****

Latvijas Universitātes Vadības un uzņēmējdarbības mācību centrs

(LU VUMC)

**IEVADS DATU BĀZĒS**

ROBERTS POLIS  
VJAČESLAVS PĒTERIS VASIĻEVSKIS

2021

ESF projekts Nr. 8.4.1.0/16/l/001  
‘’Nodarbināto personu profesionālās kompetences pilnveide”



# Lesson 02

## Uzdevums 1 : pieslēgšanās DB serverim un pirmais SQL kods

### Pieslēgšanās MySQL DB serverim ar komandrindu

Šajā uzdevumā pieslēgšanās un darbs notiks OS komandrindā. Mūsdienās ir vairākas klientu programmas ar ērtu grafisku interfeisu, kā DB servera specifiski (piemēram MySQL Workbench priekš MySQL DB, PL/SQL Developer priekš Oracle DB), tā arī universāli (piemēram DBeaver). Datubāžu administrātori arī mūsdienās bieži izmanto komandrindas klientus to vienkāršības un skriptu palaišanas ērtuma dēļ. Uzsāksim pirmā SQL koda veidošanu komandrindas klientā, lai gūtu priekšstātu par darbu ar to. Turpmāk izmantosim MySQL Workbench.

* Palaist OS komandrindu. Pieslēgties MySQL DB serverim ar šādu komandu (ielikt sev piešķirto N, X, PWD vērtības) :

mysql -h gNstudentX.ckh4seomnuyp.eu-west-1.rds.amazonaws.com -P 3306 -u admin -p PWD

* Apskatīt pieejamas datubāzes :

show databases;

* Izveidot datubāzi :

create database my\_db\_01;

* Pieslēgties jaunizveidotajai datubāzei :
* use my\_db\_01;
* -- Tā var pārbaudīt datubāzi, kurai esat pieslēdzies

select database();

* Apskatīt pieejamas tabulas :

show tables;

* Izveidot tabulu :

create table my\_tab\_01(a\_name varchar(200), a\_content varchar(4000));

* Apskatīt tabulas tagad. Izveidot jaunu ierakstu ar INSERT komandu, un apskatīt to ar SELECT komandu :
* insert into my\_tab\_01(a\_name, a\_content) values('Test', 'Hello World !');

select \* from my\_tab\_01;

* Izdzēst datubāzi. Atslēgties no MySQL servera.
* drop database my\_db\_01;
* -- Tā var redzēt ka vairs nav pieslēguma datubāzei
* select database();
* -- Ejam ārā no klienta

quit;

### Saglabāt kodu GIT sistēmā

* Izkopēt kodu no komandrindas lokalā teksta failā
* Saglabāt failu GIT repozitorijā savā direktorijā. Izmantosim vienkāršotu pieeju, kad notiek tieša saglabāšana master zarā. Programmatūras izstrādes projektos šāda pieeja tiek izmantota reti, parasti mazos projektos kuros ir iesaistīts viens vai nedaudzi programmētāji. Krietni biežāk tiek izmatota zarošanas pieeja (branching), kad katram uzdevumam tiek veidots master zara kopijas zars, kurā notiek izstrāde, kuras laikā kopijas zarā var tikt veikti desmitiem un simtiem komītu ; pēc darba pabeigšanas pie konkrēta uzdevuma notiek koda apvienošana ar master zaru (merge operation), kura tiek veikta caur pull request veidošanu, kas parasti paredz arī koda kvalitātes kontroli.
  + Aiziet lokāli uz direktoriju kur ir noklonēts repozitorijs
  + Meklēt savas grupas direktorijā, atbilstoši vārdam un uzvārdam savu direktoriju
  + Iekopēt tur teksta failu
  + Aiziet uz GIT klientu. Tur jāredz, ka parādījās izmaiņas. Jāizdara šādas darbības :
    - Stage all : atzīmēs visus jaunus / mainītus failus kā gatavus saglabāšanai
    - Commit : izdarīs lokālu saglabāšanu
    - Push : iesūtīs izmaiņas serverī, mūsu gadījumā GitHub-ā