Baze skripta

[1. Ishod 4. 2](#_Toc138782574)

[1.1 SELECT 2](#_Toc138782575)

[1.1.1 FROM 2](#_Toc138782576)

[1.1.2 WHERE 3](#_Toc138782577)

[1.1.3 ALIAS / AS 4](#_Toc138782578)

[1.1.4 DISTINCT 4](#_Toc138782579)

[1.1.5 Sortiranje 4](#_Toc138782580)

[1.1.6 TOP 5](#_Toc138782581)

[1.2 JOIN / SPAJANJE TABLICA 5](#_Toc138782582)

[1.3 EDITING 6](#_Toc138782583)

[1.3.1 Ubacivanje 6](#_Toc138782584)

[1.3.2 UPDATE 7](#_Toc138782585)

[1.3.3 DELETE 7](#_Toc138782586)

[2. Ishod 5 8](#_Toc138782587)

[2.1 Matematičke funkcije 8](#_Toc138782588)

[2.2 Datumske i vremenske funkcije 9](#_Toc138782589)

[2.3 Stringovske funkcije 9](#_Toc138782590)

[2.4 Ostale funkcije 10](#_Toc138782591)

[2.5 Agregatne funkcije 10](#_Toc138782592)

[2.6 Grupiranje 11](#_Toc138782593)

[3. ISHOD 6 12](#_Toc138782594)

[3.1 Podupiti 12](#_Toc138782595)

[3.1.1 PODJELA PODUPITA 13](#_Toc138782596)

[3.1.2 Konstantni podupiti i korelirani podupiti 13](#_Toc138782597)

[3.1.3 Podupiti u SELECT 14](#_Toc138782598)

[3.1.4 Podupiti u WHERE/HAVING 14](#_Toc138782599)

[4. Kako koristiti funkcije u MSMS-u 15](#_Toc138782600)

# Ishod 4.

## SELECT



### FROM

Tablica, spojene tablice, virtualna tablica, tablična funkcija…, koristimo kako bi specificirali odakle želimo uzeti podatke.

* Sintaksa

Generalno uvijek ista, „FROM [ODAKLE UZIMAMO PODATKE]“

FROM Ucenik

FROM Grad

FROM Drzava

### WHERE

Koristi se za filtriranje rezultata, odnosno specificiramo koje uvjete postavljamo za rezultat.

* Sintaksa

WHERE Ime = 'Ivan'

WHERE Cijena >= 500.00

WHERE BrojRacuna != 10

WHERE Ime = 'Ivan' AND Prezime = 'Chernobyl'

WHERE Ime = 'Ivan' OR Ime = 'Marin'

WHERE NOT IME = 'Tomo'

WHERE Prezime = 'Senjanin' AND Ime= 'Tomo' OR Ime = 'Tomislav'

* Pazite s ovim jer će ovo označiti sve kojima je ime i prezime Tomo Senjanin ali i sve kojima je ime Tomislav bez obzira na prezime jer uvjet nije u zagradama

WHERE Prezime = 'Senjanin' AND (Ime= 'Tomo' OR Ime = 'Tomislav')

WHERE BrojRacuna BETWEEN 100 AND 500

WHERE ImaVozacku IS NOT NULL

WHERE Blokada IS NULL

WHERE Boja IN ('Crvena', 'Zelena')

* Koristimo za skupove, alternativno WHERE Boja = 'Crvena' OR Boja = 'Zelena', ovo je bolje kad imamo više pojmova)

WHERE Boja NOT IN ('Crvena', 'Zelena')

* Koristimo za skupove, alternativno WHERE NOT Boja = 'Crvena' AND NOT Boja = 'Zelena', ovo je bolje kad imamo više pojmova)

WHERE Ime LIKE 'A%'

* % Označava nula ili više znakova u nastavku riječi

WHERE Ime LIKE 'F\_lip'

* \_ označava jedan znak

WHERE Prezime NOT LIKE '%ić'

* Opet, označava sva prezimena koja ne završavaju na ić.

### ALIAS / AS

Koristimo kada nekom stupcu hoćemo dodati drugo ime

* Sintaksa

SELECT Ime AS Ime\_Kupca

SELECT Prezime AS KupacPrezime

SELECT IME + '' + Prezime AS PunoIme

* Za spajanje 2 stupca u jedan rezultat

### DISTINCT

Koristimo kada ne želimo ponavljajuće rezultate

* Sintaksa

SELECT DISTINCT Ime FROM Kupac

* Ispisuje samo imena iz tablice Kupac bez ponavljanja

### Sortiranje

Kada želimo sortirati rezultat prema nekom kriteriju.

* Sintaksa

ORDER BY Ime, ASC

* ASC za znakove poreda od A do Z

ORDER BY Ime DESC

ORDER BY Prezime, Ime ASC

* Ako imamo 2 ili više filtera u ORDER BY prvo poreda po prvom, a onda po drugom ukoliko ima više stavki s prvim uvjetom a različitim drugim uvjetima, npr. Ive Ivić, Adam Ivić itd.

### TOP

Kada želimo prvih x rezultata ili prvih x posto rezultata. Najčešće koristimo kad nas zanimaju 3 najveća rezultata (npr. top 3 kupaca itd.)

* Sintaksa

SELECT TOP (3) \* FROM KUPAC

* Vraća prva 3 rezultata iz Kupac (sve stupce jer je \*)

SELECT TOP (3) PERCENT \* FROM Kupac

* Vraća prva 3% rezultata iz Kupac (sve stupce jer je \*)

## JOIN / SPAJANJE TABLICA

Spajamo tablice uvijek preko stranih ključeva jedne tablice na strane ključeve druge tablice.

Postoje:

* INNER JOIN
* LEFT OUTER JOIN
* RIGHT OUTER JOIN
* FULL OUTER JOIN
* CROSS JOIN

Kada spajamo više tablica spaja se rezultat prvog spoja s drugom tablicom, i tako rezultat prvog i drugog spoja s trećom tablicom itd.

Kod spajanja nije bitan poredak (u spoju/lijeva desna). Bitno je da se spajaju primarni i strani ključ.

Najčešće se koristi INNER JOIN jer baca rezultate koji su zajednički 2 tablice. Ako koristimo ostale joinove poput LEFT OUTER JOIN prikazat će se svi rezultati lijeve tablice, dok će desni neki biti NULL.

Postoji i CROSS JOIN koji se koristi za Kartezijev produkt, ali realno skoro nikad.

* Sintaksa

FROM Drzava [FULL OUTER, LEFT OUTER, RIGHT OUTER, INNER, CROSS] JOIN Grad ON Drzava.IDDrzava = Grad.DrzavaID

## EDITING

### Ubacivanje

Možemo unositi nove retke ili kopirati postojeće.

* Sintaksa

INSERT INTO tablica (stupac1, stupac2, …, stupacn) VALUES (vrijednost1, vrijednost2, …, vrijednostn)

INSERT INTO Kupac (Ime, Prezime, Email, Telefon, GradID) VALUES ('Jura', 'Jurić', 'jura@jurino.com', NULL, 3)

INSERT INTO RacunHistory (IDRacun, DatumIzdavanja, BrojRacuna, KupacID, KomercijalistID, KreditnaKarticaID, Komentar, ZapisDatum, ZapisOsoba) SELECT IDRacun, DatumIzdavanja, BrojRacuna, KupacID, KomercijalistID, KreditnaKarticaID, Komentar, GETDATE(), 'Iva Ivić' FROM Racun WHERE DatumIzdavanja BETWEEN '20040601' AND '20040610' ORDER BY BrojRacuna

* Kopiranje, u 99% slučajeva koristimo za kopiranje iz jedne tablice u drugu

### UPDATE

Koristimo za izmjenu podataka.
OBAVEZNO napisati uvjet u WHERE dio inače mijenjamo sve retke.

* Sintaksa

UPDATE tablica SET stupac\_1 = nova\_vrijednost\_1, … stupac\_m = nova\_vrijednost\_m WHERE uvjet

UPDATE Kupac SET Ime='Ed' WHERE Ime='Eduardo' AND Prezime ='Da Silva'

### DELETE

Brišemo cijeli redak.
OBAVEZNO napisati uvjet u WHERE inače brišemo sve.

* Sintaksa

DELETE FROM tablica WHERE uvjet

DELETE FROM Student WHERE Ime='Phillip' AND Prezime='Vunjak'

# Ishod 5

Ishod 5 podrazumijeva funkcije u SQL-u, u principu slične kao u Excelu.

## Matematičke funkcije

Koriste se u bilo kojem dijelu SELECT naredbe.

Vrste:

• ROUND(izraz, duljina) – zaokružuje zadani broj

• FLOOR(izraz), CEILING(izraz) – vraća prvi manji/veći cijeli broj

• ABS(izraz) – vraća apsolutnu vrijednost broja

• SQUARE(izraz) – vraća kvadrat broja

• SQRT(izraz) – vraća drugi korijen broja

• SIGN(izraz) – vraća -1 za negativni broj, 0 za 0 i 1 za pozitivni broj

• RAND() – vraća slučajni broj između 0 i 1

* Sintaksa

SELECT SIGN(IznosRacuna) FROM BankovniRacun

* Vraća je li bankovni račun u minusu ili nije

## Datumske i vremenske funkcije

Koriste se za manipulacija datumskih i vremenskih vrijednosti.

Primjeri:

• GETDATE() – vraća današnji datum i vrijeme

• DATEPART(mjera, datum) – vraća dio datuma i vremena

• year, quarter, month, dayofyear, day, week, weekday, hour, minute, second

• YEAR(datum), MONTH(datum), DAY(datum) – ekvivalentno DATEPART funkciji s prvim parametrom jednakim year, month, odnosno day

• DATEDIFF(mjera, početak, kraj) – vraća koliko je vremena proteklo između zadanih datuma i vremena u zadanoj mjeri (gledaju se granice mjere)

• DATEADD(mjera, količina, datum) – zadanom datumu i vremenu dodaje zadanu količinu mjere

## Stringovske funkcije

Koriste se za manipulaciju stringova.

Vrste:

• CHARINDEX(traženi\_string, string, od) – traži string unutar stringa od zadanog položaja i vraća broj znaka na kojem je string pronađen (ili 0 ako ga nema)

• LEN(string) – vraća broj znakova u stringu

• LOWER(string), UPPER(string) – pretvara string u mala/velika slova

• LTRIM(string), RTRIM(string), TRIM(string) – uklanja razmake

• REVERSE(string) – okreće string

• SUBSTRING(string, od, koliko) – uzima dio stringa počevši od zadanog znaka i uzevši zadano znakova

## Ostale funkcije

Vrste:

• CAST(izraz AS tip\_podataka) – pretvara zadani izraz u zadani tip podataka

• ISDATE(izraz) – vraća je li zadani izraz ispravan datum

• ISNUMERIC(izraz) – vraća je li zadani izraz ispravan broj (cijeli ili decimalni)

• ISNULL(izraz, zamjena) – vraća izraz ako nije NULL; ako jest, vraća zamjenu

## Agregatne funkcije

Spajaju više rezultata u jedno.

Vrste:

• AVG – vraća prosječnu vrijednost

• COUNT – vraća broj elemenata

• MAX – vraća najveću vrijednost

• MIN – vraća najmanju vrijednost

• SUM – vraća zbroj vrijednosti

!Kada koristimo agregatne funkcije u SELECT dijelu, ne možemo vratiti podatke koji nisu agregirani!

* Sintaksa

SELECT MIN(CijenaBezPDV) AS NajnizaCijena, MAX(CijenaBezPDV) AS NajvisaCijena, AVG(CijenaBezPDV) AS ProsjecnaCijena FROM Proizvod WHERE CijenaBezPDV > 0

* Obavezno postaviti uvjet da je veće od nule ako koristimo MIN i očekujemo da ima cijena koje su 0 ili manje.
* MIN ignorira NULL vrijednosti [NULL i 0 nisu isto!] kao i većina agregatnih funkcija

Ako želimo uzeti u obzir rezultate koji su NULL tipa ako je u prodaji došlo do zamjene proizvoda, koristiti funkciju ISNULL(stupac, zamjena[NAJČEŠĆE 0]).

NAPOMENE

• Neke agregatne funkcije rade s više tipova podataka

* Ako AVG radi s int podacima, vratit će int rezultat

• Ako AVG radi s money podacima, vratit će money rezultat

• Za pretvaranje tipova podataka koristimo funkciju CAST():

• CAST(vrijednost AS tip)

• Primjerice: CAST(Broj as float)

• Agregatna funkcija COUNT ima specifično ponašanje:

• COUNT(stupac) će vratiti broj redaka koji nemaju NULL vrijednost upisanu u stupac

• COUNT(\*) će vratiti ukupni broj redaka, bez obzira na NULL vrijednosti

## Grupiranje

Koristi se da odabrane retke dijelimo u grupe (najčešće klase ili vrste)

Grupiraju se stupci naredbom GROUP BY [STUPAC]

Može se dodati i uvjet sa HAVING [UVJET]

* Sintaksa

SELECT Boja, COUNT(\*) AS BrojProizvoda FROM Proizvod WHERE Boja IS NOT NULL GROUP BY Boja



SELECT Boja, COUNT(\*) AS BrojProizvoda FROM Proizvod WHERE Boja IS NOT NULL GROUP BY Boja HAVING COUNT(\*) > 30



# ISHOD 6

## Podupiti

Upit koji se nalazi unutar drugog upita. Uvijek negdje u SELECT naredbi

Može biti unutar

* SELECT [PODUPIT]
* WHERE [PODUPIT]
* HAVING [PODUPIT]

Podupiti se često mogu zamijeniti spajanjem tablica, pa ćete na ispitu vidjeti uputu „BEZ SPAJANJA TABLICA“ u I6.



PODUPIT

### PODJELA PODUPITA

• Podjela prema načinu rada:

1. Konstantni podupiti (ne ovise o retku vanjskog upita)

2. Korelirani podupiti (ovise o retku vanjskog upita)

• Podjela prema mjestu pojavljivanja:

1. Podupiti u SELECT dijelu

2. Podupiti u WHERE/HAVING dijelu

3. Podupiti u FROM dijelu

• Podjela prema povratnoj vrijednosti:

1. Skalarni podupiti (vraćaju jednu vrijednost)

2. Tabularni podupiti (vraćaju tablicu)

### Konstantni podupiti i korelirani podupiti

• Podupit je koreliran ako koristi stupce vanjskog upita

• Ako ih ne koristi, onda je konstantan

• Konstantni podupit se izvršava jedanput

• Konstantan je za sve retke vanjskog upita

• Korelirani podupit se izvršava jedanput za svaki redak vanjskog upita

• Što ako rezultat ima 500.000 redaka? Oprezno s koreliranim podupitima!

• Vanjski upit ne može koristiti stupce podupita, već samo njegov rezultat

• Kao kod spajanja, preporuka je korištenje aliasa za tablice kako bi se izbjegle zabune ili nejasnoće

### Podupiti u SELECT

Mora vratiti jednu vrijednost, može vratiti istu vrijednost za sve retke (KONSTANTNI) ili za svaki redak drugačiju (KORELIRANI).

Može se koristiti za sortiranje u ORDER BY dijelu.

* Konstantni

SELECT Naziv, (SELECT COUNT(\*) FROM Proizvod) AS BrojProizvoda FROM Proizvod

* Korelirani

SELECT TOP 5 Naziv, ( SELECT SUM(s.Kolicina) FROM Stavka AS s WHERE s.ProizvodID =p.IDProizvod ) AS Prodano FROM Proizvod AS p ORDER BY Prodano DESC

### Podupiti u WHERE/HAVING

Podupit u WHERE/HAVING dijelu može vratiti jednu ili više vrijednosti, ovisno o operatoru s kojim se koristi:

• Operatori uspoređivanja =, <, <=, >, >=, !=

• Podupit mora vratiti jednu vrijednost

• Operatori IN i NOT IN

• Podupit mora vratiti tablicu s jednim stupcem

• Operatori EXISTS i NOT EXISTS

• Podupit može vratiti bilo što (za EXISTS: ako podupit vrati 0 redaka, rezultat je false, inače je true; za NOT EXISTS je obratno)

• Podupit može biti konstantni ili korelirani

• Vraćena vrijednost se koristi za filtriranje redaka (WHERE) ili grupa (HAVING)

* Konstantni

SELECT Naziv FROM Proizvod WHERE CijenaBezPDV > ( SELECT AVG(CijenaBezPDV) FROM Proizvod )

* Korelirani s operatorom uspoređivanja

SELECT \* FROM Kupac AS k WHERE ( SELECT COUNT(\*) FROM Racun AS r WHERE r.KupacID =k.IDKupac ) >= 27

* Konstantni s operatorom IN

SELECT \* FROM Proizvod AS p WHERE IDProizvod IN ( SELECT DISTINCT s.ProizvodID FROM Stavka AS s )

* Korelirani s operatorom EXISTS

SELECT \* FROM Proizvod AS p WHERE EXISTS ( SELECT \* FROM Stavka AS s WHERE s.ProizvodID =p.IDProizvod )

# Kako koristiti funkcije u MSMS-u

