

სემინარი 11. MySQL მონაცემთა ბაზა

კურსის განმავლობაში გამოვიყენებთ MySQL რელაციურ მონაცემთა ბაზას..

ინსტალაციისთვის ოფიციალური საიტიდან გადმოიწერეთ ბაზის open-source, არაკომერციული ვერსია - MySQL Community Server:

<http://dev.mysql.com/downloads/mysql/>

MS Windows-ის შემთხვევაში შეგიძლიათ გადმოწეროთ ბაზის ინსტალერი გრაფიკული ინტერფეისით და მას მიჰყვეთ.

სემინარის ფარგლებში მხოლოდ 'სერვერის' ინსტალაცია საკმარისია, სურვილის შემთხვევაში შეგიძლით მიუთითოთ დამატებითი აპლიკაციები, მაგალითად MySQL Workbench რომელიც საშუალებას მოგცემთ გრაფიკულ ინტერფეისით იმუშაოთ მონაცემთა ბაზასთან. წინააღმდეგ შემთხვევაში ცხრილების შექმნა და მართვა მოგიწევთ command-line კლიენტით.

სერვერის ინსტალაციის პროცესში შეგხვდებათ ერთი ველი: Start the MySQL Server at System Startup

თუ არ გჭირდებათ რომ ბაზის სერვერი ოპერაციულთან ერთად ჩაირთოს ხოლმე, ეს პარამეტრი გაუთიშეთ.

ინსტალაციისას ავტომატურად შეიქმნება ბაზის root მომხმარებელი, რომლისთვისაც პაროლს გკითხავთ ინსტალატორი.

სერვერის ინსტალაციის შემდეგ თქვენს ხელთ ასევე იქნება პატარა command-line კლიენტი, რითიც შეძლებთ ბაზასთან ურთიერთობას: mysql.exe

სერვერთან დაკავშირება / ავტორიზაცია:

```
mysql --user=root -p [password]
```

მონაცემთა ბაზის შექმნა:

```
CREATE DATABASE university;
```

ბაზის არჩევა:

```
USE university;
```

ცხრილის შექმნა:

```
CREATE TABLE students (  
    id INT NOT NULL AUTO_INCREMENT,  
    firstname VARCHAR(50),  
    lastname VARCHAR(50),  
    register_date DATE,
```

```
PRIMARY KEY(id)
);
```

ცხრილის ყველა ჩანაწერების წამოღება:

```
SELECT * FROM students;
```

ცხრილში ჩანაწერის ჩამატება:

```
INSERT INTO students (firstname, lastname, register_date)
VALUES ('John', 'Doe', now());
```

ჩანაწერების გაფილტვრა:

```
SELECT * FROM students WHERE id = 1;
```

ცხრილში ჩანაწერის განახლება

```
UPDATE students SET firstname = 'Johnathan' WHERE firstname = 'John';
```

ცხრილიდან ჩანაწერის წაშლა

```
DELETE FROM students WHERE id = 1;
```

შექმენით მეორე ცხრილი cards, სადაც შეინახება სტუდენტების დაშვების ბარათების ინფორმაცია:

```
CREATE TABLE cards (
    id INT NOT NULL AUTO_INCREMENT,
    code VARCHAR(50),
    status INT DEFAULT 1,
    PRIMARY KEY(id)
);
```

სტუდენტების ცხრილი გადააკეთეთ და მიაბით ბარათის ინფორმაცია:

```
ALTER TABLE students ADD COLUMN card_id INT NOT NULL DEFAULT 0;
```

სტუდენტების ცხრილში არსებულ card_id ველს დაადეთ foreign key შეზღუდვა:

```
ALTER TABLE students
ADD CONSTRAINT fk_card_id
FOREIGN KEY (card_id) references cards(id);
```

ორი ცხრილიდან ინფორმაციის წამოღება:

```
SELECT students.*, cards.code  
FROM students  
JOIN cards  
ON students.card_id = cards.id;
```

გასამეორებელი ფუნქციები, ბრძანებები და keyword-ები:

```
where, and / or, like, inner / left / right join, order by, limit,  
drop, sum(), count()
```