****

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 07

## 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.

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.

Temporary table : Pagaidu tabula. Tā ir tabula, kura eksistē tikai uz klienta sesijas laiku. Tā ir redzama tikai tam klientam, kurš to izveidojis, līdz ar to tā nekādā veidā neietekmē citus datubāzes lietotājus. To var izmantot lai aizvietotu oriģinālo tabulu, izveidojot tās pilnīgi precīzu kopiju ar tādiem pašiem datiem.

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.

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.

DDL : Data Definition Language / Datu Definēšanas Valoda. Tā ir SQL valodas daļa, kura nodrošina datu struktūru definēšanu. Izmantojot šīs valodas komandas ir iespējams veidot, mainīt un dzēst datubāzes objektus. DDL galvenie atslēgvārdi ir : CREATE (TABLE / INDEX / CONSTRAINT / VIEW ...), ALTER TABLE, DROP (TABLE / INDEX / VIEW ...).

DML : Data Manipulation Language / Datu Manipulēšanas Valoda. Tā ir SQL valodas daļa, kura nodrošina darbības ar datiem. Izmantojot šīs valodas komandas ir iespējams atlasīt, pievienot, mainīt un dzēst datus datubāzē. DML galvenie atslēgvārdi ir : INSERT, UPDATE, DELETE, SELECT, COMMIT, ROLLBACK.

Commit command : Izmaiņu apstiprināšanas komanda. Relāciju datubāzē DML operācijas tiek apvienotas transakcijās. Apstiprinot izmaiņas, transakcija tiek pabeigta, un datu izmaiņas kuras ir iekļautas tajā tiek pastāvīgi saglabātas, kas savukārt nozīmē ka tās kļūst pieejamas visiem datubāzes lietotājiem.

Rollback command : Izmaiņu atcelšanas komanda. Sasniedz pretējo rezultātu nekā commit komanda. Visas nesaglabātas izmaiņas tiek atceltas un pazūd, kļūstot nepieejamas visiem datubāzes lietotājiem.

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)

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.