DROP TABLE student; //brisemo tabelu koja se zove student

CREATE TABLE student( //pravimo tabelu koja se zove student I koja u sebi sadrzi sledece kolone

student\_id INT PRIMARY KEY, //ovu kolonu smo oznacili kao PK, tip je integer

student\_name VARCHAR (20), //tipa varchar I moze da sadrzi 20 karaktera

student\_major VARCHAR (20)

);

CREATE TABLE student(

student\_id INT PRIMARY KEY,

student\_name VARCHAR (20) NOT NULL, // vrednost ne moze da bude podesena na NULL

student\_major VARCHAR (20) UNIQUE //sve vrednosti u koloni moraju da budu razlicite

);

CREATE TABLE student(

student\_id INT AUTO\_INCREMENT, (MySQL AUTO\_INCREMENT) // da sam uvecava sledeci red za 1

student\_name VARCHAR (20),

student\_major VARCHAR (20) DEFAULT 'undecided',

PRIMARY KEY (student\_id) //PK mozemo I ovako da oznacimo

);

INSERT INTO student (student\_name, student\_major) VALUES('Jack', 'Biology'); //ubacujemo podatke u tabelu

INSERT INTO student (student\_name, student\_major) VALUES('Kate', 'Sociology');//redom kako smo definisali

ALTER TABLE student ADD gpa DECIMAL(3, 2);//promeni tabelu student tako sto ces dodati kolonu gpa koja ima

//decimalnu vrednost gde ispred tacke imamo 3 cifre a iza 2

SELECT \* FROM student; //izaberi sve kolone iz tabele student

ALTER TABLE student DROP COLUMN gpa; //promeni tabelu tako sto ces izbrisati kolonu gpa

INSERT INTO student VALUES(1, 'Jack', 'Biology'); // ubaci u tabelu student red sa vrednostima

//student\_id=1; student\_name=’Jack’; student\_major = ‘Biology’

INSERT INTO student VALUES(2, 'Kate', 'Sociology');

INSERT INTO student(student\_id, student\_name) VALUES(3, 'Claire'); //ubaci u tabelu student red gde ces u samo ove dve

//kolone ubaciti ove vrednosti

INSERT INTO student(student\_id, student\_name) VALUES(4, 'Claire');

INSERT INTO student VALUES(5, 'Mike', 'Computer Science');

UPDATE student //promeni tabelu student

SET student\_major = 'Biology' //tako sto ces podesiti kolonu student\_major da ima vrednost ‘Biology’

WHERE student\_id = 4; //svuda gde je vrednost kolone student\_id = 4

UPDATE student

SET student\_major = 'Bio'

WHERE student\_major = 'Biology';

UPDATE student //promeni tabelu student

SET student\_major = 'Biochemistry' //tako sto ces podesiti kolonu student\_major da ima vrednost ‘Bichemistry’

WHERE student\_major = 'Bio' OR student\_major = 'Chemistry'; // svuda gde je vrednost kolone student\_major ‘Bio’ ILI ‘Chemistry’

UPDATE student

SET student\_name = 'Tom', student\_major = 'undecided' // podesi student\_name na ‘Tom’ I student\_major na ‘undecided’

WHERE student\_id = 1; //svuda gde je student\_id = 1

DELETE FROM student // izbrisi iz tabele student

WHERE student\_id = 5; //sve redove gde je student\_id = 5

QUERIES

SELECT student\_name //izaberi kolonu student\_name

FROM student; //iz tabele student

SELECT student\_name, student\_major //izaberi kolonu student\_name I kolonu student\_major

FROM student; //iz tabele student

SELECT student.student\_name, student.student\_major //moze I spred kolone da se stavi ime tabelei tacka – tako je preciznije

FROM student;

SELECT student.student\_name, student.student\_major

FROM student

ORDER BY student\_name; //I poredjaj po rastucem redu kolone student\_name (od A do Z)

SELECT student.student\_name, student.student\_major

FROM student

ORDER BY student\_name DESC; //poredjaj po opadajucem redu (Od Z do A)

SELECT student.student\_name, student.student\_major

FROM student

ORDER BY student\_id DESC;//mozemo da poredjamo I po kriterijumu koji nismo stavili u SELECT

SELECT \*

FROM student

ORDER BY student\_major, student\_id; //moze da poredja I po vise kriterijuma, prvo po major pa onda id

SELECT \*

FROM student

LIMIT 2; //ispisuje samo prva 2 reda

SELECT \*

FROM student

ORDER BY student\_id DESC //prvo ce poredjati od najvecg do najmanjeg id

LIMIT 2; // I onda ispisati prva dva … u ovom slucaju redove sa student\_ide 5 I 4

SELECT \* //izaberi sve kolone

FROM student //iz tabele student

WHERE student\_major = 'Comp Sci'; //gde je kolona student\_major “Comp Sci”

SELECT student\_name, student\_major //izaberi kolone student\_name I student\_major

FROM student //iz tabele student

WHERE student\_major = 'Biochemistry'; //gde je kolona student\_major “Biochemistry”

SELECT student\_name, student\_major

FROM student

WHERE student\_major = 'Biochemistry' OR student\_major = 'Comp Sci'; // gde je kolona student\_major “Biochemistry” ili “Comp Sci”

SELECT student\_name, student\_major

FROM student

WHERE student\_major = 'Biochemistry' OR student\_name = 'Kate'; // gde je kolona student\_major “Biochemistry” ili ime “Kate”

-- //komentar su dve crte

<, >, <=, >=, =, <>, AND, OR //operatori poredjenja…<> znaci not equal to

SELECT student\_name, student\_major

FROM student

WHERE student\_major <> 'Biochemistry'; //gde student\_major nije “Biochemistry”

SELECT \*

FROM student

WHERE student\_id <3;

SELECT \*

FROM student

WHERE student\_id <3 AND student\_name <> 'Kate';

SELECT \*

FROM student

WHERE student\_name IN ('Tom', 'Kate', 'Jack'); // gde su u koloni student\_name vrednosti Tom ili Kate ili Jack

SELECT \*

FROM student

WHERE student\_name IN ('Tom', 'Kate', 'Jack') AND student\_id >2; // gde su u koloni student\_name vrednosti Tom ili Kate ili Jack I gde je student\_id veci od 2

COMPANY DATABASE

CREATE TABLE employee(

emp\_id INT PRIMARY KEY,

first\_name VARCHAR(40),

last\_name VARCHAR(40),

birth\_day DATE,

sex VARCHAR(1),

salary INT,

super\_id INT, //kad napravimo tabelu gde je super\_id PK ovde cemo moci da naznacimo da je ovo FOREIGN KEY

branch\_id INT //kad napravimo tabelu gde je branch\_id PK ovde cemo moci da naznacimo da je ovo FOREIGN KEY);

CREATE TABLE branch(

branch\_id INT PRIMARY KEY,

branch\_name VARCHAR(40),

mgr\_id INT,

mgr\_start\_date DATE,

FOREIGN KEY(mgr\_id) REFERENCES employee(emp\_id) ON DELETE SET NULL

);

// FOREIGN KEY(mgr\_id) REFERENCES employee(emp\_id) ON DELETE SET NULL

//postavi kolonu mgr\_id iz ove tabele koju kreiras (TABLE BRANCH) da bude FOREIGN KEY za kolonu emp\_id tabele employee

//ON DELETE SET NULL se uvek pise za FK…kasnije ce biti objasnjeno

//sada mozemo da dodamo FOREIGN KEY u tabelu employee

ALTER TABLE employee //napravi izmenu u tabeli employee

ADD FOREIGN KEY(branch\_id) //tako sto ces dodati (ADD) strani kljuc(FOREIGN KEY) da bude kolona branch\_id

REFERENCES branch(branch\_id) //tako da u tabeli branch kolona branch\_id je veza (ima iste vrednosti)

ON DELETE SET NULL;

ALTER TABLE employee //napravi izmenu u tabeli employee

ADD FOREIGN KEY(super\_id) //tako sto ces postaviti kolonu super\_id da bude strani kljuc

REFERENCES employee(emp\_id) // koji je isto sto I emp\_id u tabeli employee …u ovom slucaju znaci super\_id je employee id supervizora

ON DELETE SET NULL;

CREATE TABLE client (

client\_id INT PRIMARY KEY,

client\_name VARCHAR(40),

branch\_id INT,

FOREIGN KEY(branch\_id) REFERENCES branch(branch\_id) ON DELETE SET NULL

);

CREATE TABLE works\_with (

emp\_id INT,

client\_id INT,

total\_sales INT,

PRIMARY KEY(emp\_id, client\_id), // kompozitni PK

FOREIGN KEY(emp\_id) REFERENCES employee(emp\_id) ON DELETE CASCADE,

FOREIGN KEY(client\_id) REFERENCES client(client\_id) ON DELETE CASCADE

);

CREATE TABLE branch\_supplier (

branch\_id INT,

supplier\_name VARCHAR(40),

supply\_type VARCHAR(40),

PRIMARY KEY(branch\_id, supplier\_name), //kompozitni PK

FOREIGN KEY(branch\_id) REFERENCES branch(branch\_id) ON DELETE CASCADE

);

-- -----------------------------------------------------------------------------

//ubacujemo podatke u tabelu

-- Corporate

INSERT INTO employee VALUES(100, 'David', 'Wallace', '1967-11-17', 'M', 250000, NULL, NULL); //ovde za branch\_id

INSERT INTO branch VALUES(1, 'Corporate', 100, '2006-02-09');