****

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 06

## 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 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.

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.

Forward engineering : Programmatūras veidošanas process, kurā par pamatu tiek izmantots augsta līmeņa modelis, un būvēšanas process notiek no augšas uz leju, no augsta līmeņa modeļa uz lielākas detalizācijas un sarežģītības pakāpes līmeni. Datubāžu izstrādē klasiski šo metodi izmanto veidojot datubāzes struktūru : par pamatu tiek ņemts ERD modelis, no kura tiek uzģenerēts SQL DDL kods, ar kuru palīdzību tiek veidoti datubāzes objekti.

Entity Relationship Diagram (ERD) : Vienību Relāciju Diagramma. Tā ir datu modeļa diagramma, kura apraksta savstarpēji saistītas datu vienības specifiskā informācijas apgabalā. ER diagramma sastāv no datu vienību tipiem (entity types; klasificē datu vienības) un saitēm (relācijām) kas var pastāvēt starp datu vienībām (datu vienību tipu instancēm). Programmatūras izstrādē ER diagramma tiek izmantota lai atspoguļot un saglabāt konkrētam biznesam vajadzīgas lietas biznesa procesu nodrošināšanai; līdz ar to tā dabiski kļūst par abstraktu datu modeli, kura definē informācijas struktūru ko var implementēt datubāzē, parasti relāciju datubāzē.

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 ...).

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)

Query execution plan : Vaicājuma izpildes plāns. Soļu secība, kura tiek izmantota lai piekļūtu datiem pieprasītiem SQL vaicājumā relāciju datubāzē. To nosaka konkrēta RDBMS, izmantojot speciālu rīku, ko sauc par vaicājumu optimizētāju : tas novērtē datu struktūras (tabulas, indeksus) un datus un izvēlas iespējami labāko un ātrāko izpildes plānu. Liela nozīme ir gan datu apjomam, gan indeksiem, gan statistikas datiem ar kuriem operē optimizētājs : tas neveic pilnu datu lasīšanu (tas prasītu pārāk daudz laika), tā vietā izmantojot ievāktus statistiskus datus par datiem tabulās. Līdz ar to ir svarīgi pareizi nokonfigurēt RDBMS, lai statistikas dati būtu regulāri atjaunoti, jo novecojuši un neaktuāli tie var novest pie nopietnām kļūdām vaicājumu optimizētāja darbības rezultātos. Optimizētājs izvēlas vislabāko izpildes plānu vairumā gadījumu, bet ne vienmēr. Jāpiemin arī, ka vaicājumu ir iespējams optimizēt, to pārrakstot, izveidojot trūkstošus indeksus un vēl ar citiem paņēmieniem, proti, ļoti daudz kas ir atkarīgs no datubāzes izstrādātāja kompetences un pieredzes.

Tables join : Tabulu savienošana. Lai atlasītu datus no vairākām tabulām vienlaicīgi, izmanto tabulu savienošanu SQL vaicājumā. Tabulu savienošana notiek ar primāru un norāžu atslēgu palīdzību.

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.