****

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 08

## Terminu vārdnīca

Funkcija : Programmas, jeb koda instrukciju kopa, kas ir apvienoti ar mērķi sasniegt konkrētu rezultātu; funkcijas tiek veidotas ar mērķi atkārtoti izmantot kodu, novērst vienādu koda gabalu rakstīšanu. Funkcija definē parametrus (katram parametram tiek definēts nosaukums un datu tips) un atgriežama rezultāta datu tipu. Datubāzēs funkcijas tiek izmantotas gan procedurālās programmēšanas valodās, veidojot biznesa loģiku, gan SQL vaicājumos lai paveiktu datu apstrādi / formatēšanu. Mūsdienās visās programmēšanas valodās ir pieejams plašs iebūvēto funkciju klāsts; tajā pašā laikā teju katrā IT sistēmā izstrādes laikā rodas vajadzība pēc specifiskām tieši šai sistēmai funkcijām, kuras tad arī tiek izveidotas.

Agregācijas funkcija : SQL valodas specifisks termins. Tā ir funkcija, kura operē ar viena lauka vērtībām no vairākiem ierakstiem noteiktā ierakstu grupā, atgriežot vienu vērtību. Visbiežāk izmantotas agregācijas funkcijas ir MIN, MAX, SUM. Dažas agregācijas funkcijas operē tikai ar skaitliskām vērtībām; dažas operē ar jebkādu primitīvu datu tipu, piemēram, MIN un MAX. Pastāv arī specifiskas agregācijas funkcijas, kuras operē, piemērām, ar teksta datiem. Ierakstu grupas tiek veidotas vaicājuma GROUP BY daļas nosacījumos; gadījumā ja GROUP BY daļas vaicājumā nav, tad pastāv viena ierakstu grupa, proti, visi vaicājuma atlasīti ieraksti.

Analītiska funkcija : SQL valodas specifisks termins. Līdzīgi kā agregācijas funkcijas, tās aprēķina vairāku ierakstu kopīgu vērtību vienam laukam. Piemēri ir tie paši : MIN, MAX, SUM, AVG utt. Atšķirībā no agregācijas funkcijām, analītiskajās funkcijās ir paplašinātā notācija, kura ļauj definēt ierakstu grupēšanas nosacījumus pašā funkcijā, tādējādi dodot iespēju iegūt kopīgas vērtības ierakstu grupā bez tiešas ierakstu apvienošanas grupās, proti, saglabājot pilnu ierakstu atlasi. Tādā veidā ir iespējams, piemēram, atlasīt visus kompānijas darbiniekus, sarēķinot vidējo, maksimālo un minimālo darbinieku algu katrā departamentā. Analītisku funkciju notācija atļauj definēt ne tikai ierakstu grupēšanas (GROUP BY), bet arī sakārtošanas (ORDER BY) nosacījumus, kā arī ierobežot ierakstu skaitu grupā sākot no konkrēta ieraksta skaita pirms tekošā ieraksta un beidzot ar konkrētu ierakstu skaitu pēc tekošā ieraksta -- to definē tā saucami "loga nosacījumi" (window conditions).

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.

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.

OLTP : On-Line Transaction Processing (Tiešsaistes transakciju apstrāde). Tas ir tāds datu apstrādes veids, kurā notiek vairāku vienlaicīgi notiekošu transakciju izpilde, piemēram, banku transakcijas, iepirkumi, pasūtījumu ievade, teksta ziņu sūtīšana u.tt. u.tml. OLTP sistēmas tiek projektētas un veidotas tādā veidā, lai sasniegtu OLTP mērķus : ātra, efektīva un droša transakciju datu apstrāde, kā arī datu efektīva glabāšana. Datu modelī izmanto normalizāciju, lai izvairītos no lieku, atkārtojušos datu glabāšanas.

OLAP : On-Line Analytical Processing (Tiešsaistes analītiska apstrāde). Tas ir tāds datu apstrādes veids, kura mērķis ir sniegt multi-dimensionālas atbildes, citiem vārdiem, veikt datu analīzi. OLAP sistēmas parasti tiek pretnostatītas OLTP sistēmām : OLAP sistēmas ir optimizētas datu analīzei un izgūšanai, bet OLTP mērķis ir efektīvi nodrošināt visa veida datu apstrādi, proti, gan lasīšanu, gan pievienošanu, mainīšanu un dzēšanu. OLAP sistēmās datu struktūra bieži tiek transformēta, un transformācijas rezultāts mēdz būt denormalizācija; tas tiek darīts ar nolūku padarīt datu analīzi ātrāku un efektīvāku.

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.