Poizvedovanje z jezikom SQL

Izbiranje. Stavek SELECT v jeziku SQL DML.

```SQL
SELECT ...
```

Vstavljanje vrstic. Stavek INSERT v jeziku SQL DML.

```SQL
INSERT INTO t [(A, B)] VALUES (1, "foo"), (2, "bar"), (3, "baz"), (4, NULL);
```

Brisanje vrstic. Stavek DELETE v jeziku SQL DML.

```SQL
DELETE FROM t [WHERE A = 1];
```

Posodabljanje vrstic. Stavek UPDATE v jeziku SQL DML.

```SQL
UPDATE t SET A = A - 1 [WHERE A != 1];
```

Ustvarjanje tabele. Stavek CREATE TABLE v jeziku SQL DDL.

```SQL
CREATE TABLE t (A INT NOT NULL, B VARCHAR(32) [NULL]);
```

Tipi stolpcev tabele so lahko logične vrednosti, cela števila, realna števila, nizi znakov, datumi, časovne značke itd.

```SQL
BOOL[EAN], INT[EGER], FLOAT, [VAR]CHAR, DATE[TIME], TIMESTAMP ...
```

Spreminjanje tabele. Stavek ALTER TABLE v jeziku SQL DDL.

```SQL
ALTER TABLE t ADD PRIMARY KEY (A);
```

Brisanje tabele. Stavek DROP TABLE v jeziku SQL DDL.

```SQL
DROP TABLE t;
```

Ustvarjanje pogleda. Stavek CREATE VIEW v jeziku SQL DDL.

```SQL
CREATE VIEW subt AS SELECT * FROM t WHERE B IS NOT NULL;
```

Ustvarjanje indeksa. Stavek CREATE INDEX v jeziku SQL DDL.

```SQL
CREATE INDEX ind ON t (B); -- preskoči
```

Ustvarjanje baze. Stavek CREATE DATABASE v jeziku SQL DDL.

```SQL
CREATE DATABASE toy [DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci];
```

Domena. *Toy* spodaj.

$$\mathrm{t}\,(\underline{\mathrm{A}}, \mathrm{B})$$

1. naloga. *(Toy)* V smiselnem zaporedju izvedite vse zgornje SQL stavke.

```SQL
[DROP DATABASE IF EXISTS toy;]
CREATE DATABASE toy [DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci];
[USE toy;]
CREATE TABLE t (A INT NOT NULL, B VARCHAR(8) [NULL]);
ALTER TABLE t ADD PRIMARY KEY (A);
CREATE VIEW subt AS SELECT * FROM t WHERE B IS NOT NULL;
INSERT INTO t [(A, B)] VALUES (1, "foo"), (2, "bar"), (3, "baz"), (4, NULL);
DELETE FROM t WHERE A = 1;
UPDATE t SET A = A - 1;
DROP TABLE t;
[DROP DATABASE toy;]
```

Domena. *Study* spodaj.

$$\begin{gathered}\mathrm{person}\,(\underline{\mathrm{PID}}, \mathrm{Name}, \mathrm{Surname}\,\ldots)\\ \mathrm{course}\,(\underline{\mathrm{CID}}, \mathrm{Title}\,\ldots\,\#\mathrm{PID})\\ \mathrm{assistant}\,(\#\underline{\mathrm{PID}}, \#\underline{\mathrm{CID}}\,\ldots)\\ \mathrm{study}\,(\#\underline{\mathrm{PID}}, \#\underline{\mathrm{CID}}, \underline{\mathrm{Year}}\,\ldots)\end{gathered}$$

2. naloga. *(Study)* Z uporabo jezika SQL ustvarite domeno *Study* in jo napolnite s spodnjimi podatki.

```SQL
CREATE DATABASE study;
[USE study;]
```

PID	Name	Surname	…
1	Marko	Bajec	…
2	Lovro	Šubelj	…
3	Uroš	Paščinski	…
4	⟨*moje ime*⟩	⟨*moj priimek*⟩	…

```SQL
CREATE TABLE person (
    PID INT NOT NULL AUTO_INCREMENT,
    Name VARCHAR(16) NOT NULL,
    Surname VARCHAR(32),
    PRIMARY KEY (PID));
INSERT INTO person (Name, Surname) VALUES
    ("Marko", "Bajec"), ("Lovro", "Šubelj"),
    ("Uroš", "Paščinski"), ("<moje ime>", "<moj priimek>");
```

CID	Title	…	PID
1	Osnovne podatkovnih baz	…	1
2	⟨*naziv predmeta*⟩	…	?

```SQL
CREATE TABLE course (
    CID INT PRIMARY KEY,
    Title VARCHAR(32),
    PID INT NOT NULL,
    FOREIGN KEY (PID) REFERENCES person (PID));
INSERT INTO course VALUES
    (1, "Osnovne podatkovnih baz", 1), (2, "<naziv predmeta>", 2);
```

PID	CID	…
2	1	…
3	1	…

```SQL
CREATE TABLE assistant (
    PID INT,
    CID INT,
    PRIMARY KEY (PID, CID),
    FOREIGN KEY (PID) REFERENCES person (PID),
    FOREIGN KEY (CID) REFERENCES course (CID));
INSERT INTO assistant VALUES  (2, 1), (3, 1);
```

PID	CID	Year	…
4	1	2020/21	…

```SQL
CREATE TABLE study (
    PID INT,
    CID INT,
    Year DATE DEFAULT '2000-01-01',
    PRIMARY KEY (PID, CID, Year),
    FOREIGN KEY (PID) REFERENCES person (PID),
    FOREIGN KEY (CID) REFERENCES course (CID));
INSERT INTO study VALUES (4, 1, '2020-10-01');
```

Ustvarjanje uporabnika. Stavek CREATE USER v jeziku SQL DCL.

```SQL
CREATE USER student@'%' -- 'localhost', '<moj ip>'
    IDENTIFIED BY "4"; -- naključno geslo (met poštene kocke)
```

Nadzor pravic. Stavka GRANT in REVOKE v jeziku SQL DCL.

```SQL
GRANT ALL PRIVILEGES ON study.* TO student@"%"; -- branje, pisanje, ustvarjanje
GRANT SELECT ON toy.t TO student@'%'; -- le branje tabele t v toy
[FLUSH PRIVILEGES;]
```

Uveljavljanje sprememb. Stavka COMMIT in ROLLBACK v jeziku SQL TPO.

```SQL
...
```