

1. SQL ("Structured Query Language")

Jezik koji omogućuje:

- pristup bazi podataka
- dohvaćanje podataka iz baze
- dodavanje novih podataka u bazu
- brisanje postojećih podataka iz baze
- izmjenu postojećih podataka u bazi
- definiranje objekata (elemenata) baze podataka
- administraciju korisnika i prava pristupa podacima

Baza podataka (engl. database) - objekt koji sadrži sve ostale objekte.

Ostali objekti:

- tablice (engl. table),
- pogledi (engl. view),
- indeksi (engl. index),
- procedure (engl. procedure),
- okidači (engl. trigger), itd.

Tablice

Relacijska baza podataka koristi tablice za pohranu podataka. Sastoje se od stupaca i redaka (slogovi).

- Stupci sadrže vrijednosti pojedinih atributa promatranog zapisa (atributi entiteta)
- Slogovi sadrže vrijednost svih atributa promatranog zapisa (instance entiteta)

Pogledi (engl. view)

Postoje samo kao definicije upita nad jednom ili više tablica i ne čuvaju podatke u sebi
Omogućavaju:

- prilagodbu logičkog modela podataka specifičnim potrebama korisnika
- provođenje zaštite protiv neovlaštenog pristupa podacima
- Izvršavaju se u trenutku upita

Procedure

- Objekt koji u sebi sadrži programski kod, a aktivira se eksplicitno na zahtjev korisnika

Okidači (engl. trigger)

- objekti koji su čvrsto vezani uz tablicu (nad kojom su definirani)
- sadrže programski kod koji se izvršava implicitno u strogo određenim situacijama

2. SQL tipovi podataka

Znakovni tipovi podataka

CHAR(*n*) - fiksna duljina zapisa, max. 8000 karaktera.

VARCHAR(*n*) - varijabilna duljina zapisa, max. 8000 karaktera.

TEXT - varijabilna duljina zapisa, max. $2^{31}-1$ karaktera.

Numerički tipovi podataka

Cjelobrojni tipovi:

INT - cijeli brojevi od -2^{31} do $2^{31} - 1$

BIGINT - cijeli brojevi od -2^{63} do $2^{63}-1$

SMALLINT - cijeli brojevi od -2^{15} (-32,768) to $2^{15}-1$ (32,767)

TINYINT - cijeli brojevi od 0 to 255

Decimalni brojevi:

DECIMAL(*p,s*) ili **NUMERIC**

- $-10^{38} + 1$ do $10^{38} - 1$
- *p* - ukupan broj znamenki (od 1 do 38)
- *s* - broj decimalnih znamenki ($0 \leq s \leq p$), neobavezan

Unicode tipovi podataka

NCHAR(*n*) -fiksna duljina zapisa, max. 4000 karaktera.

NVARCHAR(*n*) - varijabilna duljina zapisa, max. 4000 karaktera.

NTEXT - max. $2^{30} - 1$ (1,073,741,823) karaktera

Datum i vrijeme

DATETIME - od 01.01.1753 do 31.12.9999, zaokruženo na 0.003 sekunde.

SMALLDATETIME - od 01.01.1900 do 06.06.2079, zaokruženo na minutu.

TIMESTAMP - specijalna namjena, najčešće u milisekundama.

Binarni tipovi podataka

BINARY(*n*) - fiksna duljina binarnog zapisa, max. 4000 bytes.

VARBINARY(*n*) - varijabilna duljina binarnog zapisa, max. 4000 bytes.

IMAGE - max. $2^{31}-1$ (2,147,483,647) bytes.

3. Naredbe SQL jezika

SQL DML - (engl. **D**ata **M**anipulation **L**anguage)

Koriste se za dohvaćanje, pohranu, promjenu i podataka u bazi.

- SELECT - dohvaćanje podataka iz baze
- UPDATE - izmjena postojećih podataka
- DELETE - brisanje postojećih podataka
- INSERT - dodavanje novih podataka

SQL DDL - (engl. **D**ata **D**efinition **L**anguage)

Koristi se za definiciju objekata u bazi: kreiranje i izmjenu strukture objekata.

- CREATE - kreiranje objekata baze
- DROP - uklanjanje objekata baze
- ALTER - izmjena definicije objekata baze
- GRANT - definiranje prava pristupa podacima
- REVOKE - uklanjanje definicije prava pristupa podacima

CREATE TABLE - Naredba za kreiranje tablica

Sintaksa:

```
CREATE TABLE ime_tablice  
("ime_stupca1" tip_podataka_stupca1,  
"ime_stupca2" tip_podataka_stupca2,  
... )
```

Ograničenja (engl. constraints) mogu se navesti kod izrade tablice iza tipa podatka:

PRIMARY KEY - primarni ključ tablice. Može se definirati najviše 1 primarni ključ nad jednom tablicom. SUBP će implicitno kreirati indeks nad jedinstvenim ključem. Prema definiciji svi atributi koji čine primarni ključ su implicitno **NOT NULL**.

UNIQUE - jedinstveni ključ tablice, osigurava da svaki red u stupcu ima različitu vrijednost.

FOREIGN KEY - strani ključ tablice, služi povezivanju podataka između dvije (ili unutar iste) tablice.

NOT NULL - označava da stupac ne može poprimiti NULL vrijednost.

CHECK - osigurava da vrijednosti u stupcu zadovoljavaju navedene uvjete.

DEFAULT - postavlja zadanu vrijednost ukoliko ništa nije uneseno.

Primjer:

```
CREATE TABLE student
(
    ime          CHAR(30),
    prezime     VARCHAR(30) NOT NULL,
    oib         CHAR(11) PRIMARY KEY,
    jmbag       CHAR(13) UNIQUE,
    dat_rod     DATETIME,
    spol        CHAR(1) DEFAULT 'M',
    prosjek     DECIMAL (3, 2),
    slika       IMAGE,
    god_stud    TINYINT
);
```

4. Vrste ograničenja - CONSTRAINTS

- PRIMARY KEY - primarni ključ tablice
- UNIQUE - jedinstveni ključ tablice
- FOREIGN KEY - strani ključ tablice
- CHECK - provjera vrijednosti atributa

4.1. Primarni i jedinstveni (PRIMARY i UNIQUE) ključ

- Prema definiciji svi atributi koji čine primarni ključ su implicitno **NOT NULL**
- SUBP će implicitno kreirati indeks nad primarnim ključem
- Može se definirati **najviše 1** primarni ključ nad jednom tablicom
- SUBP će implicitno kreirati indeks nad jedinstvenim ključem
- U jednoj tablici može se definirati **više** jedinstvenih ključeva
- Atribut koji je jedinstveni ključ može imati i NULL vrijednost

Primjer:

```
CREATE TABLE recepti
(
    oznaka    CHAR(4),
    naziv     VARCHAR(100),
    sastojci  VARCHAR(1000),
    priprema  TEXT,
    slika     IMAGE,
    CONSTRAINT primarni_kljuc PRIMARY KEY(oznaka),
    CONSTRAINT jedinstveni_kljuc UNIQUE(naziv)
);
```

Jednako točno, ali manje detaljno definiranu tablicu, možemo izraditi i na malo drugačiji način (vidi primjer ispod). U ovom slučaju nemamo kontrolu nad nazivima ograničenja (ona se definiraju automatski od strane SUBP), te se primarni ključ može definirati samo nad jednim atributom (a nekad ga je potrebno definirati nad skupom dva ili više atributa).

Primjer:

```
CREATE TABLE recepti
(
    oznaka    CHAR(4) PRIMARY KEY,
    naziv     VARCHAR(100) UNIQUE,
    sastojci  TEXT,
    priprema  TEXT,
    slika     IMAGE
);
```

4.2. Strani (FOREIGN) ključ

Služi povezivanju podataka između dvije (ili unutar iste) tablice.

- Definira se atribut (ili skup atributa) promatrane tablice koji se referenciraju na primarni ključ iste ili neke druge tablice
- Kaskadno ponašanje - automatska izmjena vrijednosti svih stranih ključeva u bazi podataka od strane SUBP-a

Ukoliko želimo definirati način kaskadnog ponašanja za neku tablicu činimo to prilikom definiranja stranog ključa. Kaskadne akcije su moguće u slijedećim situacijama (u referenciranoj tablici):

- ON DELETE
- ON UPDATE

Mogu se kombinirati sa slijedećim opcijama:

- NO ACTION (ako se briše vrijednost u referenciranoj tablici, a postoje strani ključevi koji ju sadrže – brisanje se ne dozvoljava)
- CASCADE (ako se obriše vrijednost u referenciranoj tablici, brišu se slogovi stranih ključeva koji sadrže tu vrijednost)
- SET NULL (ako se obriše vrijednost u referenciranoj tablici, vrijednosti stranih ključeva koji sadrže tu vrijednost se postavljaju na NULL (ako je dozvoljeno))
- SET DEFAULT (ako se obriše vrijednost u referenciranoj tablici, vrijednosti stranih ključeva koji sadrže tu vrijednost se postavljaju na DEFAULT-nu (ako je postavljena i ako postoji u referenciranoj tablici, ako nema DEFAULT-ne → postavlja se NULL vrijednost, ako je dozvoljena))

Primjer:

```
CREATE TABLE racuni
(
    broj INT,
    CONSTRAINT racuni_pk PRIMARY KEY(broj)
...);

CREATE TABLE stavke_racuna
(
    broj_racuna INT,
    CONSTRAINT stavke_fk_racuni FOREIGN KEY(broj_racuna) REFERENC
ES racuni(broj
    ) ON DELETE no action ON UPDATE CASCADE,
...);
```

Primjer:

```
CREATE TABLE grad
(
    pbr    SMALLINT,
    naziv  VARCHAR(50),
    CONSTRAINT grad_pk PRIMARY KEY(pbr)
);

CREATE TABLE stanovnici
(
    jmbg          INT,
    "ime osobe"    VARCHAR(30) NOT NULL,
    "prezime osobe" VARCHAR(30) NOT NULL,
    pbr           SMALLINT,
    adresa        VARCHAR(100) NOT NULL,
    CONSTRAINT stanovnici_pk PRIMARY KEY(jmbg),
    CONSTRAINT stanovnici_fk_grad FOREIGN KEY(pbr) REFERENCES grad(pbr)
);
```

4.3.CHECK

Definira se pravilo za promatrani atribut prema kojemu se provjerava svaka vrijednost koja se upisuje u taj atribut.

Primjer:

```
CREATE TABLE racuni
(
    broj    INT,
    datum  DATETIME NOT NULL,
    kupac   CHAR(100),
    CONSTRAINT racuni_pk PRIMARY KEY(broj)
);

CREATE TABLE stavke_racuna
(
    broj_racuna INT,
    rb          INT,
    kolicina    DECIMAL(5, 2) NOT NULL DEFAULT 1,
    jm          CHAR(3) NOT NULL DEFAULT 'kom',
    naziv_robe  VARCHAR(30) NOT NULL,
    cijena      DECIMAL(6, 2) NOT NULL,
    CONSTRAINT stavke_pk PRIMARY KEY(broj_racuna, rb),
    CONSTRAINT stavke_fk_racuni FOREIGN KEY(broj_racuna) REFERENCES racuni(broj)
    ) ON DELETE CASCADE,
    CONSTRAINT stavke_chk_jm CHECK(jm IN ('kom', 'lit', 'kg')),
    CONSTRAINT stavke_chk_kolicina CHECK(kolicina > 0)
);
```


Primjer:

```
CREATE TABLE djelatnik
(
    mbr      SMALLINT PRIMARY KEY,
    ime      VARCHAR(30) NOT NULL,
    prezime  VARCHAR(30) NOT NULL,
    spol     CHAR(1) DEFAULT 'M',
    placa    DECIMAL(7, 2) DEFAULT 0,
    CONSTRAINT chk_spol CHECK (spol IN ('M', 'Ž')),
    CONSTRAINT chk_placa CHECK (placa >= 0)
);
```

5. Uređivanje tablice

ALTER TABLE - Izmjena strukture postojeće tablice, dodavanje, uklanjanje i izmjena atributa tablice, dodavanje i uklanjanje ograničenja.

Sintaksa:

```
ALTER TABLE table
{ [ ALTER COLUMN column_name
  { new_data_type [ ( precision [ , scale ] ) ]
  [ NULL | NOT NULL ] } ]
| ADD { [ < column_definition >
  | < table_constraint > ] } [ ,...n ]
| DROP { [ CONSTRAINT ] constraint_name
  | COLUMN column } [ ,...n ] }
```

Dodavanje / uklanjanje / izmjena atributa

Primjer - dodavanje atributa:

```
ALTER TABLE racuni
ADD nacin_placanja VARCHAR(30);
```

Primjer - uklanjanje atributa:

```
ALTER TABLE radno_mjesto
DROP COLUMN broj_zaposlenih;
```

Ponekad nije moguće ukloniti svaki stupac (atribut) tablice. Ovo se odnosi na sljedeće situacije:

- Ako je atribut korišten u nekom od indeksa tablice
- Ako je atribut korišten u nekom od pravila ograničenja (CHECK, FOREIGN KEY, UNIQUE, ili PRIMARY KEY)
- Ako je definiran s DEFAULT vrijednošću

Primjer - izmjena postojećih atributa:

```
ALTER TABLE racuni
ALTER COLUMN nacin_placanja CHAR(50);
```

Ovdje treba imati na umu da se prilikom izmjene definicije nekog atributa tablice mora voditi računa o nizu ograničavajućih okolnosti. Naime, nije uvijek moguće izmijeniti definiciju atributa jer bi to za sobom moglo povući i niz drugih izmjena na promatranoj tablici. Slijedi popis najvažnijih ograničenja za atribut koji se želi izmijeniti:

- Ne smije biti nekog od slijedećih tipova: text, image, ntext, ili timestamp
- Ne smije biti korišten kao primarni ili strani ključ

- Prethodni tip atributa mora se moći implicitno izmijeniti u novi tip

Dodavanje / uklanjanje ograničenja

Primjer - dodavanje ograničenja:

```
ALTER TABLE racuni
  ADD CONSTRAINT racuni_chk_placanje CHECK(nacin_placanja IN ('G',
  'K', 'C'));
```

Primjer - uklanjanje ograničenja:

```
ALTER TABLE stavke_racuna
  DROP CONSTRAINT stavke_chk_kolicina;
```

Ne može se uvijek ukloniti svako ograničenje sa tablice što se odnosi ponajprije na primarni ključ koji je referenciran od strane nekog stranog ključa. Npr. nije moguća sljedeća naredba:

```
ALTER TABLE racuni
  DROP CONSTRAINT racuni_pk;
```

ukoliko prije toga nije uklonjeno ograničenje stranog ključa na tablici stavke_racuna:

```
ALTER TABLE stavke_racuna
  DROP CONSTRAINT stavke_fk_racuni;
```

DROP TABLE - Uklanjanje (brisanje) tablice iz baze podataka.

Sintaksa:

```
DROP TABLE table_name
```

Primjer:

```
DROP TABLE recepti;
```

Sintaksa prema SQL99 standardu:

```
DROP TABLE table_name restrict | CASCADE
```

RESTRICT – Ova opcija sprječava izvršavanje naredbe ukoliko postoje pogledi ili druga ograničenja koja referenciraju navedenu tablicu.

CASCADE – Nalaže uklanjanje svih referenciranih objekata zajedno sa navedenom tablicom.

Napomena:

Tablice koje se referenciraju nekim od stranih ključeva ne mogu se ukloniti sve dok se ne ukloni ograničenje stranog ključa.

Tablica kojoj su svi podaci (slogovi, n-torke) obrisani ostaje prisutna u bazi podataka sve dok se eksplicitno ne obriše.

6. Zadaci

Zadatak 1.

Spajanje na MSSQL server:

1. Otvoriti **SQL Server Management Studio** ili **LinqPad**
2. Otvoriti formu za konekciju na server („Connect to server“ ili „Add connection“)
3. Upisati IP adresu servera **koju će Vam dati nastavnik** u polje „Server name“ ili „Server“
4. Označiti **SQL Server Authentication** (ako već nije označeno)
5. Upisati login name: **student** i password: **student**
6. Kliknuti na „Connect“ ili „Ok“

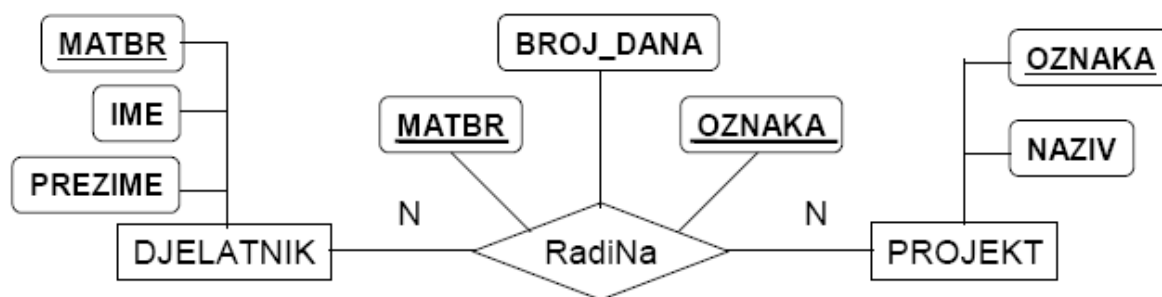
Zadatak 2.

Izraditi tablicu **smartphone** sa sljedećim atributima:

serijski broj uređaja (imei), naziv proizvođača, naziv modela, količina memorije, brzina procesora, veličina ekrana, naziv OS-a, datum zadnje nadogradnje OS-a, ime korisnika i prezime korisnika. Atributima odredite odgovarajuća imena i tip podatka.

Zadatak 3.

Zadan je slijedeći E-R dijagram:



Napravite tablice koje odgovaraju tom E-R modelu.

Zadatak 4.

Izraditi tablicu s grupama studenata za izvođenje vježbi (*grupe*) s odgovarajućim atributima (*id_grupe*, *oznaka_grupe*, *smjer*, *broj_studenata*) te definirati primarni i jedinstveni ključ.

Zadatak 5.

- Izraditi tablicu studenata (*student*) sa sljedećim atributima: *br_indeksa*, *ime*, *prezime*, *grupa*, *godina_upisa*, *godina_studija*, te za istu definirati primarni ključ.
- Također je potrebno izraditi strani ključ nad stupcem (atributom) *grupa* koji se referencira na odgovarajući stupac tablice *grupe* iz prethodnog zadatka.

Zadatak 6.

- Proširiti tablicu *student* iz prethodnog zadatka na način da dodamo još jedan stupac *prosjek* odgovarajućeg tipa.
- Potom je potrebno dodati ograničenje za novi stupac *prosjek* i na taj način onemogućiti upis broja manjeg od 1 u taj stupac.
- Na kraju je potrebno izbrisati stupac *godina_studija* iz iste tablice.

7. Assignments

Assignment 1.

Connecting to the MSSQL:

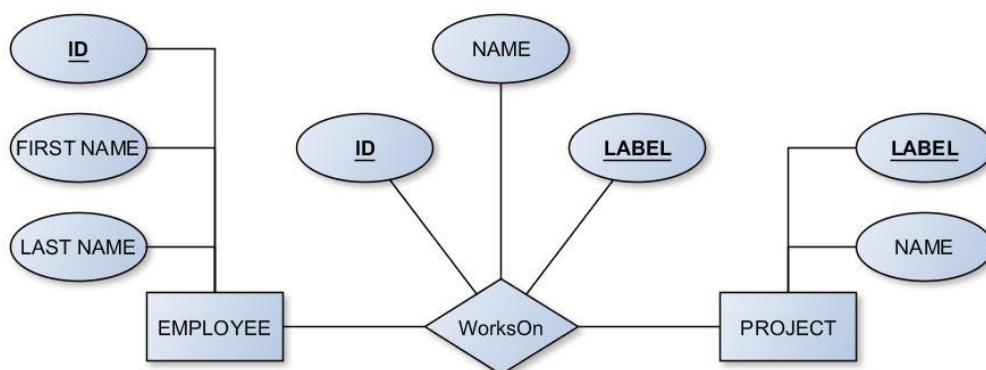
1. Open **SQL Management studio** or **LinqPad**
2. Use connect to server form („Connect to server“ or „Add connection“)
3. In the textboxes „Server name“ or „Server“ use IP address of the server which you will receive from the teacher
4. Check **SQL Server Authentication** (if not checked already)
5. Provide login details (username: **student**, password: **student**)
6. Click „Connect“ or “Ok”

Assignment 2.

Create the SQL table **smartphone** using the following attributes: serial number (IMEI), manufacturer, model, amount of memory, processor speed, screen size, OS type, date of the last update of the OS, user first name, user last name. Give corresponding names and data types to all the attributes.

Assignment 3.

For the given E-R diagram:



create the corresponding SQL tables.

Assignment 4.

Create a table for the student groups for labs (*groups*) with the corresponding attributes (*group_id*, *group_label*, *programme*, *num_of_students*), and primary and foreign keys.

Assignment 5.

- Create table *student* with the following attributes: *index_no*, *first_name*, *last_name*, *group*, *enrollment_year*, *study_year*.
- For the table *student* define a primary key.
- Create a foreign key for the column *group* which references the corresponding column in the table *groups* (from the previous assignment).

Assignment 6.

- Expand the table *student* from the previous assignment by adding the column *average* (having the corresponding datatype).
- Then add the constraint on the newly created column *average* which prevents entering a number less than 1 in it.
- Finally, delete the column *study_year* from the same table.