koliko ima različitih godina upisa u bazi student i koje su to godine.

Npr.

2016

2017

2018

2019

 Korišćenjem ove klauzule možete da imate uvid u sadržaj određene baze podataka. Npr. za bazu tranzit je pogodna naredba da vidite koliko imate različitih vrsta prevoza ili npr. različitih uputnih stanica i sl.

**5. Operator LIKE**

- omogućava traženje odnosno poređenje vrednosti kolone sa zadatom vrednošću.

- osnovni oblik: prezime LIKE 'Milan' … u okviru kolone ime traži podatke u kojima se nalazi reč Milan

 - oblik prezime LIKE 'Milan%' ..znak % zamenjuje nisku bilo kojih i bilo koliko drugih karakternih ili numeričkih simbola. Rezulat ovog upita mogu biti reči - Milan, Milanović, Milančevic, itd. važno je napomenuti da se karakteri nalza iza reči Milan

- Oblik prezime LIKE '%Milan%' daće za rezultat sve slogove koji u koloni prezime imaju na bilo kom mestu nisku Milan.

Primer:

SELECT ime, prezime, brindexa, godina\_upisa FROM student WHERE ime LIKE 'ana'

rezultat mogu biti imena studenata sa niskom ana, kao na primer: LJiljana, Ana, Bojana itd.

- poseban oblik naredbe LIKE je kada umesto znaka % koji zamenjuje bilo koji niz karaktera, koristimo znak pitanja (?) koji nam zamenjuje samo jedno karakterno mesto.

Primer:

SELECT ime, prezime, brindexa, godina\_upisa FROM student WHERE brindexa LIKE '?7%'

prikazaće nam sve studente koji imaju u broju indeksa na drugom mestu broj 7.

**NOVI DEO PREDAVANJA**

**1. KLAUZULA GROUP BY**

korsti se za grupisanje reuzltata. Kada se koristi zajedno sa nekom agregatnom funkcijom (max, min, sum, count) daje nam sumarne rezultate za svaku grupu.

Na primer

SELECT COUNT(godina\_upisa), godina upisa FROM student GROUP BY godina\_upisa

daće nam broj studenata po godini upisa

|  |  |
| --- | --- |
| COUNT(godina\_upisa) | godina upisa |
| 45 | 2016 |
| 34 | 2017 |
| 48 | 2018 |
| 36 | 2019 |

ako dodamo AS iza agregatne funkcije, mozemo da dodelimo naziv novoj koloni

Na primer:

SELECT COUNT(godina\_upisa) **AS Ukupan\_broj**, godina upisa FROM student GROUP BY godina\_upisa

daće nam broj studenata po godini upisa

|  |  |
| --- | --- |
| Ukupan\_broj | godina upisa |
| 45 | 2016 |
| 34 | 2017 |
| 48 | 2018 |
| 36 | 2019 |

2. Klauzula HAVING

koristi se za definisanje kriterijuma u okviru klauzule GROUP BY . (znaci nema WHERE u slučaju group by)

Primer:

SELECT COUNT(godina\_upisa) **AS Ukupan\_broj**, godina upisa FROM student GROUP BY godina\_upisa HAVING COUNT(godina\_upisa)>40

daće nam samo grupe u kojima ima više od 40 studenta po godini

|  |  |
| --- | --- |
| Ukupan\_broj | godina upisa |
| 45 | 2016 |
| 48 | 2018 |

3. Klauzula JOIN

koristi se za spajanje dve ili više tabela po zajedničkom ključu

Na primer

Select \* FROM student JOIN ocene ON student.brindeksa=ocene.brojindeksa

daće nam novu tabelu u kojoj su pridruženi svi podaci iz tabele student sa podacima iz tabele ocene po koloni broj indeksa.

Vezbajte ovo sa Andrijom.

Imamo INNER JOIN - presek dva skupa. Odnosno izdvaja sve brojeve indeksa koji su zajednički u tabelama student i ocene, povezuje ih i to prikazuje

LEFT JOIN - uzima brojeve indeksa iz leve tabele (student) i njima pridodaje odgovarjuće brojeve indeksa iz desne tabele (ocene) i rezultat prikazuje. Ako u levoj tabeli ima neki br indeksa kojeg nema u desnoj tabeli izbacuje rezultat nula.

RIGHT JOIN - isto kao left join samo sa desne strane

FULL JOIN - daje rezultat i LEFT JOIN i RIGHT JOIN istovremeno. Ako nema poklapanja, daće NULL vrednost.

**PRIPREMA ZA TEST**

1. Idite na sajt http://elearning.rcub.bg.ac.rs/moodle/

2. Prijavite se



3. Kad vam odobre prijavu (stići će na mail za dan dva) idite opet na sajt i prijavite se za kurs BAZE PODATAKA- kopiranje 1 (ovo kopiranje su oni dodali u nazivu posto jos na nekom fakultetu ima predmet Baze podataka)

4. Šifra je: BPMM

sledeći korak je da vas sve dodam, treba da mi se pojavite u grupi



a izgleda ću to morati ručno. Zato da bih vas nasla potrebe su mi vaše mail adrese. Molim da mi svako posalje svoj mail, ime i prezime na mail: mmisita@mas.bg.ac.rs

Imamo razne alate na raspolaganju, kao što možete da vidite na slici.



Za početak da se pojavite svi na spisku korisnika, pa ću vam dati sledeće uputstvo.

Razmiljam da prvi test iz baza bude u vidu upitnika, a da odgovarate na pitanja u vidu odgovora pod a,b ili c.

Kao na primer da li je DML:

- jezik za manipulaciju podacima

- jezik za upravljanje podacima

- jezik za definiciju podataka.

Svaki drugi oblik pitanja bio bi copy-paste. Vreme za odgovore je ograničeno. Napravićemo i probnu simulaciju pre samog testa.

Znači prvi korak registracija na moodle, drugi korak saljite svoje mailove na moj mail. Napisite da je prijava za Baze podataka, posto imam više predmeta. Pozdrav svima!