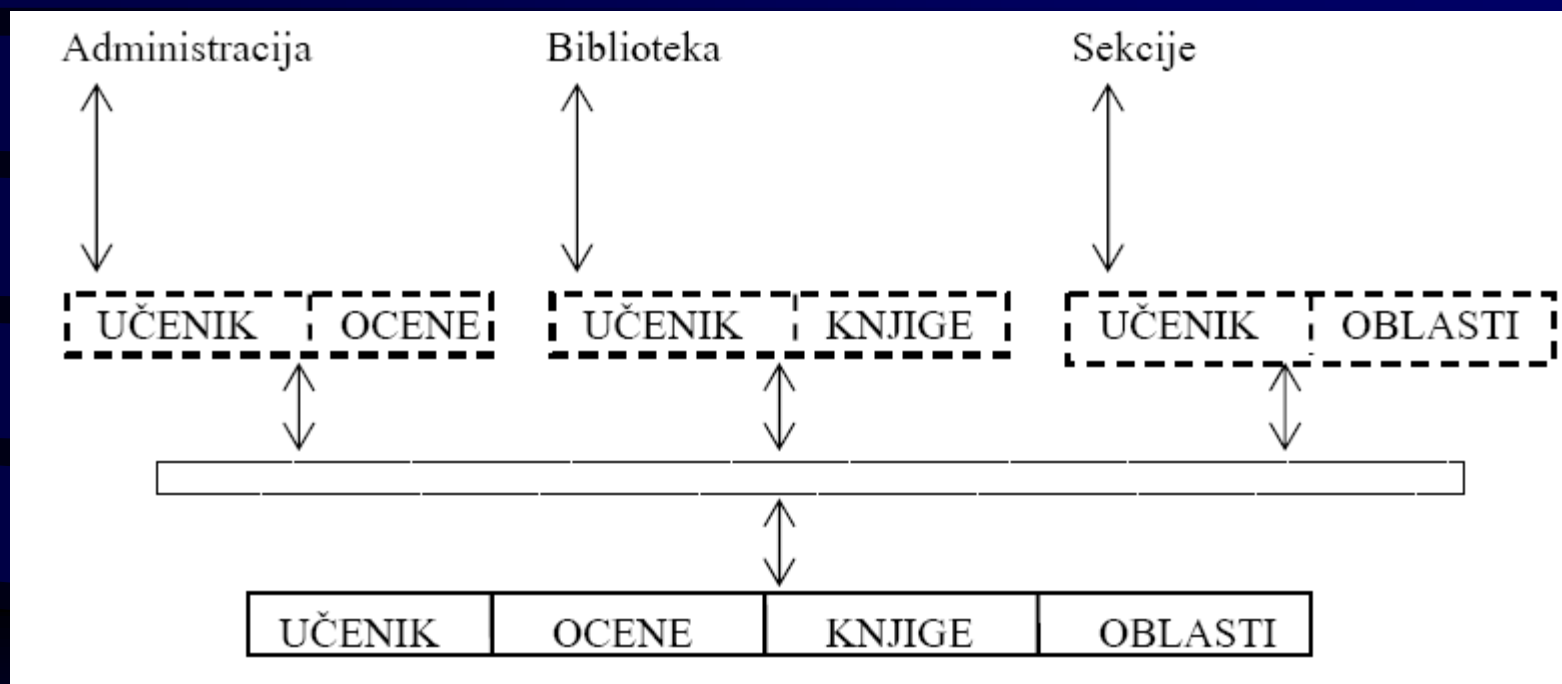


Relacione baze podataka

Sadržaj

- Uvod u baze podataka
- Osnove relacionog modela
- Sistemi za upravljanje bazama podataka
- SQL

Obrada sa bazom podataka



- Integrisanost
- Organizacija prema potrebama korisnika

Sistem koji radi s bazom podataka

Aplikacija se sastoji od obrazaca i izveštaja s kojima radi korisnik aplikacije



Mašina baze podataka nije sastavni deo baze podataka



Baza podataka sadrži fizički oblik šeme i podataka

Šema baze podataka opisuje model podataka koji se čuvaju u bazi

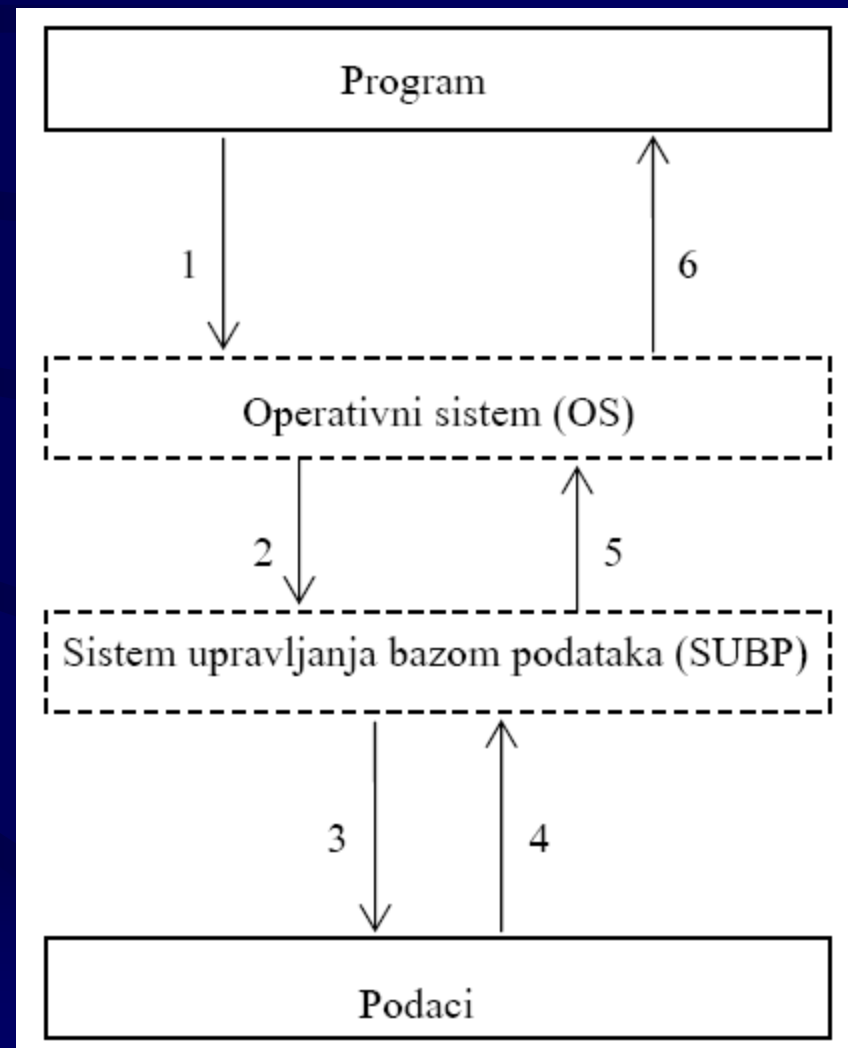
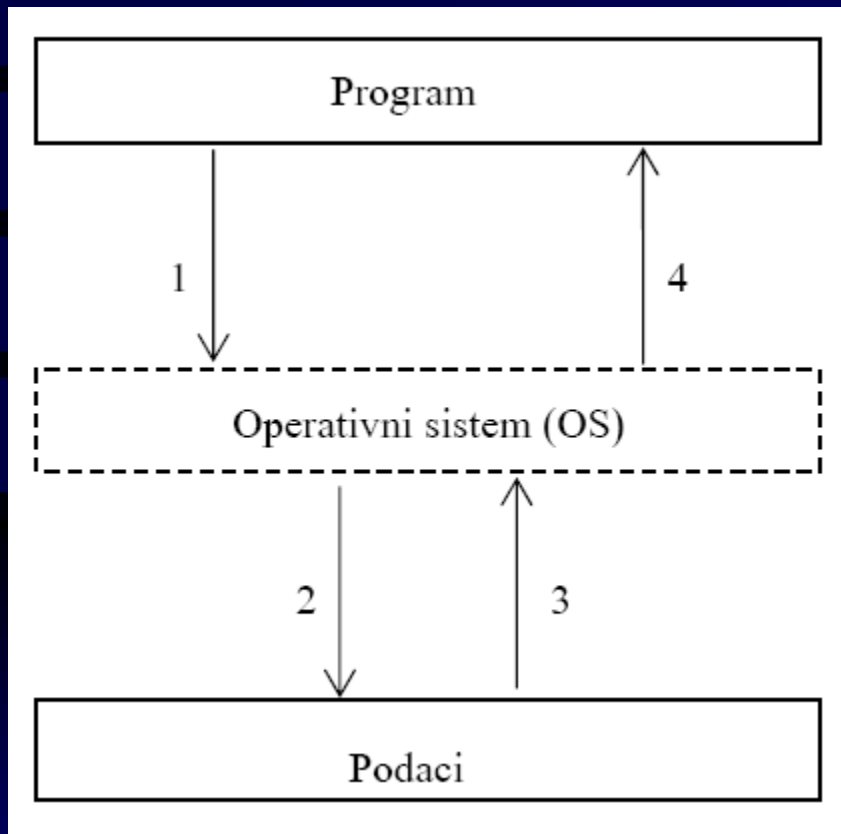
Model podataka je konceptulani opis prostora problema

Prostor problema je neki dobro definisan deo stvarnog sveta

Definicija baze podataka

- Baza podataka je integrisani skup podataka o nekom sistemu organizovan prema potrebama korisnika i elementarni skup postupaka za njihovo održavanje i korišćenje

Nezavisnost podataka i programa



Zadaci baze podataka

- Organizacija prema objektima i odnosima koji postoje u sistemu
- Integrisanost i kontrolisana redundansa
- Organizacija prema potrebama korisnika
- Sigurnost
- Konkurentnost
- Integritet
- Import i eksport podataka
- Performanse
- Ekonomičnost
- Standardizacija

Osnove relacionog modela

Osnovne karakteristike

- Struktura je veoma jednostavna i prihvatljiva korisniku - baza se predstavlja kao skup tabela
- Moguća je formalno-matematička interpretacija modela
 - definisanje formalnog “jezika” baze
 - optimizacija performansi
- Zbog svoje efikasnosti i prilagodljivosti, najpopularnija tehnika rada s bazama podataka

Pravila relacionog modela

- 1) Oblik u kojem se podaci predstavljaju - struktura podataka
- 2) Način na koji se podaci štite - integritet podataka
- 3) Operacije koje se mogu izvršavati nad podacima - manipulisanje podacima

Struktura podataka: osnovni pojmovi

- Cela struktura, u kojoj se podaci konceptualno predstavljaju organizovani u redove i kolone naziva se relacija
- Svaki red podataka zove se n-torka (engl. *tuple*)
- Ukupan broj n-torki u relaciji određuje kardinalnost (engl. *cardinality*) relacije
- Svaka kolona n-torke zove se atribut
- Ukupan broj atributa određuje stepen (engl. *degree*) relacije

Atributi

Atributi			Zaglavlje	
Torke	SupplierName:CompanyName	ProductName:ProductName	UnitPrice:Currency	Telo
	Pavlova, Ltd.	Alice Mutton	\$39.00	
	Plutzer Lebensmittelgroßmärkte AG	Thüringer Rostbratwurst	\$123.79	
	New Orleans Cajun Delights	Chef Anton's Gumbo Mix	\$21.35	
	G'day, Mate	Perth Pasties	\$32.80	
	Formaggi Fortini s.r.l.	Gorgonzola Telino	\$12.50	
	Specialty Biscuits, Ltd.	Sir Rodney's Scones	\$10.00	
	Tokyo Traders	Longlife Tofu	\$10.00	
	New Orleans Cajun Delights	Louisiana Hot Spiced Okra	\$17.00	
	Lyngbysild	Røgede sild	\$9.50	
	Grandma Kelly's Homestead	Northwoods Cranberry Sauce	\$40.00	
	Specialty Biscuits, Ltd.	Scottish Longbreads	\$12.50	
	Formaggi Fortini s.r.l.	Mascarpone Fabioli	\$32.00	
	Nord-Ost-Fisch Handelsgesellschaft mbH	Nord-Ost Matjeshering	\$25.89	
	Karkki Oy	Maxilaku	\$20.00	
	Svensk Sjöföda AB	Gravad lax	\$26.00	
	Exotic Liquids	Aniseed Syrup	\$10.00	
	Formaggi Fortini s.r.l.	Mozzarella di Giovanni	\$34.80	
	Heli Süßwaren GmbH & Co. KG	Gumbär Gummibärchen	\$31.23	

Osobine relacija

- Relacija nije uređena, u njoj su n-torke nagomilane bez ikakvog posebnog redosleda
- Redosled vrsta i kolona nije značajan
- Nije svaka tabela relacija, moraju da se zadovolje sledeći kriterijumi:
 - ne postoje duplikati vrsta tabele
 - nisu dozvoljeni atributi sa ponavljanjem (tabele u tabeli) - Prva normalna forma

Ključevi

- Ključ je atribut ili grupa atributa koji jedinstveno identifikuju jednu n-torku u relaciji
- Ključ zadovoljava
 - osobinu jedinstvenosti
 - osobinu neredudantnosti
- Može postojati više n-torki koje zadovoljavaju zahteve za ključ - kandidati
- U praktičnoj primeni jedna bira se jedna kombinacija - primarni ključ, ostali su alternativni ključevi
- Spoljni ključ: atribut(i) relacije R1 čija se vrednost koristi za povezivanje sa vrednošću primarnog ključa relacije R2

Primer

- Relacija: PROFESOR
- Atributi: JMBG, PREZIME, IME, ...
- Relacija: PREDMET
- Atributi: SIFRA_PRED, NAZIV, JMBG, ...

Još neki pojmovi

- Pogled: relacija izvedena iz skupa baznih i izvedenih relacija, preko operacija nad relacijama
 - U komercijalnim SUBP pogled se izvodi kao tabela koja postoji samo u memoriji
- Nula vrednost: još nepoznata vrednost (NULL)

Ograničenja u relacionom modelu

- 1) Ograničenja vrednosti atributa
- 2) Ograničenja integriteta entiteta
- 3) Ograničenja strukturnog integriteta

Ograničenja vrednosti atributa

- Definicija domena sama po sebi predstavlja ograničenje vrednosti atributa
- Savremeni relacioni SUBP omogućavaju da se ovo elegantno definiše
- Primer: ST_MAG.KOL_MAG ≥ 0
RADNIK.GODINE JE_U (16,65)

Ograničenja integriteta entiteta

- Nijedan atribut koji je PK ili deo PK ne može da uzme NULL vrednost

Ograničenja strukturnog (referencijalnog) integriteta

- Ako relacija R2 poseduje SK koji je povezuje sa relacijom R1 preko njenog PK, tada svaka vrednost SK mora biti jednaka nekoj vrednosti PK ili biti NULL
- Referencijalni integritet obezbeđuje korektno povezivanje objekata u relacionom modelu
- Jedan od najvažnijih koncepata relacionog modela i relacionih SUBP

Sistemi za upravljanje bazama podataka

SUBP, RDBMS

Komercijalni SUBP (RDBMS)

- Komercijalna implementacija principa relacionog modela
- Više kvalitetnih proizvoda na tržištu
 - Microsoft SQL Server
 - Oracle
 - mySQL

Transakcije

- Transakcija: jedno izvršenje logičke jedinice rada nad BP
- Atomski skup aktivnosti, koji se izvršava po principu “sve ili ništa”
- Jedan od ključnih koncepata relacionih SUBP
- Osnovni cilj SUBP je da obezbedi efikasnu obradu transakcija

- Početak transakcije:
BEGIN TRANSACTION
- Završetak transakcije:
COMMIT ili ROLLBACK
- Transakcije ne mogu biti “ugnježdene”
- Transakcija se obično prvo upisuje u log (žurnal BP), a zatim u samu bazu, zbog opravka u slučaju otkaza

Konkurentna obrada transakcija

- Transakcija se ne obavlja izolovano, već uporedo sa drugim transakcijama u sistemu
- Transakcije mogu istovremeno zahtevati isti resurs - isti zapis BP
- Da bi se izbegla nekonzistentnost BP primenjuje se zaključavanje (“lokoti”)

Oporavak baze podataka

- Oporavak BP: vraćanje u korektno stanje posle otkaza sistema
- VRLO VAŽNO U SVAKOM OZBILJNOM RADU !!!
- Vrlo kompleksne tehnike, zasnovane na redudansi podataka
- Veoma značajno: strategija back-upa

Integritet i sigurnost BP

- Tačnost i/ili konzistentnost BP, zaštita od pogrešnih ulaza, grešaka i sistemskih otkaza
- Integritet: zaštita od slučajnog pogrešnog ažuriranja
- Sigurnost: zaštita od neovlašćenog ažuriranja i korišćenja podataka

Pravila integriteta

- Definišu:
 - koje uslove podaci u BP treba da zadovolje
 - kada se vrši provera (pre/posle T)
 - akcije kada uslovi nisu zadovoljeni
- Definišu se za osnovne operacije ažuriranja (Unos, Modifikacija, Brisanje)
- Treba da budu podržana od strane samog SUBP, jezikom visokog nivoa

Pravila integriteta domena

- Definišu vrednosti pojedinih atributa nezavisno od ostalih u bazi
- Mogu se formulisati relativno jednostavnim logičkim izrazima
- Implementacija u MS SQL Serveru:
 - CHECK CONSTRAINT
 - RULES
 - UDDT & RULES
 - TRIGGER

Pravila integriteta relacija

- Definišu uslove pod kojima zapis može da ažurira relaciju i zavisnost n-torki jedne relacije od n-torki druge
- Najznačajnija pravila integriteta
- Akcije:
 - RESTRICTED
 - CASCADES
 - DEFAULTS
 - NULLIFIES

Sigurnost podataka

- Zaštita SUBP od neovlašćenog korišćenja definiše:
 - subjekat zaštite (korisnika, grupu)
 - objektat (tabela, pogled, kolona, procedura, administracija)
 - operacije
- GRANT, REVOKE

SQL

Structured Query Language

Strukturni upitni jezik

Istorijat

- Počeo da se razvija ranih 70-ih u IBM-u, SEQUEL (Structured Query English Language)
- 1986. standardizovan ANSI X3.135, zatim 1989. i ISO
- Prvo korišćen od strane IBM (SQL/DS i DB2) i Oracle, a sada opšte prihvaćen kod svih komercijalnih SUBP

Opis BP (Data definition)

- CREATE TABLE ime_tabele
(ime_kolone tip_podatka [NOT NULL]
[, ime_kolone tip_podatka [NOT NULL]...])
- CREATE TABLE jm
(sifra_jm SMALLINT NOT NULL,
naziv_jm CHAR(30))

- ALTER TABLE ime_tabele
ADD ime_kolone tip_podatka [NOT NULL]
- ALTER TABLE jm
ADD status CHAR(1)
- DROP TABLE ime_tabele

Operacije za izveštavanje - SELECT

- Osnovni oblik SELECT bloka:
SELECT A1, A2,, An
FROM R1, R2,, Rm
WHERE P
- Osnovnom obliku se mogu dodati mnogi detalji, a postoji veći broj dijalekata i “obogaćenja” (**oprez**)

SELECT [ALL | DISTINCT] [TOP nExpr [PERCENT]]

[Alias.] Select_Item [AS Column_Name]

[, [Alias.] Select_Item [AS Column_Name] ...]

FROM [FORCE]

[DatabaseName!]Table [Local_Alias]

[[INNER | LEFT [OUTER] | RIGHT [OUTER] | FULL [OUTER] JOIN

DatabaseName!]Table [Local_Alias]

[ON JoinCondition ...]

[[INTO Destination]

| [TO FILE FileName [ADDITIVE] | TO PRINTER [PROMPT] | TO SCREEN]]

[PREFERENCE PreferenceName]

[NOCONSOLE]

[PLAIN]

[NOWAIT]

[WHERE JoinCondition [AND JoinCondition ...]

[AND | OR FilterCondition [AND | OR FilterCondition ...]]]

[GROUP BY GroupColumn [, GroupColumn ...]]

[HAVING FilterCondition]

[UNION [ALL] SELECTCommand]

[ORDER BY Order_Item [ASC | DESC] [, Order_Item [ASC | DESC] ...]]

Selekcija nad jednom tabelom

- SELECT **br_ind**, **prezime**, **ime** FROM student
- SELECT **naziv** FROM predmet

Eliminisanje duplikata - DISTINCT

- `SELECT DISTINCT` ocena FROM ispitni_spisak
- Eliminisanje duplikata je skupa operacija, treba je izvoditi onda kada je neophodno

Uređivanje skupa rezultata - ORDER

- Redosled zapisa u relaciji nije važan, ali u izveštavanju jeste
- `SELECT * FROM student ORDER BY br_ind`
- `SELECT * FROM student;`
`ORDER BY prezime, ime`

Filtriranje podataka - WHERE

- SELECT * FROM proizvod **WHERE** kolicina > 1000
- SELECT * FROM proizvod ;
 WHERE sifra_jm = 1 **AND** kolicina > 1000
- SELECT * FROM ulaz **WHERE** ;
 BETWEEN (datum_u, {01.01.08}, {31.03.08})
- SELECT * FROM proizvod **WHERE** ;
 UPPER (naziv_jm) **IN** ("GRAM", "KILOGRAM")
- SELECT rad1.ime, rad2.ime, rad1.god_r
 FROM **radnik rad1, radnik rad2**
 WHERE rad1.god_r = rad2.god_r

Pretraga po delimično zadanim vrednostima - LIKE

- SELECT * FROM proizvod WHERE ;
naziv LIKE “%ispravljač%”
- SELECT * FROM radnik WHERE ;
prezime LIKE “%P”
- SELECT * FROM radnik WHERE ;
prezime LIKE “__R%”

Liste polja

- SELECT prezime, ime, adresa, mesto FROM student
- SELECT student.* FROM student
- SELECT proizvod.naziv_p AS artikal, ;
(proizvod.kol1 - proizvod.kol2) AS kolicina ;
FROM proizvod

Ugrađene funkcije

- AVG (atribut)
- SUM (atribut)
- MIN (atribut)
- MAX (atribut)
- COUNT (atribut)
- COUNT (*)
- COUNT (DISTINCT atribut)

- SELECT COUNT (ocena), AVG (ocena) ;
FROM isptni_spisak
- SELECT COUNT (ocena), AVG (ocena);
FROM isptni_spisak WHERE ;
BETWEEN (rok{01.18}, {12.18})
- SELECT COUNT (DISTINCT datum_rodj) FROM
radnik

Grupisanje

- Deli relaciju na grupu n-torki koje između sebe imaju istu vrednost zadatog atributa
- Omogućava dobijanje sumarnih informacija
- ```
SELECT profesor.* FROM profesor ;
GROUP BY sifra_kat;
ORDER BY sifra_kat, prezime, ime
```
- ```
SELECT COUNT (kolicina), SUM (kolicina) ;  
FROM ulaz WHERE ;  
BETWEEN (datum_u, {01.01.98}, {31.03.98}) ;  
GROUP BY sifra_p
```

Uređivanje grupa - HAVING

- Koristi se sa GROUP BY i ima na grupu isti efekat kao WHERE na pojedinačnu n-torku
- `SELECT COUNT (kolicina), SUM (kolicina) ;`
`FROM ulaz WHERE ;`
`GROUP BY sifra_p ;`
`HAVING BETWEEN (datum_u, {01.01.98},`
`{31.03.98}) ;`
`SUM(kolicina) > 1000`

Izdvajanje podataka iz više tabela

- ```
SELECT student.br_ind, student.prezime, ;
student.ime, ispitni_spisak.ocena ;
FROM student, ispitni_spisak ;
WHERE student.jmbg = ispitni_spisak.jmbg
```
- ```
SELECT student.br_ind, student.prezime, ;  
student.ime, ispitni_spisak.ocena, predmet.naziv ;  
FROM student, ispitni_spisak ;  
WHERE student.jmbg = ispitni_spisak.jmbg AND ;  
ispitni_spisak.sifra_pr = predmet.sifra_pr
```

Ugnježdjena SELECT komanda

- SELECT student.* ;
FROM proizvod ;
WHERE student.jmbg =
 (SELECT jmbg FROM spisak_st ;
 WHERE spisak_st.sifra_pr = 403)

Formiranje pogleda

- `CREATE VIEW ime_pogleda`
`[(ime_atributa [, ime_atributa]...)]`
`AS SELECT... FROM... WHERE...`
- `CREATE VIEW stavka_mag AS`
`SELECT st_mag.*, proizvod.naziv_p`
`FROM st_mag, proizvod`
`WHERE st_mag.sifra_p = proizvod.sifra_p`

Operacije za održavanje BP

- INSERT
- UPDATE
- DELETE

Unos podataka u bazu - INSERT

- INSERT INTO ime_tabele
[(atribut [, atribut]...)]
VALUES (konstanta [, konstanta]...)
- INSERT INTO ime_tabele
[(atribut [, atribut]...)]
SELECT... FROM... WHERE ...
- INSERT INTO student (jmbg, prezime, ime)
VALUES ("1701999710445", "Milić", "Ivan")
- INSERT INTO student (jmbg, prezime, ime)
VALUES (m.jmbg, m.prezime, m.ime)

Modifikacija podataka - UPDATE

- UPDATE ime_tabele
SET ime_kolone = izraz
[, ime_kolone = izraz]...
[WHERE ...]
- UPDATE ispitni_spisak
SET ocena = 10
WHERE jmbg = 1701999710445 AND ...
- UPDATE ispitni_spisak
SET ocena = m.ocena

Brisanje podataka - DELETE

- DELETE FROM ime_tabele
[WHERE ...]
- DELETE FROM st_mag
WHERE kol = 0 AND datum_ul < 01.01.97. AND ;
status = 0
- DELETE FROM spisak_st
WHERE jmbg = m.jmbg AND ;
sifra_pr = m.sifra_pr