****

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 10

## Uzdevums 1 : Tables Joins : Vaicājumu no vairākām savienotām tabulām

### Tables Joins

Tabulu savienošana vaicājumos ir absolūtu bāzes lieta, bez kuras no relāciju datu bāzēm nebūtu jēgas. Iepriekšējās nodarbībās mēs jau izskatījām relāciju datu bāžu konceptu un datu normalizāciju, kā arī indeksēšanu un arī mazliet mēģinājām tabulu savienošanu. No pieredzes varu teikt, ka absolūts vairākums vaicājumu reālā RDBMS sistēmā ir no vairākām tabulām.

Izmēģināsim interesantākus vaicājumus ar reāliem datiem.

* Pieslēdzieties datu bāzei sakila ar MySQL Workbench.
* Sākumam izpildiet vienkāršu vaicājumu, atlasot visus datus no tabulas film. Labs stils ir izmantot tabulas aliasu pilnvērtīgi, proti, dot aliasu FROM daļā, un izmantot to arī visās pārējās daļās, gan SELECT, gan WHERE, utt.
* Tagad papildiniet savu vaicājumu ar JOIN daļu, pievienojot tabulas film\_category un category.
  + Lai to izdarītu, no sākuma atlastiet datus no katras tabulas atsevišķi, lai labāk saprastu ko Jūs pievienosiet. Varat arī apskatīt tabulu uzbūvi objektu navigatorā. Vēl labākai sapratnei varat uztaisīt Database => Reverse Engineer, uzģenerējot Entity-Relationship diagrammu.
  + Iesaku uzreiz pierast pie ANSI tabulu savienošanas sintakses. Tas paredz tabulu savienošanas nosacījumus rakstīt FROM daļā, ar (RIGHT / LEFT) JOIN atslēgvārdu palīdzību.
    - Tas ir SQL valodas standarts, kurš jāatbalsta visām relāciju datu bāzēm.
    - Tas efektīvi atdala JOIN nosacījumus no WHERE daļas un padara kodu lasāmāku.
    - Iesaku neizmantot vārdus INNER un OUTER, jo tie pēc būtības ir lieki un tikai padara kodu garāku. Vienkārši JOIN in vienmēr INNER, bet abi LEFT un RIGHT -- ir OUTER. Līdz ar to rakstiet JOIN, LEFT JOIN, RIGHT JOIN.
    - Diemžēl MySQL neatbalsta diezgan reti vajadzīgu FULL JOIN. Citas datu bāzes to atbalsta, kā Oracle, PostgreSQL utt. Lai vai kā, šis ir vēl viens iemesls lai izmantotu ANSI standartu, savienojot tabulas. Piemēram, Oracle atbalsta savu savienošanas notāciju, kad tabulu JOINs ir WHERE daļā, un OUTER JOINs tiek realizēti, liekot klāt pie atslēgas plusiņu iekavās : t1.t2\_id(+) = t2.id. Ir eleganti un īsi, bet šāds vaicājums feilos jebkurā citā datubāzē, kā arī Oracle neparedz plusiņu pielikšanu abos galos (FULL JOIN), bet ANSI standartu atbalsta pilnībā, arī FULL JOIN.
* Pievienojiet vienu ierakstu film tabulai ar INSERT komandu. Obligāti aizpildāmi lauki ir title un language\_id. Primary key lauks film\_id tiks aizpildīts automātiski, tas ir autoincrement lauks, kas ir izplatīta un pareiza prakse relāciju datu bāzēs -- veidot unikālas atslēgas vērtības no sekvencēm. Pārbaudiet ar vaicājumu no šīs tabulas, sakārtojot apgrieztā secībā pēc primārās atslēgas (ORDER BY ... DESC). Ieraudzīsit ka jūsu ieraksts ir pievienots.
* Izpildiet vaicājumu ar JOINiem, sakārtojot datus tāpat kā iepriekšējā vaicājumā. Redzēsit ka jūsu ieraksta nav rezultātos, kam par iemeslu filmas-kategorijas saites ieraksta neesamība starpniektabulā film\_category. Citiem vārdiem, jūsu filmai nav norādīta kategorija. Iespējams, tā jānorāda, bet iespējams Jūs esat kinematogrāfa gēnijs, kurš tikk radīja jaunu kino žanru :) .
* Nomainiet JOINus tā, lai dabūtu rezultātos arī Jūsu jaunpievienoto filmu.

Saglabājiet savu kodu failā. Failu saglabājiet GIT sistēmā.