

RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

2.praktiskais darbs

mācību priekšmetā

“Datu bāzes tehnoloģijas pamati (DB1)”

**SQL vaicājumi**

Izstrādāja: Jolanta Bērziņa

Stud.apl.nr 181RDB178

Pārbaudīja: lektors A.Auziņš

2019./2020. m.g.

**Saturs**

[1. Uzdevums 2](#_Toc41848220)

[1.1. Darba mērķis 2](#_Toc41848221)

[1.2. Veicamie uzdevumi 2](#_Toc41848222)

[2. Vaicājumi 3](#_Toc41848223)

[2.1. Komandu “create, insert, update” izmantošana 3](#_Toc41848224)

[2.1.1. Komandas CREATE izmantošana 3](#_Toc41848225)

[2.1.2. Komandas INSERT izmantošana 4](#_Toc41848226)

[2.1.3. Komandas UPDATE izmantošana 5](#_Toc41848227)

[2.2. SQL vaicājumi, kuros ir uzstādīti 2 un vairāk kritēriji 6](#_Toc41848228)

[2.3. From ar 2 un vairāk tabulām 7](#_Toc41848229)

[2.4. From ar ārējo savienošanu 9](#_Toc41848230)

[2.5. Apakšvaicājumi Where vai Having daļā 10](#_Toc41848231)

[2.6. Apakšvaicājumi Select vai From daļā 11](#_Toc41848232)

[2.7. Having pareiza izmantošana 12](#_Toc41848233)

[2.8. Datu sakārtošana 12](#_Toc41848234)

[2.9. Funkciju izmantošana (piemēram Sum(), Count()) dažādās vaicājuma sadaļās 13](#_Toc41848235)

[2.10. Predikātu izmantošana (LIKE ar “?”. “\*”, &; ANY; ALL; IN; Distinct) 14](#_Toc41848236)

[2.11. Aritmētisko operācioju izmantošana 15](#_Toc41848237)

[2.12. Rezultātu apvienošana UNION query 15](#_Toc41848238)

[2.13. Šķērsgriezuma tabulas izveide ar *transform/pivot* konstrukcijas izmantošanu (Crosstab Query) 16](#_Toc41848239)

[Secinājumi 17](#_Toc41848240)

# Uzdevums

## Darba mērķis

2.prakstiskā darba studiju priekšmetā “Datu bāzu vadības sistēmas” mērķis ir iepazīt, pielietot un apgūt dažādas SQL valodas konstrukcijas. Šī mērķa sasniegšanai in nepieciešams apskatīt un veikt variētus SQL vaicājumus ar dažādām konstrukcijām un izmantojot operandus, realizēt apakšvaicājumus .

## Veicamie uzdevumi

1. Izmantot komandas **create, insert, update**. Izveidot 3 tabulas (izveidot struktūru, definēt ierobežojumus, ievadīt datus, atjaunot datus). Šīs tabulas turpmākajā darbā ar SQL var neizmantot.
2. SQL vaicājumā uzstādīti 2 un vairāk kritēriji (vismaz 3 vaicājumi) (*loģiskās un salīdzināšanas operācijas*).
3. From ar 2 un vairāk tabulām (vismaz 6 vaicājumi, vismaz 1 ar saiti From daļā (Microsoft pieraksts)).
4. From ar ārējo savienošanu (vismaz 2 piemēri).
5. Apakšvaicājumi Where vai Having daļā (vismaz 2 dažādi salīdzināšanas operatori, vēlams izmantot gan vienkāršu salīdzināšanu, gan salīdzināšanu, izmantojot kopu operatoru).
6. Apakšvaicājumi Select vai From daļā (vismaz 2 piemēri).
7. Having pareiza izmantošana (vismaz 2 piemēri, katrs nepareizs piemērs novelk novērtējumu uz leju).
8. Datu sakārtošana (vismaz 2 piemēri) (*izmantot dažādas izteiktsmes*)
9. Funkciju izmantošana (piemēram Sum(), Count()) dažādās vaicājuma sadaļās (vismaz 3 dažādas funkcijas).
10. Predikātu izmantošana (LIKE ar “?”. “\*”, &; ANY; ALL; IN; Distinct) (vismaz 2 dažādi predikāti) Piem, divu sarakstu salīdzināšana.
11. Aritmētisko operācioju izmantošana (vismaz 2 piemēri)
12. Rezultātu apvienošana UNION query (vismaz 1 piemērs).
13. Šķērsgriezuma tabulas izveide ar **transform/pivot** konstrukcijas izmantošanu (Crosstab Query).
14. Secinājumi (Aizpildīt rezultējošo tabulu + secinājumi par darbu).

# Vaicājumi

## Komandu “create, insert, update” izmantošana

### Komandas CREATE izmantošana

Tabulas GAMES izveidošanas skrips un izveidotā tabula (skat. 2.1.att):

CREATE TABLE GAMES

(

GAME\_ID INT NOT NULL,

BLUETEAM INT NOT NULL,

REDTEAM INT NOT NULL,

WINNER INT

);

|  |
| --- |
|  |
| 2.1.attēls.Izveidota tabula GAMES |

Tabulas MEETINGS izveidošanas skripts un izveidotā tabula (skat 2.2.att.).:

CREATE TABLE MEETINGS

(

FANS\_ID INT NOT NULL,

TEAM\_ID INT NOT NULL,

TEAM\_NAME VARCHAR(30),

DATUMS DATE,

LOCATION VARCHAR (30)

);

|  |
| --- |
|  |
| 2.2.attēls. Izveidota tabula MEETINGS |

Tabulas INTERVIEWS izveidošanas skripts un izveidotā tabula (skat. 2.3.att.).

CREATE TABLE INTERVIEWS

(

INTERVIEW\_ID INT NOT NULL,

INTERVIEWER\_NAME VARCHAR (30),

INTERVIEWER\_SURNAME VARCHAR(30),

I\_TIME TIME,

TEAM\_ID INT NOT NULL

);

|  |
| --- |
|  |
| 2.3.attēls. Izveidota tabula INTERVIEWS |

### Komandas INSERT izmantošana

Datu ievietošana tabulā GAMES skripts un tabula pēc datu ievietošanas (skat. 2.4.att.).

INSERT INTO GAMES ( GAME\_ID, BLUETEAM, REDTEAM, WINNER ) VALUES (1 ,2, 3 ,3);

INSERT INTO GAMES ( GAME\_ID, BLUETEAM, REDTEAM, WINNER ) VALUES (2, 3 ,4, 4);

INSERT INTO GAMES ( GAME\_ID, BLUETEAM, REDTEAM, WINNER ) VALUES ( 3, 2, 5, 2);

|  |
| --- |
|  |
| 2.4.attēls. Tabulas GAMES jaunie ieraksti |

Datu ievieto;sana tabulā INTERVIEWS skripts un tabula pēc datu ievietošanas (skat. 2.5. att.).

INSERT INTO INTERVIEWS (INTERVIEW\_ID, INTERVIEWER\_NAME, INTERVIEWER\_SURNAME, I\_TIME, TEAM\_ID) VALUES (1, "Charlie", "Nemetzski", #2020-04-21 14:25:00#, 2);

INSERT INTO INTERVIEWS (INTERVIEW\_ID, INTERVIEWER\_NAME, INTERVIEWER\_SURNAME, I\_TIME, TEAM\_ID) VALUES (2, "Charlie", "Nemetzski", #2020-04-21 14:50:00#, 4);

INSERT INTO INTERVIEWS (INTERVIEW\_ID, INTERVIEWER\_NAME, INTERVIEWER\_SURNAME, I\_TIME, TEAM\_ID) VALUES (2, "Charlie", "Nemetzski", #2020-04-21 15:20:00#, 5);

|  |
| --- |
|  |
| 2.5.attēls. Tabulas INTERVIEWS jaunie ieraksti |

Datu ievietošana tabulā MEETINGS skripts un tabula ar jaunajiem ierakstiem (skat. 2.6.att).

INSERT INTO MEETINGS (FANS\_ID, TEAM\_ID, TEAM\_NAME, DATUMS, LOCATION) VALUES (1, 1, "Dudes" ,#2020-06-21#, "London");

INSERT INTO MEETINGS (FANS\_ID, TEAM\_ID, TEAM\_NAME, DATUMS, LOCATION) VALUES (2, 8, "Kitties" ,#2020-06-21#, "London");

INSERT INTO MEETINGS (FANS\_ID, TEAM\_ID, TEAM\_NAME, DATUMS, LOCATION) VALUES (3, 8, "Kitties" ,#2020-06-21#, "London");

|  |
| --- |
|  |
| 2.6.attēls. Tabulas MEETINGS jaunie ieraksti |

### Komandas UPDATE izmantošana

Ar komandas UPDATE palīdzību tiek izmainīts laks , kad notiek intervija ar komandu 5, tabulā INTERVIEWS (skat. 2.7.att.).

UPDATE INTERVIEWS SET I\_TIME=#2020-04-21 16:00:00# WHERE I\_TIME =#2020-04-21 15:20:00#;

|  |
| --- |
|  |
| 2.7.attēls. Izmainīts laiks tabulā INTERVIEWS |

## SQL vaicājumi, kuros ir uzstādīti 2 un vairāk kritēriji

SELECT \*

FROM SPELETAJS

WHERE KILLS>=5 AND DEATHS<=7;

Ar šo vaicājumu tika iegūti sekojoši spēlētāji (skat. 2.8.att)

|  |
| --- |
|  |
| 2.8.attēls. Spēlētāji ar vismaz 5 KILLS un ar ne vairāk kā 7 DEATHS |

SELECT TURNIRS.TURNIRA\_NOS, PIEDALAS.K\_NOSAUKUMS

FROM TURNIRS, PIEDALAS

WHERE TURNIRS.TURNIRS\_ID=PIEDALAS.TURNIRS\_ID AND PIEDALAS.TURNIRS\_ID=1;

Ar šo vaicājumu tika iegūti sekojoši dati (skat.2.9.att):

|  |
| --- |
|  |
| 2.9.attēls. Saraksts ar komandām, kas piedalās turnīrā ar TURNIRS\_ID=1, un turnīra nosaukums |

SELECT TURNIRS.TURNIRA\_NOS, PIEDALAS.K\_NOSAUKUMS

FROM TURNIRS, PIEDALAS

WHERE TURNIRS.TURNIRS\_ID=PIEDALAS.TURNIRS\_ID AND PIEDALAS.KOMANDA\_ID=6;

Ar šo vaicājumu tiek rasti dati par komandu ar id=6, un turnīriem, kuros tā spēlē (skat.2.10.att).

|  |
| --- |
|  |
| 2.10.att. Saraksts ar turnīriem, kuros piedalās komanda ar KOMANDA\_ID=6 |

## From ar 2 un vairāk tabulām

SELECT TURNIRS.TURNIRA\_NOS, PIEDALAS.K\_NOSAUKUMS

FROM TURNIRS, PIEDALAS

WHERE TURNIRS.TURNIRS\_ID=PIEDALAS.TURNIRS\_ID AND PIEDALAS.TURNIRS\_ID=1;

Ar šo vaicājumu tika iegūti sekojoši dati (skat.2.9.att):

|  |
| --- |
|  |
| 2.11.attēls. Saraksts ar komandām, kas piedalās turnīrā ar TURNIRS\_ID=1, un turnīra nosaukums |

SELECT F\_VARDS, SP\_NICKNAME

FROM FANS, SPELETAJS

WHERE FANS.F\_SPELETAJA\_ID=SPELETAJS\_ID;

Ar šo vaicājumu iegūst fana vārdu un viņa mīļākā spēlētāja nickname (skat.2.12.att).

|  |
| --- |
|  |
| 2.12.attēls. Fanu vārdi un to mīļāko spēlētāju nicknames |

SELECT PRODUKTA\_NOS, K\_NOSAUKUMS

FROM PRODUKTS, KOMANDA, SPONSORS

WHERE KOMANDA.KOMANDA\_ID=SPONSORS.KOMANDA\_ID AND SPONSORS.PRODUKTA\_ID=PRODUKTS.PRODUKTA\_ID;

Ar šo vaicājumu tiek noskaidroti produktu nosaukumi un komandu nosaukumi, kuru atbalstošie sponsori reklamē

|  |
| --- |
|  |
| 2.13.attēls. Komandu nosaukumi produktu nosaukumi, kurus reklamē komandu atbalsotšie sponsori |

SELECT SP\_VARDS, SP\_UZVARDS, SP\_NICKNAME, DATUMS

FROM SPELETAJS, MEETINGS

WHERE SPELETAJS.KOMANDA\_ID=MEETINGS.TEAM\_ID;

Ar šo vaicājumu tiek iegūti spēlētāju vārdi, uzvādi, nicknames un meeting datumi, kuros spēlētāji ierodas pārstāvēt savu komandu.

|  |
| --- |
|  |
| 2.14.attēls. spēlētāji, kuri ierodas uz noteiktiem meetings |

SELECT T\_VARDS, T\_UZVARDS, K\_NOSAUKUMS, TAKTIKAS\_NOS

FROM TRENERIS, KOMANDA, TAKTIKA, IZMANTO

WHERE TRENERIS.KOMANDA\_ID=KOMANDA.KOMANDA\_ID AND TRENERIS.TRENERIS\_ID=IZMANTO.TRENERIS\_ID AND IZMANTO.TAKTIKA\_ID=TAKTIKA.TAKTIKA\_ID;

Ar šo vaicājumu tiek iegūti treneru vārdi, uzvārdi, komandas nosaukumus, kuru tie pārstāv, un taktikas nosaukums, kuru tie izmanto (skat.2.15.att.).

|  |
| --- |
|  |
| 2.15.attēls. Treneri, komandas ko tie pārstāv, un taktika, ko izmanto |

SELECT K\_NOSAUKUMS, I\_TIME

FROM KOMANDA, INTERVIEWS

WHERE KOMANDA.KOMANDA\_ID=INTERVIEWS.TEAM\_ID;

Ar šo vaicājumu tiek iegūti komandu nosaukumi un interviju laiki, kad attiecīgajai komandai ir paredzēta intervija (skat.2.16.att).

|  |
| --- |
|  |
| 2.16.attēls. Komandu nosaukumi un to interviju laiki |

## From ar ārējo savienošanu

Tā kā Access datu bāže nav FULL OUTER JOIN, tad tā rezultātu var iegūt apvienojot LEFT un RIGHT join.

SELECT KOMANDA.KOMANDA\_ID, KOMANDA.K\_NOSAUKUMS, SPONSORS\_ID

FROM KOMANDA

LEFT JOIN SPONSORS

ON KOMANDA.KOMANDA\_ID=SPONSORS.KOMANDA\_ID

UNION

SELECT KOMANDA.KOMANDA\_ID, KOMANDA.K\_NOSAUKUMS, SPONSORS\_ID

FROM KOMANDA

RIGHT JOIN SPONSORS

ON KOMANDA.KOMANDA\_ID=SPONSORS.KOMANDA\_ID;

|  |
| --- |
|  |
| 2.17.attēls. Visas komandas ar to visiem sponsoriem. |

SELECT KOMANDA.KOMANDA\_ID, KOMANDA.K\_NOSAUKUMS, SPONSORS\_ID

FROM KOMANDA

RIGHT JOIN SPONSORS

ON KOMANDA.KOMANDA\_ID=SPONSORS.KOMANDA\_ID;

|  |
| --- |
|  |
| 2.18.attēls. Visi sponsori un komandas, ko tie atbalsta |

## Apakšvaicājumi Where vai Having daļā

SELECT COUNT (GAMES.WINNER), KOMANDA.KOMANDA\_ID

FROM KOMANDA, GAMES

WHERE KOMANDA.KOMANDA\_ID=GAMES.WINNER AND KOMANDA.KOMANDA\_ID<>(SELECT KOMANDA.KOMANDA\_ID FROM KOMANDA WHERE KOMANDA\_ID=4)

GROUP BY KOMANDA.KOMANDA\_ID;

|  |
| --- |
|  |
| 2.19.attēls. apakšvaicājums WHERE daļā |

SELECT COUNT (PIEDALAS.KOMANDA\_ID) AS TURNIRUSKAITS, KOMANDA.K\_NOSAUKUMS

FROM KOMANDA, PIEDALAS

WHERE KOMANDA.KOMANDA\_ID=PIEDALAS.KOMANDA\_ID AND KOMANDA.KOMANDA\_ID<>(SELECT KOMANDA.KOMANDA\_ID FROM KOMANDA WHERE KOMANDA\_ID=1)

GROUP BY KOMANDA.K\_NOSAUKUMS;

|  |
| --- |
|  |
| 2.20.attēls.apakšvaicājuma WHERE daļā reuzltāts |

## Apakšvaicājumi Select vai From daļā

SELECT KOMANDA.K\_NOSAUKUMS, (SELECT COUNT (SPONSORS.SPONSORS\_ID)

FROM FINANSIALAIS\_SPONSORS, SPONSORS

WHERE KOMANDA.KOMANDA\_ID=SPONSORS.KOMANDA\_ID AND SPONSORS.SPONSORS\_ID=FINANSIALAIS\_SPONSORS.FSPONSORS\_ID

GROUP BY SPONSORS.KOMANDA\_ID) AS NUMBEROFSPONSORS

FROM KOMANDA;

|  |
| --- |
|  |
| 2.21.attēls. Komandas un to finansiālo sponsoru skaits |

SELECT KOMANDA.KOMANDA\_ID, KOMANDA.K\_NOSAUKUMS, (SELECT COUNT (GAMES.WINNER)

FROM GAMES

WHERE KOMANDA.KOMANDA\_ID=GAMES.WINNER

GROUP BY KOMANDA.KOMANDA\_ID) AS GAMESWINNED

FROM KOMANDA

ORDER BY KOMANDA.KOMANDA\_ID;

|  |
| --- |
|  |
| 2.22.attēls. Komandas un to uzvarēto spēļu skaits |

## Having pareiza izmantošana

SELECT COUNT (SPELETAJS\_ID) AS SPELETAJIKOMANDA,KOMANDA\_ID

FROM SPELETAJS

GROUP BY KOMANDA\_ID

HAVING COUNT (SPELETAJS\_ID)>0;

|  |
| --- |
|  |
| 2.23.attēls. Komandas ar eksistējošu spēlētāju skaitu |

SELECT COUNT (PIEDALAS.KOMANDA\_ID) AS TURNIRUSKAITS, KOMANDA.K\_NOSAUKUMS

FROM KOMANDA, PIEDALAS

WHERE KOMANDA.KOMANDA\_ID=PIEDALAS.KOMANDA\_ID AND KOMANDA.KOMANDA\_ID<>(SELECT KOMANDA.KOMANDA\_ID FROM KOMANDA WHERE KOMANDA\_ID=1)

GROUP BY KOMANDA.K\_NOSAUKUMS

HAVING COUNT (PIEDALAS.KOMANDA\_ID)>1;

|  |
| --- |
|  |
| 2.24.attēls. Komandas, kas piedalās vairāk kā vienā turnīrā |

## Datu sakārtošana

SELECT KOMANDA.KOMANDA\_ID, KOMANDA.K\_NOSAUKUMS, (SELECT COUNT (GAMES.WINNER)

FROM GAMES

WHERE KOMANDA.KOMANDA\_ID=GAMES.WINNER

GROUP BY KOMANDA.KOMANDA\_ID) AS GAMESWINNED

FROM KOMANDA

ORDER BY KOMANDA.KOMANDA\_ID;

|  |
| --- |
|  |
| 2.25.attēls. Komandas un to uzvarēto spēļu skaits sakārtots pēc komandas ID |

SELECT KOMANDA.KOMANDA\_ID,KOMANDA.K\_NOSAUKUMS, (SELECT COUNT (GAMES.WINNER)

FROM GAMES

WHERE KOMANDA.KOMANDA\_ID=GAMES.WINNER

GROUP BY KOMANDA.KOMANDA\_ID) AS GAMESWINNED

FROM KOMANDA

ORDER BY KOMANDA.K\_NOSAUKUMS;

|  |
| --- |
|  |
| 2.26.attēls. Komandas un to uzvarēto spēļu skaits, sakārtos pēc komandu nosaukuma |

## Funkciju izmantošana (piemēram Sum(), Count()) dažādās vaicājuma sadaļās

SELECT DISTINCT COUNT (SPELETAJS.KOMANDA\_ID), SPELETAJS.KOMANDA\_ID

FROM SPELETAJS

GROUP BY SPELETAJS.KOMANDA\_ID;

|  |
| --- |
|  |
| 2.27.attēls. Spēlētāju skaits komandā, balstoties uz datiem tabulā SPELETAJS |

SELECT SUM(SPELETAJS.KILLS), SPELETAJS.KOMANDA\_ID

FROM SPELETAJS

GROUP BY SPELETAJS.KOMANDA\_ID;

|  |
| --- |
|  |
| 2.28.attēls. Kopējā KILL summa starp komandas spēlētājiem |

## Predikātu izmantošana (LIKE ar “?”. “\*”, &; ANY; ALL; IN; Distinct)

SELECT DISTINCT K\_NOSAUKUMS

FROM PIEDALAS

ORDER BY K\_NOSAUKUMS;

|  |
| --- |
|  |
| 2.29.attēls.Komandu nosaukumi, kuras piedalās turnīros |

SELECT \*

FROM SPELETAJS

WHERE KILLS BETWEEN 10 AND 50;

|  |
| --- |
|  |
| 2.30.Speletāji ar KILLS starp 10 un 50 |

SELECT SPELETAJS\_ID, SP\_NICKNAME

FROM SPELETAJS

WHERE SP\_NICKNAME LIKE '\*A\*';

|  |
| --- |
|  |
| 2.31.attēls. Spēlētāji, kuru nickname satur burtu ‘a’ |

## Aritmētisko operācioju izmantošana

SELECT SPELETAJS\_ID, SP\_NICKNAME, K\_NOSAUKUMS, (KILLS+DEATHS+ASSISTS) AS TOTAL\_KDA

FROM SPELETAJS

WHERE (KILLS+DEATHS+ASSISTS)>20;

|  |
| --- |
|  |
| 2.32.attēls. Labāko spēlētāju absolūtais KDA |

## Rezultātu apvienošana UNION query

SELECT KOMANDA.KOMANDA\_ID, KOMANDA.K\_NOSAUKUMS, SPONSORS\_ID

FROM KOMANDA

LEFT JOIN SPONSORS

ON KOMANDA.KOMANDA\_ID=SPONSORS.KOMANDA\_ID

UNION

SELECT KOMANDA.KOMANDA\_ID, KOMANDA.K\_NOSAUKUMS, SPONSORS\_ID

FROM KOMANDA

RIGHT JOIN SPONSORS

ON KOMANDA.KOMANDA\_ID=SPONSORS.KOMANDA\_ID;

|  |
| --- |
|  |
| 2.33.attēls. Visas komandas ar to visiem sponsoriem. |

## Šķērsgriezuma tabulas izveide ar *transform/pivot* konstrukcijas izmantošanu (Crosstab Query)

TRANSFORM Count(Speletajs.[Speletajs\_ID]) AS CountOfSpeletajs\_ID

SELECT Speletajs.[Komanda\_ID], Count(Speletajs.[Speletajs\_ID]) AS [Total Of Speletajs\_ID]

FROM Speletajs

GROUP BY Speletajs.[Komanda\_ID]

PIVOT Speletajs.[Sp\_nickname];

|  |
| --- |
|  |
| 2.34.attēls. Pārveidota tabula |

# Secinājumi

Darba mērķis bija iepazīt un pielietot SQL valodas konstrukcijas. Lai sasniegtu mērķi, tika realizēti dažādi vaicājumi. Arī apakšvaicājumi tika pielietoti, kas prasa detalizētu uzmanību sintaksei, bet sniedz iespēju iegūt prasīto informāciju vāirāk kā vienā veidā. Tomēr, izmantojot daudz apakšvaicājumu vienā vaicājumā, SQL vaicājuma pārkatāmība strauji sarūk.

Kopsummā katrā nospraustajā darba uzdevuma iedaļā tika realizēti viens vai vairāki vaicājumi, izmantojot attiecīgo konstrukciju vai operandus, kas noteikti ir prasījis laika, bet ir ļāvis apgūt SQL vaivājumu veidošanu.