****

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 03

## Terminu vārdnīca

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.

RDBMS : Relational Database Management System / Relāciju Datubāzes Vadības Sistēma. Sistēma, kura tiek izmantota relāciju datubāzes uzturēšanai. Daudzas RDBMS izmanto SQL valodu šā uzdevuma veikšanai.

Database instance : Datubāzes instance. Tas ir datubāzes servera atmiņu struktūru apgabals, kas pārvalda datubāzes failus. Līdz ar to ka pati datubāze pēc būtības ir fizisku failu kopa uz servera datora, kas satur datubāzes datus,-- instance pārvalda šos failus un apkalpo datubāzes lietotāju pieslēgumus, sesijas, pieprasījumus utt.

Database schema : Datubāzes shēma. Tā ir datubāzes objektu kolekcija. Shēmas īpašnieks (owner) ir datubāzes lietotājs, un shēmai ir tas pats nosaukums kā šim lietotājam. MySQL RDBMS shēma ir ekvivalents datubāzei, kas nav tipiski priekš RDBMS : parasti datubāzes instancē var būt vairākas shēmas.

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.

Database row : Datubāzes ieraksts. Tā ir lauku kolekcija vai kopa, kas apraksta kādu informācijas vienību. Datubāzes ieraksts pēc būtības ir noteiktas struktūras informācijas komplekts datubāzes tabulā.

Datu tips : datu atribūts, kurš paziņo koda kompilatoram vai interpretatoram veidu kā programmētājs varēs izmantot šos datus. Vairākums programmēšanas valodu atbalsta pamata datu tipus : veseli skaitļi (dažāda garuma), peldoša komata skaitļi (tie aptuveni atspoguļo reālus skaitļus), teksta datu tips un loģisks datu tips (*Boolean*). Datu tips ierobežo vērtības kuras kāda izteiksme, piemēram mainīgais, funkcija u.tml., var glabāt vai apstrādāt.

Table index : Tabulas indekss. Indekss ir neobligāta datu struktūra, kura ir saistīta ar tabulu; tā ir paredzēta piekļuves datiem paātrināšanai dažos gadījumos (tas ir iespējams ne visos gadījumos). Indekss tiek veidots vienam vai vairākiem tabulas laukiem.

Key : Atslēga. Relāciju datubāzēs atslēga ir atribūtu (lauku, kolonnu) kopa, kas unikāli identificē konkrētu tabulas ierakstu. Divi galvenie atslēgu veidi ir primārā (primary) un norādes (foreign) atslēga. Primārā atslēga ir atslēga pašā tabulā, kura kalpo ieraksta unikalitātes nodrošināšanai. Norādes (foreign) atslēga ir primārās atslēgas kopija citā tabulā; tā norāda uz konkrētu ierakstu saistītajā tabulā. Norādes atslēgas var būt unikālas vai neunikālas; pirmajā gadījumā tiek veidota viens pret vienu relācija, otrajā viens pret daudziem.

Table column(s) constraint : Tabulas lauka(-u) ierobežojums. Tas ir datu bāzē definēts noteikums, kas apraksta un nodrošina pieļaujamu saglabāšanai datu ierobežošanas nosacījumus konkrētajā laukā vai lauku kombinācijā. Pastāv šādi galvenie izplatītākie ierobežojumu tipi:

* primārās atslēgas ierobežojums (primary key constraint)
* norādes atslēgas ierobežojums (foreign key constraint)
* unikalitātes ierobežojums (unique value constraint)
* vērtību saraksta ierobežojums (value check constraint)
* vērtības obligātuma ierobežojums (not null value constraint)

Varchar : Datu tips datubāzēs, kurš tiek izmantots lai glabātu mainīga garuma tekstu. Mainīgā definīcijā tiek norādīts maksimālais pieļaujamais teksta garums, bet reāli tas aizņem datora atmiņā tika daudz vietas, cik ir nepieciešams konkrēta teksta vērtības glabāšanai.

Success / Failure : Programmētāju žargons, burtiski nozīmē Veiksme / Neveiksme. Parasti apzīmē kādas darbības, programmas daļas, testa u.tml. gala rezultātu.

Regular expression : Regulāra izteiksme. Tā ir simbolu virkne, pierakstīta speciālā notācijā, kura definē teksta meklēšanas paraugu.

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.