****

Latvijas Universitātes Vadības un uzņēmējdarbības mācību centrs

(LU VUMC)

**IEVADS DATU BĀZĒS**

2022

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



# Nodarbība 11

## Terminu vārdnīca

SQL : Structured Query Language / Strukturētu Vaicājumu Valoda. Tā ir lietošanas apgabalam specifiska programmēšanas valoda. Tā ir projektēta un veidota datu pārvaldīšanai iekš RDBMS. Tā ir sevišķi noderīga strukturētu datu apstrādē, proti, tādu datu, kuri ietver sevī vienības, vērtības un saites (relācijas) starp tām.

Relational Database (Relāciju datubāze) : Digitāla datubāze, balstīta uz relāciju datu modeļa. Relāciju datu modelis ir tāds, kurā dati ir organizēti vienībās, kuras ir savstarpēji saistītas; vārdam relācija tulkojums latviešu valodā varētu būt saite. Relāciju datubāzē vienības ir tabulas, kuras savukārt sastāv no mazākām vienībām -- ierakstiem, kuri sastāv no laukiem (vēl saukti par atribūtiem). Tabulām ir unikālu vērtību lauki, kuri tiek saukti par atslēgām; ar to palīdzību tabulas tiek saistītas savā starpā, proti, ar atslēgu palīdzību tiek uzturētas relācijas.

Database table : Datubāzes tabula. Tā ir saistītu datu kolekcija, kura ir organizēta vertikālu kolonu, jeb lauku un horizontālu ierakstu struktūrā. Lauki tiek identificēti pēc nosaukuma, kuriem jābūt unikāliem tabulas robežās. Tabulā ir ierobežots definēts kolonu skaits, katrai kolonai ir ne tikai savs nosaukums, bet arī datu tips un / vai citi ierobežojumi; ierakstu skaits ir tiešā veidā neierobežots un ir atkarīgs tikai no datubāzes servera konfigurācijas, pieejamas diska vietas u.tml.

Subquery : Apakšvaicājums. Tas ir vaicājums, kas ir iekļauts citā vaicājumā. Apakšvaicājums var būt iekļauts cita vaicājuma dažādās daļās, piemēram SELECT daļā, FROM daļā, WHERE daļā. Apakšvaicājums var tikt saistīts ar iekļaujoša vaicājuma laukiem, bet, atkarībā no daļas kurā tas ir iekļauts var atšķirties piekļuve galvenā vaicājuma laukiem.

Alias: SQL valodā lietots termins, kas nozīmē kāda objekta vai atribūta alternatīvu lokāli izmantojamu nosaukumu. Tas tiek plaši izmantots ērtībai, koda apjoma samazināšanai un lasāmībai. Piemērs : SELECT ptrx.payment\_date pmt\_date FROM payment\_transaction ptrx; : gari tabulas un lauka nosaukumi ir vajadzīgi datubāzes modelēšanas posmā, lai datu modelis būtu kvalitatīvs un atribūtu jēga nolasāma no nosaukumiem ; savukārt kodā šie nosaukumi ir pārāk gari un neērti, un aliasu izmantošana atrisina šo problēmu.

Cartesian join : Tabulu savienošanas veids bez nosacījumiem. Vēl ir pazīstams kā dekarta reizinājums. Tiek atlasītas visas iespējamas divu tabulu ierakstu kombinācijas; piemēram, ja abās tabulās ir 10 ieraksti, tiks atlasīti 100 ieraksti, jo tiek atlasīts katrs ieraksts no vienas tabulas kombinācija ar visiem ierakstiem no otras tabulas. Nav ieteikts izmantošanai, jo var saražot milzīgu daudzumu kombināciju, sevišķi pieaugot iesaistītu tabulu skaitam.

GIT : programma, kas ir paredzēta izmaiņu atsekošanai un kontrolei jebkurā failu kopā. Parasti tiek izmantota programmatūras izstrādātāju komandās, kurās vairāki izstrādātāji kopā veic koda izstrādi. Šīs programmas galvenie mērķi ir : ātrums; datu integritāte; atbalsts sadalītām, nelineārām plūsmām (tūkstoši paralēlu zaru, uzturētu dažādās sistēmās). Nosaukuma skaidrojumam ir vairākas versijas, bet divas populārākās ir subjektīvi atkarīgas no izmantošanas pieredzes, garastāvokļa un citiem faktoriem :

* « Global information tracker » (globāls informācijas atsekotājs) : kad viss izdodas, viss strādā un saglabājas korekti.
* « Goddamn idiotic truckload of sh\*t » : pieļauju ka nav vajadzīgs tulkojums, bet ja nu tomēr -- Google translate :). Šis saīsinājums ir spēkā kad viss lūst un plīst, tiek izmestas kļūdas par saglabājama satura konfliktiem un citām problēmām, kas neļauj veikt failu saglabāšanu.

Repository : Repositorijs, repo. Glabāšanas vieta, paredzēta dažāda satura failu uzglabāšanas, versionēšanas un koplietošanas nodrošināšanai; informāciju tehnoloģiju pasaulē visvairāk tie ir programmatūras koda faili. Programmatūras repozitorijs parasti tiek pārvaldīts ar versiju kontroles un vadības sistēmu, piemēram, GIT.

GIT stage : GIT komanda, kas sagatavo vienu vai vairākus failus saglabāšanai (*commit*). Faili, kuriem ir izpildīta stage komanda, tiek indeksēti un atzīmēti GIT sistēmā (GIT terminoloģijā tie tiek pārvietoti *staging* apgabalā), lai tā var zināt ka šie faili ir jāiekļauj nākamajā saglabāšana, jeb komītā (*commit*).

GIT commit : GIT komanda, kas pārvieto sagatavotus saglabāšanai failus no *staging* apgabala uz saglabāšanas (*commit*) apgabalu. GIT commit komanda izveido izmaiņu GIT lokalā repozitorijā momentuzņēmumu; vēlāk tās izmaiņas var aizsūtīt (*push*) galvenajā repozitorijā.

GIT push : GIT komanda, kas aizsūta pēdējās saglabātas izmaiņas (*committed changes*) no lokāla repozitorija uz galveno attālināto (*remote*) repozitoriju.