****

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”



# Nodarbība 08

## Uzdevums 1 : Funkcijas

### Funkciju izmantošana SQL kodā

Funkcijas spēlē ļoti svarīgu lomu jebkurā programmēšanas valodā. Funkcija apstrādā datus un atgriež apstrādes rezultātu. Ir daudz gatavu funkciju, bet reālas izstrādes laikā diezgan bieži nākas veidot specifiskas funkcijas.

* Pieslēdzieties datu bāzei Sakila ar MySQL Workbench.
* Uzrakstiet dažus vienkāršus vaicājumus :
  + visas valstis ar nosaukumu garāku par 10 burtiem.
  + visas pilsētas, kuru nosaukumā ir teksts 'ah'.

Pievērsiet uzmanību tam, ka visi rezultāti ir neatkarīgi no lieliem / maziem burtiem (case-insensitive). Lai atlasītu datus atšķirot lielus/mazus burtus (case-sensitive), pievienojiet collation specifikāciju vaicājumā -- uzreiz pēc lauka nosaukuma atlases kriterijos pievienojiet collation specifikāciju utf8\_bin, lai sanāktu : ... city collate utf8\_bin ... .

* + Atlasiet visus ierakstus no tabulas actor. Iekļaujiet rezultāta atlasē arī pilnu vārdu -- izmantojiet funkciju CONCAT.
  + Uzrakstiet citus vaicājumus izmantojot dažādas lekcijā izskatītas teksta, laika un matemātiskas funkcijas. Pamēģiniet kombinēt funkcijas, saucot funkciju no funkcijas.
* Pamainiet kādu vienu ierakstu pēc izvēles valstu tabulā, izmantojot UPDATE komandu. Jāmaina lauka last\_update vērtība, piedevām tā jāmaina uz šo konkrētu vērtību : 1919.gada 11.novembris. Izmantojiet STR\_TO\_DATE funkciju.

### Analītiskas funkcijas

Analītiskas funkcijas strādā ar datu kolekcijām. Tām ir vienkārša sintakse vienkāršajā variantā, kā, piemēram, max() funkcija; un tās arī satur daudz iespēju, proti, tajās var norādīt izpildei daudz sarežģītākas lietas, izmantojot papildu sintaksi. Tās tiek plaši pielietotas gan OLTP sistēmās (OLTP = On-Line Transaction Processing, jeb datu ievades datu bāzes sistēma), gan jo sevišķi DWH sistēmās (DWH = Data WareHouse, jeb datu noliktava).

* Turpiniet darbu ar Sakila datu bāzi. Uzrakstiet vienkāršu vaicājumu, lai noskaidrotu maksimālu maksājuma summu tabulā payment.
* Tagad pievienojiet vaicājumam citu lauku, customer\_id, bez jebkādas funkcijas. Proti, maksājuma summa joprojām paliek funkcijā max, bet otrs lauks ir pats par sevi. MySQL šajā gadījumā atgriež vienu ierakstu, iekļaujot tajā, es pieļauju, pirmo pieejamo vērtību no tabulas. Citas DB šādā gadījumā izmetīs kļūdu, jo tā rakstīt nav korekti : vai nu jāizmanto visiem laukiem funkcija, kas atgriezis vienu vērtību no datu seta (min, max, etc), vai nu jāpievieno GROUP BY tiem laukiem, kuri būs vaicājumā bez funkcijas. Pievienojiet nosacījumu GROUP BY customer\_id. Ieraudzīsiet, ka tagad saņemsiet visus klientu identifikatorus, un katram maksimālo maksājuma summu.
* Tagad pamainiet esošo vaicājumu, atstājot tajā tikai klientus ar maksimālo maksājumu summu vismaz 10 vai lielāku. Izmantojiet operandu HAVING uzreiz pēc GROUP BY.

Saglabājiet savu kodu failā. Failu saglabājiet GIT sistēmā.