

RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

2.praktiskais darbs

mācību priekšmetā

“Datu bāzu pamati un Relāciju datu bāzes (DB1)”

**Datu glabāšanas struktūras**

Izstrādāja: Viktors Stepičevs

< II.kurss RDBD0, 191RDB100>

Pārbaudīja: lektors A.Auziņš

2020./21. māc. gads

# 

# Saturs

[Saturs 2](#_Toc67494017)

[Uzdevums - Datu glabāšanas struktūru izpēte 3](#_Toc67494018)

[Anotācija. 4](#_Toc67494019)

[Kaudzes tipa tabulas 5](#_Toc67494020)

[Tabula ar indeksa organizāciju 9](#_Toc67494021)

[Indeksa tipa klasteris 10](#_Toc67494022)

[Klasteris ar heš funkcijas izmantošanu 11](#_Toc67494023)

[Secinājumi 13](#_Toc67494024)

[Literatūra 14](#_Toc67494025)

# Uzdevums - Datu glabāšanas struktūru izpēte

Darbā jāanalizē šādas datu glabāšanas struktūras:

1) parastas kaudzes (heap) tipa tabulas. Divas tabulas, starp kurām ir saite viens ar daudziem (1 : N).

2) tabula ar indeksa organizāciju. Sakārtota tabula, kura apvieno indeksa funkcijas ar pārējo vērtību glabāšanu.

3) indeksa tipa klasteris. Klasteris apvieno divas tabulas. Indeksēšanai tiek izmantots B – koka tipa indekss (balansēts indekss).  
4) klasteris ar heš (heash) funkcijas izmantošanu. Klasteris apvieno divas tabulas. Sasaistei tiek izmantota heš – funkcijas adresācija.

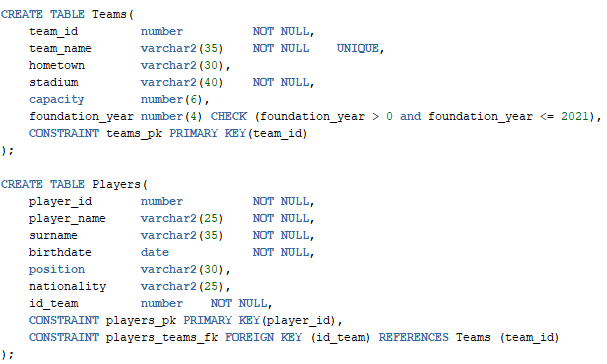
Jādefinē tabulas (CREATE TABLE), klasteri (CREATE CLUSTER) un indeksi (CREATE INDEX).  
Jāievada (INSERT)tabūlas datu rindas (vismaz 10). Vienus un tos pašus datus jāizmanto visās glabāšanas struktūrās, lai varētu veikt salīdzinājumus.  
Jāizveido 3 vaicājumi (SELECT) ar kuriem:  
1) tiek iegūti visi raksti;  
2) tiek iegūts rakstu kopējais skaits;  
3) tiek iegūts viens raksts.  
Jāfiksē vaicājuma izpildes laiks un izpildes plāna izmaksas (COST).  
Jāizdara secinājumi.

# Anotācija.

Izmantojot Oracle SQL Developer darba uzdevumu izpildēs gaitā būs analizētas vairākas datu glabāšanas struktūras, ar kuru palīdzību ir domāts optimizēt datu bāžu vadības sistēmu darbību, uzlabojot vaicājumu izpildes laiku un samazinājot nepieciešamas izmaksas.

# Kaudzes tipa tabulas

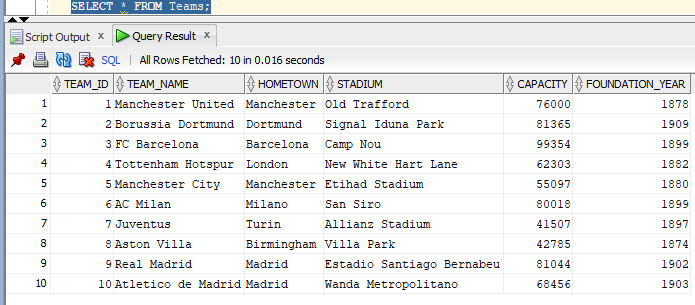
Kopīgi sakot, tika izveidotas divas tabulas Players (Spēlētāji) un Teams (Komandas), kuras glāba informāciju par dažiem futbolistiem un komandām. Teams tabulā par atribūtiem tika definēti PK komandas identifikators “team\_id”, komandas nosaukums “team\_name”, mājas pilsēta “hometown”, komandas mājas stadions “stadium”, tā stadiona kapacitāte “capacity” un komandas dibināšanas gads “foundation\_year”. Tika paņemts datu tips “number” ar četru simbolu ierobežojumu, jo ne vienmēr var noteikt tiešo izveidošanas datumu, bet ar gadu ir vieglāk. Arī ar CHECK komandu tika pārbaudīta gada kolonnā ievadīto datu semantika, neļaujot gadam būt negatīvam un lielākām par esošo 2021. gadu. Stadiona kapacitātei tika piešķirts limits 6, jo lielākās sporta pasaules arēnas pagaidām var ietilpst apmērām 100000, dažas arī nedaudz vairāk. Komandas nosaukums tika definēt kā UNIQUE, jo ir saprotams, kā nebūs vairākas komandas ar vienu un to pašu identisko nosaukumu. Runājot par tabulu Players, par atribūtiem tika izvēlēti spēlētāja identifikātors “player\_id”, spēlētāja vārds un uzvārds “player\_name” un “surname”, dzimšanas datums “birthdate” ar datu tipu DATE, spēlētāja pozīcija “position”, nacionalitāte “nationatily” un FK “id\_team”, lai spēlētājiem piešķirtu komandu no Teams tabulas.

.

Darba gaitā tiks izmantotas divas tabulas ar vieniem un tiem pašiem atribūtiem un ierobežojumiem, kā arī ar vienādiem ievades datiem. Abās tabulās tiks ievadītas 10 rindas.



Izmantotie ievades dati.



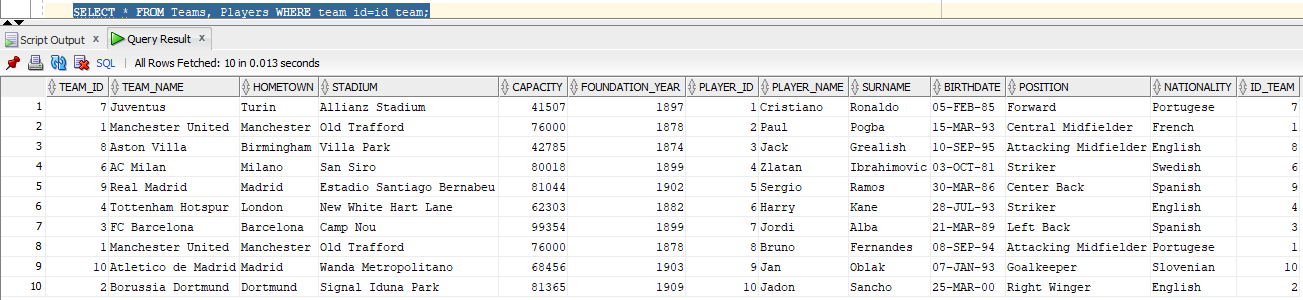
Tabulu testa izvade ar ievietotājiem datiem.



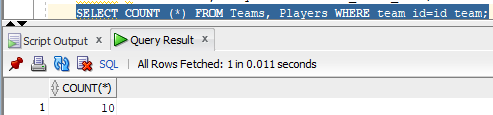
Definējot abas tabulas un ievadot datus tika izveidoti trīs vaicājumi, lai salīdzinātu datu struktūra efektivitāti, novērojot vaicājuma izpildes laiku un izmaksas COST.

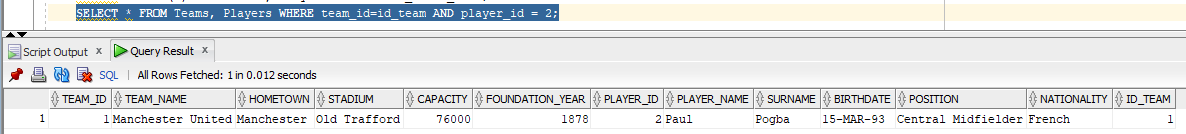


Pirmais vaicājums iegūst visus tabulas rakstus, otrais iegūst kopējo ierakstu skaitu, bet trešais iegūst tikai vienu rakstu. Ņemot vērā to faktu, kā vaicājuma izpildes laiki pēc katras izpildes reizes atšķiras, es izpildīju katru vaicājumu 5 reizes un paņēmu vidējo rezultātu, jo tas varētu būt uzskatīts par “patieso” vērtību, nekā ņemot pirmo un vienīgo rezultātu.

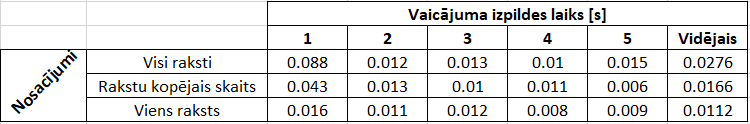


Vaicājumu izpilde.

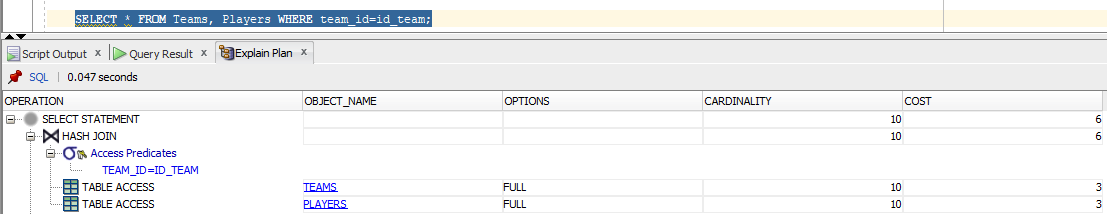


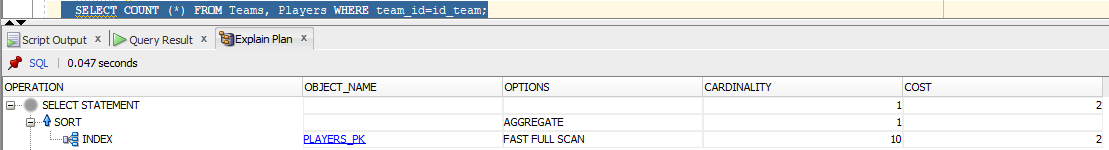


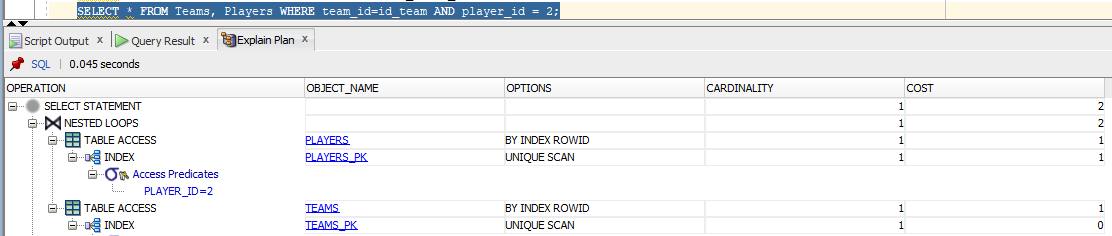
Kā ir redzams, vaicājumi strādā korekti, ir arī novēroti daži spēlētāji ar vienādo pārstāvēto komandu.



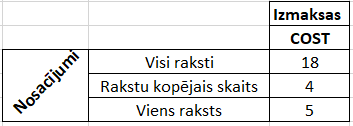
Analizējot iegūtus rezultātus ir redzams, kā visos gadījumos pirmā vaicājuma izpildes reize aizņēma novērojami vairāk laika, kas uzlabojas un svārstās ap vidējo sākot no otrās izpildes reizes. Analizējot vidējo laiku, visu rakstu iegūšana ar lielo pārsvaru aizņēma vislielāko laiku, pēc tām ir rakstu kopējo skaitu iegūšana un visātrākais izpildes laiks no šiem vaicājumiem bija viena raksta iegūšana, kura izpildījās apmērām 2,5 reizēs ātrāk nekā pirmais vaicājums.







Ar F10 taustiņas palīdzību ir iespējams izpildīt “Explain Plan” vaicājuma izpildes plāna izmaksas, kuru rezultāti ir redzami tabulā.

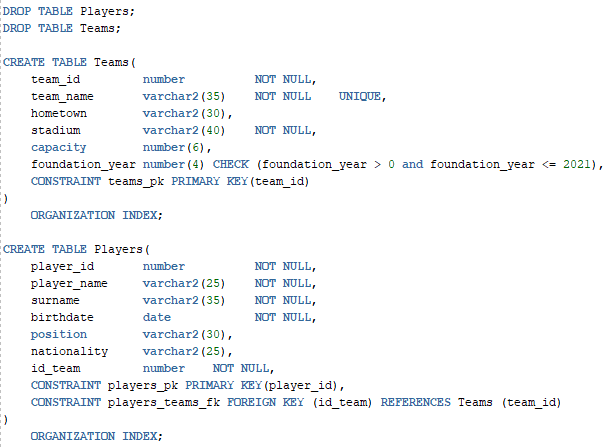


Apkopojot datus par izmaksām, visas iegūtās vērtībās atbilstošajām vaicājumam tika saskaitītas kopā un apspoguļotās tabulā tālākai salīdzināšanai.

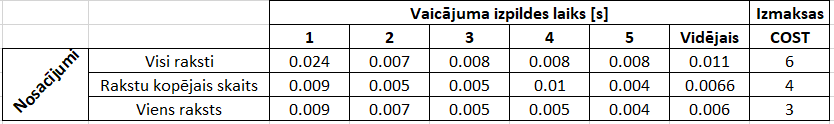
# Tabula ar indeksa organizāciju

Kā jau bija teikts, par visu tabulu pamatu būs paņemtas vienādas tabulas un balstoties uz tiem būs veiktas nepieciešamas pārmaiņas, lai pārveidotu tos atbilstošajā datu struktūrā. Pirms uzsākot turpmāko darbu ir jānodzēš iepriekšējas tabulas ar komandas DROP palīdzību. Pievienojot komandu ORGANISATION INDEX tabula tiek pārveidota par indeksa organizētu.

Indeksa organizēta tabula ir uzskatāma par tabulu ar pilno funkcionalitāti. Par vienu no galvenam priekšrocībām varētu uzskatīt to, kā tai tabulai ir ātrā piekļuve PK, kas ir iegūts samērā ātri izmantojot indeksa skanēšanu.



Analizējot iegūtos datu, atkal ir redzama tendence, ka vaicājuma izpildes laiks pirmā reizes laikā ir ievērojami lielāks par vidējo. No otrā atkārtojuma vērtības saka svārsties ap vidējo, ko jau var uzskatīt par optimalitāti. Salīdzinot vērtības ar kaudzes tipa tabulas iegūtajiem rezultātiem ir novērojams būtiskais uzlabojums. Visu vaicājumu gadījumā izpildes laiks samazinājās vairāk nekā 2 reizēs. Interesanti pievienot to, kā samaksas salīdzinājumā ar iepriekšējo datu struktūru arī samazinājās, kas dod iespēju secināt, kā indeksa organizēta tabula ir “optimizētākā” par kaudzes tipa tabulu. Lielākā atšķirībā ir tā, kā indeksa organizētā tabulā PK ir jābūt obligāti definētam un ar to palīdzību ir iespējams ātri piekļūt rindai.

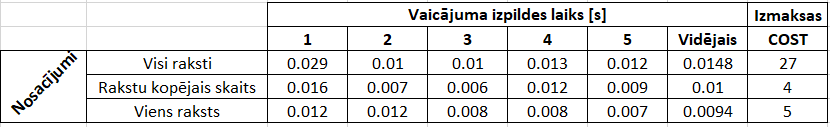


# Indeksa tipa klasteris

Indeksa tipa klasteris ir tabulu klasteris, kurš ar klastera indeksa palīdzību atrod visus nepieciešamos datus. To izveidojot, datu rindas tiek apkopoti grupētājā tabulā.

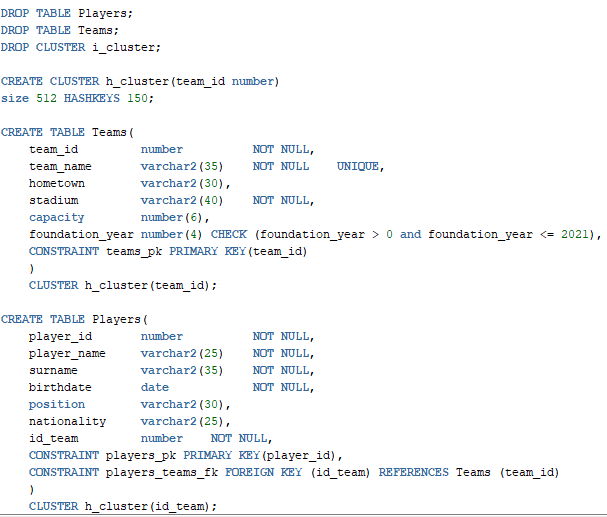


Salīdzinot novērotus rezultātus, ir iespējams redzēt, kā analizējot vidējo laiku indeksa tipa klasteris visās pozīcijās ir ātrāk nekā kaudzes tipa tabula, tomēr atšķirība nav tik liela kā iepriekšējā salīdzinājumā. Skatoties uz tabulas ar indeksa organizāciju rezultātiem ir redzams, kā indeksa tipa klasteris tām zaudē, tomēr godīgi būtu pateikt kā ne ar tādu lielu atšķirību. Tomēr, izmaksas departamentā pirmajā tās ir līdzīgas ar pirmo eksperimentu otrajā un trešajā vaicājumā, bet pirmajā tās ir lielākās. Aplūkojot izmaksas rezultātus indeksa organizētai tabulai, kopumā var secināt, kā tie ir mazāki nekā indeksa tipa klasterim. Otrajā vaicājuma izmaksas ir līdzīgas, trešajā vaicājuma atšķirība nav novērojami liela, bet skatoties uz pirmā vaicājuma rezultātiem, indeksa organizēta tabula ir vairāk nekā četras reizes izdevīgākā, kas tai dod ievērojamo priekšroku.

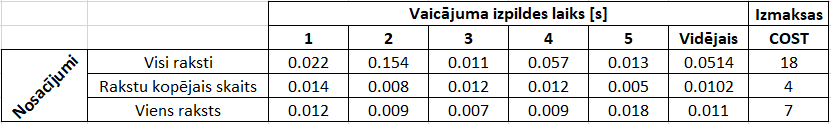


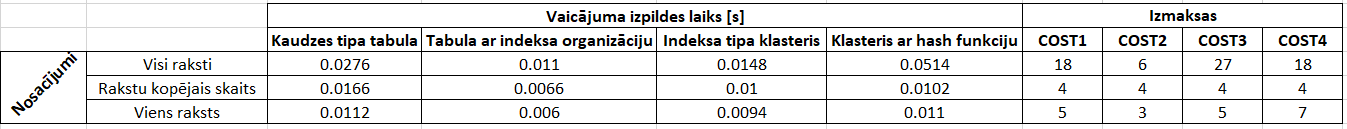
# Klasteris ar heš funkcijas izmantošanu

Runājot par hash klasteri, tas atšķirība no indeksa tipa klastera ir tā, ka hash klasteris izmanto hash funkciju(izmantota hash funkcijas vērtība indeksa tabulas vietā), kura atrod nepieciešamo datu glabāšanas adresi(tiešo vietu diskā). Tai vērtībai klasterī ir atbilstošs datu bloks, kurš ir izmantots ar datu bāzi vaicājuma laikā. Hash funkcijas izmantošana ir efektīva, kad tabula nav bieži modificēta, tiešāk sakot gadījumā jā tā ir izsaukta biežāk nekā koriģēta.



Kas man pārādījas interesants, kā izpildes laiks šai datu struktūrai ir novērojami lielāks nekā indeksa tipa klasterim un indeksa organizētai tabulai. Varbūt varētu pateikt, ka atšķirība nav būtiska rakstu skaita un viena raksta iegūšana, bet iegūstot visus raktus atšķirība ir redzama. Iegūstot visus rakstus šī datu struktūra izrādīja sevi par vislēnāko. Analizējot hash funkcijas klastera izpildes laikus ir novērojama liela svārstība rezultātos, kas acīmredzami ietekmēja vidējo laiku. Analizējot šīs datu struktūras izmaksas, tos varētu salīdzināt ar kaudzes tipa tabulas rezultātiem, jo tie ir gandrīz identiski, bet tomēr manā eksperimenta kontekstā tās nav tik tuvas indeksa organizētas tabulas rezultātiem.





Rezultātus apkopojoša tabula.

# Secinājumi

Analizējot darba gaitu pēc uzdevumu izpildes, es varu secināt, kā iegūta pieredze pirmo reizi strādājot ar Oracle SQL Developer ir ļoti vērtīga, jo uzsākot darbu man nebija iepriekšējas pieredzes strādājot ar to rīku. Ar vairāku mācību resursu izpētes laiku strādāt kļuva vieglāk, nekā darba sākumā, kad dažas darbības aizņēma vairāku laiku, bet ar pakāpeniski iegūto pieredzi vairāk sāka rasties skaidrībā par dažam ne visam skaidrām lietām un izpildāmam darbībām.

Aplūkojot darba gaitā radītās problēmas, laiku pa laiku man radījās dažas grūtības ar radīto kļūdu korekciju, ko varētu paskaidrot ar mazo pieredzi darbā ar datu bāžu vadības sistēmām. Bet kopumā, kā ierasti un gadās ar laiku izpildīt darba uzdevumus un atrisināt dažādas problēmas kļuva vieglāk.

Kopumā sakot par datu struktūrām, varu secināt, ka tiem patiešām ir lietderīgi pielietojumi, kuri ir atkarīgi no vairākiem datu bāzes niansēm un īpašībām. Izmantojot lekcijas konspektus un materiālus internetā ir lieliskā iespēja atrast visu nepieciešamo informāciju, lai apgūtu nepieciešamas zināšanas. Darbs kopumā sākumā man aizņēma daudz laika, bet ar laiku kļuva vieglāk un darba gaita sāka iet gludāk. Esmu pārliecināts, ka ar to pieredzi nākotnē izpildīt darbus man būs vieglāk, es esmu saņēmis daļu no nepieciešamo zināšanu bāzes darba ar SQL valodu.

# Literatūra

1. Prof. J. Eiduks. Lekciju konspekti priekšmeta “Datu bāzes vadības sistēmas”;
2. <https://datubaze.wordpress.com/>
3. estudijas.rtu.lv
4. <https://docs.oracle.com/cd/B28359_01/server.111/b28310/tables003.htm#ADMIN01503>
5. <https://docs.oracle.com/cd/B28359_01/server.111/b28310/indexes001.htm#ADMIN11709>
6. <https://docs.oracle.com/cd/B28359_01/server.111/b28310/clustrs001.htm#ADMIN11739>
7. <https://docs.oracle.com/cd/B28359_01/server.111/b28310/hash001.htm#i1006523>