

Uujate tulemuste andmebaas

Sisukord

Sisukord.....	1
Mõisted	2
Valdkonna kirjeldus.....	3
Andmebaasi kirjeldus	4
Graafiline mudel	7
Mudeli relatsiooniline kuju graafilise mudeli koostamise programmist	7
Reaalne andmebaas.....	9
Andmebaasi loomise käsud:	9
Andmete sisestamise käsud	10
Tabelite skeem valmis andmebaasist.....	11
Päringud andmebaasi	14

Mõisted

Alad - ujumisstiil ja distant, mida ujuja võistlusel läbib.

Basseini pikkus - võistlemiseks kasutatakse 25 m pikkust basseini või 50 m pikkust basseini.

Distant - Etteantud arv meetreid, mille ujuja peab ühe ala jooksul läbima. Võisteldakse 50 m, 100 m, 200 m, 400 m, 800 m ja 1500 m pikkustel distantidel.

EOK – Eesti Olümpiakomitee

Klubi - spordiklubi, mille ridades ujuja treenib ja võistleb.

Klubilühend - igale ujumisega tegelevale spordiklubile on tema nime järgi antud lühend. Nt Tartu Ujumisklubi = TUK.

Pädevuskategooria - Igal treeneril on tema kogemusele ja staažile Eesti Olümpiakomitee poolt antud vastav pädevustunnistus, mis tõendab, millise kategooria treener ta on.

Treener - Klubi juhtfiguur, kes õpetab ja treenid ujujat.

Ujuja - sportlane, kes tegeleb ujumise spordialaga

Ujula - koht, kus ujuja treenib ja võistleb

Ujumisstiil - on olemas 4 erinevat ujumisstiili: liblikas, selili, rinnuli ja vabalt.

Võistlused - korraldatud ujumisvaldkonna inimeste kogunemised, kus ujujad ujuvad võidu aja peale.

Valdkonna kirjeldus

Tegemist on ujumise spordiala andmebaasiga. Andmebaas on mõeldud kõigile võistlevatele ujujatele, treeneritele, ujumisliidule ja ujumissõpradele. Andmebaasis hoiustatakse kõikide ujujate kõikidel võistlustel saadud kõiki tulemusi, andmeid kõigi ujumisklubide ja nende treenerite kohta.

Ujumine on väga populaarne spordiala, mida praktiseerivad paljud. Selleks, et seda spordiala kaardistada, peab kõik seosed kirja panema ja kõik tekkivad andmed salvestama. Nii on ujujatel, treeneritel ja ka teistel ujumissõpradel võimalik omada pidevat ülevaadet kõigist tulemustest ja võistlustest.

Andmebaasi kasutajateks on võistluste korraldajad ja ujumisliidu inimesed, kellel on vaja silma peal hoida ujujate tulemustel ning teha kokkuvõtteid toimuvast, aga ka ujujad ise, kes soovivad saada ülevaadet oma tulemustest. Andmebaas annab ülevaate üle-eestilisest olukorrast: kes on kõige paremad, milliste treenerite õpilased on tulemuslikud, millised on võimalused ujulates jne.

Andmebaas annab võimaluse kõigile osapooltele andmeid vaadata jooksvalt, ilma, et ole vajadus hakata infot küsima võistluse korraldajatelt või treeneritelt e-mailide v muul teel. Samuti andmete ühte kohta koondamine aitab lihtsamalt teha kokkuvõtteid ja saada ülevaade.

Andmebaasi kirjeldus

Andmebaasis on ujujate parimate tulemuste tabel. Andmebaas koosneb seitsmest olemist: Ujuja, Klubid, Tulemused, Alad, Basseinid, Treenerid:

Ujujad (IDkood, Eesnimi, Perenimi, Klubi_luhend, Sugu, Sünniaasta)

Relatsioon on kolmandal normaalkujul, kuna atribuudid Eesnimi, Perenimi, Klubi_luhend, Sugu ja Sünniaasta on kõik sõltuvuses relatsiooni primaarvõtmest IDkood. See tähendab, et kõik atribuudid v.a IDkood on vastastikku sõltumatud ning saavutavad tähenduse ainult koos atribuudiga IDkood ehk on sellest sõltuvad. Nt atribuudid Eesnimi ja Perenimi on üksteise suhtes sõltumatud ning saavad kindla tähenduse, kui need asetada sõltuvusse IDkood-iga ehk tabelis võib esineda sama eesnime, perenime ja isegi sünniaastaga isikuid, kuid isikut saab tuvastada vaid IDkoodi abil.

Tulemused (Id, Koht, Aeg, Ala, Ujuja_IDkood, Võistluse_id)

Relatsioon on kolmandal normaalkujul kuna kolm atribuuti on välisvõtmed ning ülejäänud mitte-võtme atribuudid Koht ja Aeg ei ole omavahel sõltuvuses, kuid on sõltuvuses primaarvõtmest Id.

Alad (Alanr, Distsants_stiil)

Relatsioon Alad on kahe atribuudiga relatsioon, millest üks atribuut on primaarvõti ning teine veerg on sõltuvuses esimesest. Järelikult on relatsioon kolmandal normaalkujul.

Klubid (Klubi_luhendid, Klubi_nimi)

Relatsioon on kolmandal normaalkujul, kuna relatsioonis on kaks atribuuti, millest üks on primaarvõti ja teine mitte ning see teine on sõltuvuses esimesest ehk primaarvõtme atribuudist.

Võistlused (Id, Nimi, Toimumisaeg, Bassein)

Relatsioon Võistlused on kolmandal normaalkujul, kuna atribuudid ei ole omavahel sõltuvuses, st et sama nimega võistlus saab samas basseinis toimuda erinevatel aegadel, kuid iga erinev nime, basseini või toimumisaja kombinatsioon on sõltuvuses Id-ga.

Basseinid (Nimi, Basseini_pikkus, Radade_arv, Asukoht)

Relatsioon on kolmandas normaalkujus, kuna selle mitte-võtme atribuudid omaette või seotuna teiste mitte-võtme atribuutidega ei tekita mingisuguseid sõltuvusi ja on n-ö mitte midagi ütlevad, kuid on kõik sõltuvuses primaarvõtmest Nimi, st primaarvõti annab teistele atribuutidele tähenduse. Nt Basseini asukoht ei määra ära mitu rada selles basseinis on (sest ühes kohas võib olla mitu basseini), kuid basseini nime alusel võib öelda, basseini pikkuse, radade arvu ja asukoha.

Treenerid (Id, Eesnimi, Perenimi, Pädevus, Klubi_luhend)

Relatsiooni Treenerid puhul kehtiv sarnane põhjendus, miks see on kolmandal normaalkujul, nagu relatsiooni Ujujad puhul ehk kõik mitte-võtme atribuudid ei ole üksteisega sõltuvuses, kuid kõik on sõltuvuses primaarvõtmega Id.

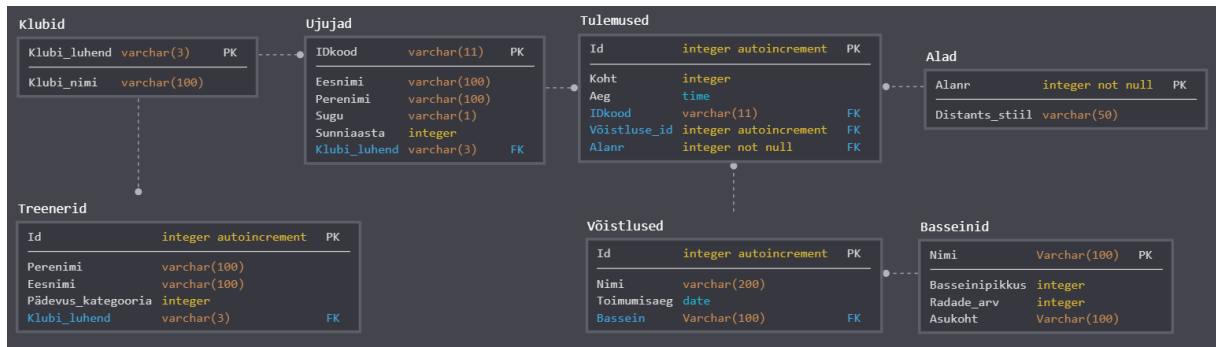
Olem "Ujujad" tunnusteks on tema ID (primaarne võti), Eesnimi, Perenimi, Klubi_luhend, Sugu ja Sünniaasta. Olem "Tulemused" näitab, millised on ujuja kõige paremad tulemused ehk isiklikud rekordid mingil alal ning igale tehtud tulemusele pannakse külge unikaalne ID, mis on ka olemi primaarvõti. Olem "Alad" näitab, milline on ujumistiil ja distants: tabelis on ühes veerus välja toodud kõik võimalikud EOK poolt kinnitatud distantse ja stiili kombinatsioonid ning teises veerus, mis on selle olemi primaarvõtme veerg, on igale distantse ja stiili kombinatsioonile antud unikaalne number. Olem "Klubid" tunnuseid on ainult kaks – Klubi_luhend (primaarvõti) ja Klubi_nimi. Olem "Võistlused" näitab, kus ja millal võisteldi ning selle olemi primaarvõti on Id. Olem "Basseinid" näitab milline on võistlusbasseini pikkus ja mitu rada seal on ning selle olemi primaarvõti on tunnus „Nimi“. Olem „Treenerid“ näitab treeneri ees- ja perenime, tema pädevust ja kuhu klubisse ta kuulub ning igale treeneril on antud personaalne Id, mis on ühtlasi ka olemi primaarvõti.

Ühte klubisse võib kuuluda mitu ujujat, kuid iga ujuja kuulub vaid ühte klubisse. Olem "Ujujad" on olemiga "Klubid" ühendatud välisvõtme Klubi_luhend kaudu (Klubid 1:N Ujujad seos). Tulemusi võib olla igal ujujal mitu, kuid ühel ujujal on ühel alal ainult üks parim tulemus (Tulemused N:1 Ujuja seos). Samas ühe alaga võib olla seotud ka mitme erineva ujuja tulemused (Tulemused N:M Alad seos). Igal klubil võib olla mitu treenerit, kuid üks treener võib olla seotud vaid ühe klubiga (Treenerid N:1 Klubid seos). Olem "Treenerid" on ühendatud olemiga "Klubid" välisvõtme "Klubi_luhend" kaudu. Võistlused võivad toimuda erinevates basseinides aga ühes basseinis saab korraga toimuda ainult üks võistlus (Bassein 1:1 Võistlused seos), olem "Võistlused" on ühendatud olemiga "Basseinid" välisvõtme „Bassein“ kaudu. Olem "Tulemused"

on ühendatud olemiga “Võistlused” välisvõtme „Võistluse_id“ abil. Olem „Alad“ on primaarvõtme „Alanr“ abil ühendatud olemi „Tulemused“ atribuudiga “Alanr”.

Konkreetne andmebaas on vaid näidis. See tähendab, et ujujate ja klubi nimed ning tulemused mingil alal ei ole tõesed andmed, vaid juhuslikult väljamõeldud.

Graafiline mudel



Joonis 1 Graafiline mudel

Mudeli relatsiooniline kuju graafilise mudeli koostamise programmist

```
CREATE TABLE 'Ujujad'
('IDkood'    varchar(11) NOT NULL ,
'Eesnimi'   varchar(100) NOT NULL ,
'Perenimi'  varchar(100) NOT NULL ,
'Sugu'      varchar(1) NOT NULL ,
'Sunniaasta' integer NOT NULL ,
'Klubid_luhend' varchar(3) NOT NULL ,
PRIMARY KEY ('IDkood'),
CONSTRAINT 'fk_Ujujad_Klubid' FOREIGN KEY ('Klubid_luhend') REFERENCES 'Klubid'
('Klubid_luhend'));
```

```
CREATE TABLE 'Tulemused'
('Id' integer autoincrement NOT NULL ,
'Koht' integer NOT NULL ,
'Aeg' time NOT NULL ,
'Alanr' varchar(50) NULL ,
'Võistluse_id' integer NOT NULL ,
'IDkood' varchar(11) NOT NULL ,
'Võistluse_id_1' integer autoincrement NOT NULL ,
PRIMARY KEY ('Id'),
CONSTRAINT 'FK_49' FOREIGN KEY ('IDkood') REFERENCES 'Ujujad' ('IDkood'),
CONSTRAINT 'FK_55' FOREIGN KEY ('Võistluse_id_1') REFERENCES 'Võistlused' ('Id')
CONSTRAINT 'fk_Alad_Tulemused' FOREIGN KEY (Alanr) REFERENCES 'Alad' ('Alanr'));
```

```
CREATE TABLE 'Alad'
('Alanr' integer NOT NULL,
'Distantstiiil' varchar(50) NOT NULL ,
PRIMARY KEY ('Alanr') );
```

```
CREATE TABLE 'Klubid'
('Klubi_luhend' varchar(3) NOT NULL ,
'Klubi_nimi' varchar(100) NOT NULL ,
PRIMARY KEY ('Klubi_luhend'));
```

```
CREATE TABLE 'Võistlused'
('Id' integer autoincrement NOT NULL ,
'Nimi' NOT NULL ,
'Toimumisaeg' date NOT NULL ,
'Bassein' varchar(100) NOT NULL ,
PRIMARY KEY ('Id'),
CONSTRAINT 'fl_Võistlused_Basseinid' FOREIGN KEY ('Bassein') REFERENCES 'Basseinid'
('Nimi'));
```

```
CREATE TABLE 'Basseinid'
('Nimi' varchar(100) NOT NULL ,
'Basseinipikkus' integer NOT NULL ,
'Radade_arv' integer NOT NULL ,
'Asukoht' varchar(100) NOT NULL ,
PRIMARY KEY ('Nimi'));
```

```
CREATE TABLE 'Treenerid'
('Perenimi' varchar(100) NOT NULL ,
'Id' integer autoincrement NOT NULL ,
'Eesnimi' varchar(100) NOT NULL ,
'Pädevus_kategooria' integer NOT NULL ,
'Klubi_luhend' varchar(3) NOT NULL ,
PRIMARY KEY ('Id'),
CONSTRAINT 'fk_Treenerid_klubid' FOREIGN KEY ('Klubi_luhend') REFERENCES 'Klubid'
('Klubi_luhend'));
```

Käskude loomiseks kasutasime SQL DBM programmi. Saadud käsklustes tegime järgmised muudatused:

- asendasime märgi ` märgiga `
- Panime seosetele oma nimed, sest programm genereeris neile automaatselt nimetused, millest oli keerulisem aru saada.

Reaalne andmebaas

Andmebaasi realiseerisime andmebaasisüsteemis Sybase SQL Anywhere.

Andmebaasi loomise käsud:

```
CREATE TABLE Ujujad
( IDkood    varchar(11) NOT NULL ,
  Eesnimi   varchar(100) NOT NULL ,
  Perenimi  varchar(100) NOT NULL ,
  Sugu      varchar(1) NOT NULL ,
  Sunniaasta integer NOT NULL ,
  Klubi_luhend varchar(3) NOT NULL ,
  PRIMARY KEY (IDkood));
```

```
ALTER TABLE Ujujad ADD CONSTRAINT fk_Ujujad_Klubid
FOREIGN KEY (Klubi_luhend)
REFERENCES Klubid (Klubi_luhend);
```

```
CREATE TABLE Tulemused
( Id integer not null default autoincrement primary key ,
  Koht integer NOT NULL ,
  Aeg time NOT NULL
  Alanr varchar(50));
```

```
ALTER TABLE Tulemused ADD CONSTRAINT fk_Ujuja_tulemused
FOREIGN KEY (IDkood)
REFERENCES Ujujad (IDkood);
```

```
ALTER TABLE Tulemused ADD CONSTRAINT fk_Võistlused_Tulemused
FOREIGN KEY (Võistluse_id) REFERENCES Võistlused (Id);
```

```
CREATE TABLE Alad
(Alanr integer not null primary key,
Distsants_stiil varchar(50) NOT NULL);
```

```
ALTER TABLE Tulemused ADD CONSTRAINT Tulemus_Ala
FOREIGN KEY (Alanr)
REFERENCES Alad(Alanr);
```

```
CREATE TABLE Klubid
( Klubi_luhend varchar(3) NOT NULL ,
  Klubi_nimi varchar(100) NOT NULL ,
  PRIMARY KEY (Klubi_luhend));
```

```
CREATE TABLE Võistlused
```

```
( Id integer not null default autoincrement primary key,  
  Nimi varchar(100) NOT NULL ,  
  Toimumisaeg date NOT NULL ,  
  Bassein varchar(100) NOT NULL);
```

```
ALTER TABLE Võistlused ADD CONSTRAINT fk_Võistlused_Basseinid  
FOREIGN KEY (Bassein) REFERENCES Basseinid (Nimi);
```

```
CREATE TABLE Basseinid
```

```
( Nimi varchar(100) NOT NULL ,  
  Basseinipikkus integer NOT NULL ,  
  Radade_arv integer NOT NULL ,  
  Asukoht varchar(100) NOT NULL ,  
  PRIMARY KEY (Nimi));
```

```
CREATE TABLE Treenerid
```

```
( Id integer not null default autoincrement primary key,  
  Perenimi varchar(100) NOT NULL ,  
  Eesnimi varchar(100) NOT NULL ,  
  Pädevus_kategooria integer NOT NULL);
```

```
ALTER TABLE Treenerid ADD CONSTRAINT fk_Treenerid_klubid  
FOREIGN KEY (Klubi_luhend)  
REFERENCES Klubid (Klubi_luhend);
```

Andmete sisestamise käsud

```
INPUT INTO alad from 'C:\Users\kaari\Documents\Ühiskonna ja infoprotsesside analüüs\1.  
semester\Andmebaasid\OSP\Alad.txt'  
FORMAT ASCII DELIMITED BY '\x09'(Alanr, Distsants_stiil);
```

```
INPUT INTO Basseinid from 'C:\Users\kaari\Documents\Ühiskonna ja infoprotsesside analüüs\1.  
semester\Andmebaasid\OSP\Basseinid.txt'  
FORMAT ASCII DELIMITED BY '\x09'(Nimi, Basseinipikkus, Radade_arv, Asukoht);
```

```
INPUT INTO Klubid from 'C:\Users\kaari\Documents\Ühiskonna ja infoprotsesside analüüs\1.  
semester\Andmebaasid\OSP\Klubid.txt'  
FORMAT ASCII DELIMITED BY '\x09'(Klubi_luhend, Klubi_nimi);
```

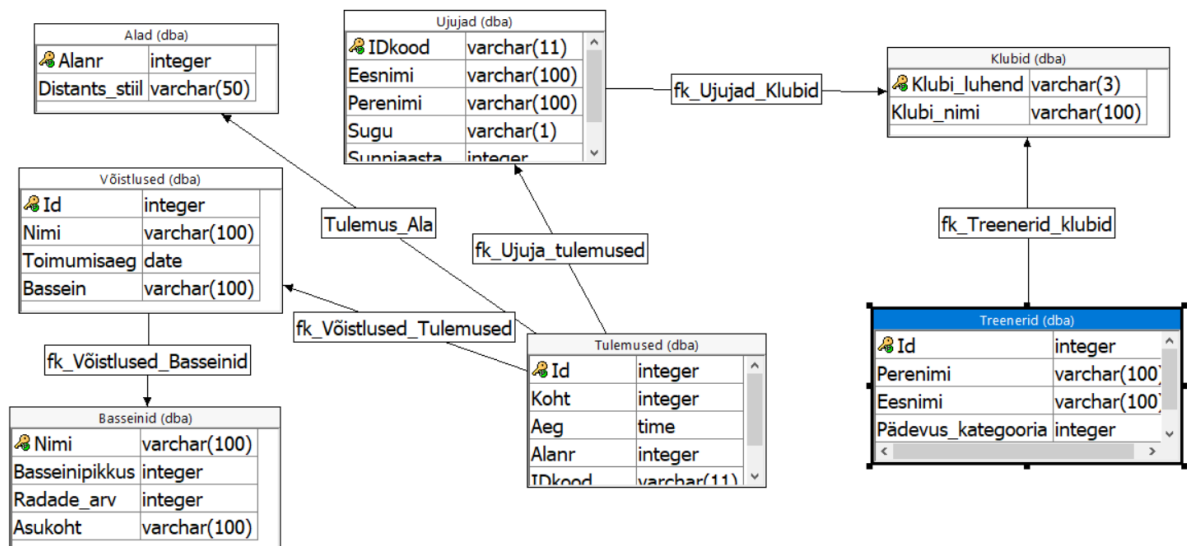
```
INPUT INTO Treenerid from 'C:\Users\kaari\Documents\Ühiskonna ja infoprotsesside analüüs\1.  
semester\Andmebaasid\OSP\Treenerid.txt'  
FORMAT ASCII DELIMITED BY '\x09'(Id, Perenimi, Eesnimi, Pädevus_kategooria,  
Klubi_luhend);
```

INPUT INTO Tulemused from 'C:\Users\kaari\Documents\Ühiskonna ja infoprotsesside analüüs\1. semester\Andmebaasid\OSP\Tulemused.txt'
 FORMAT ASCII DELIMITED BY '\x09';

INPUT INTO Ujujad from 'C:\Users\kaari\Documents\Ühiskonna ja infoprotsesside analüüs\1. semester\Andmebaasid\OSP\Ujujad.txt'
 FORMAT ASCII DELIMITED BY '\x09';

INPUT INTO Võistlused from 'C:\Users\kaari\Documents\Ühiskonna ja infoprotsesside analüüs\1. semester\Andmebaasid\OSP\Võistlused.txt'
 FORMAT ASCII DELIMITED BY '\x09';

Tabelite skeem valmis andmebaasist



Joonis 2 Tabelite skeem valmis andmebaasist

Andmebaasi imporditud andmed

Alad:

1	50 m liblikat
5	100 m liblikat
10	200 m liblikat
2	50 m selili
6	100 m selili
11	200 m selili
3	50 m rinnuli
7	100 m rinnuli
12	200 m rinnuli
4	50 m vabalt
8	100 m vabalt
13	200 m vabalt
15	400 m vabalt
17	800 m vabalt
18	1500 m vabalt
9	100 m kompleksi
14	200 m kompleksi
16	400 m kompleksi

Basseinid:

Nõmme Ujula sportbassein	25	6	Tallinn
Aura Veekeskuse võistlusbassein	50	10	Tartu
Võru Ujula suur bassein	25	6	Võru
Pirita Marina Spordikeskus sportbassein	25	6	Tallinn

Klubid:

PSK	Pärnu Spordikool
TUK	Tartu Ujumisklubi

TOP TOPi Ujumisklubi

VUK Võru Ujumisklubi

Treenerid:

101	Kalju Pool	4	VUK
102	Liisa Tiil	5	TUK
103	Sergei Teplenkov	6	TOP
104	Alar Nõmm	3	PSK

Tulemused:

33	2	04:32:32	15	49804253672	1115
34	1	00:27:34	1	39702304256	1114
35	6	01:15:19	6	49504194352	1112
36	4	02:28:33	12	39410234278	1111

Ujujad:

39702304256	Martin Roos	M	1997	PSK
49804253672	Katrin Viirpalu	N	1998	VUK
39410234278	Kaspar Tuus	M	1994	TOP
49504194352	Tuuli Tuim	N	1995	TUK

Võistlused:

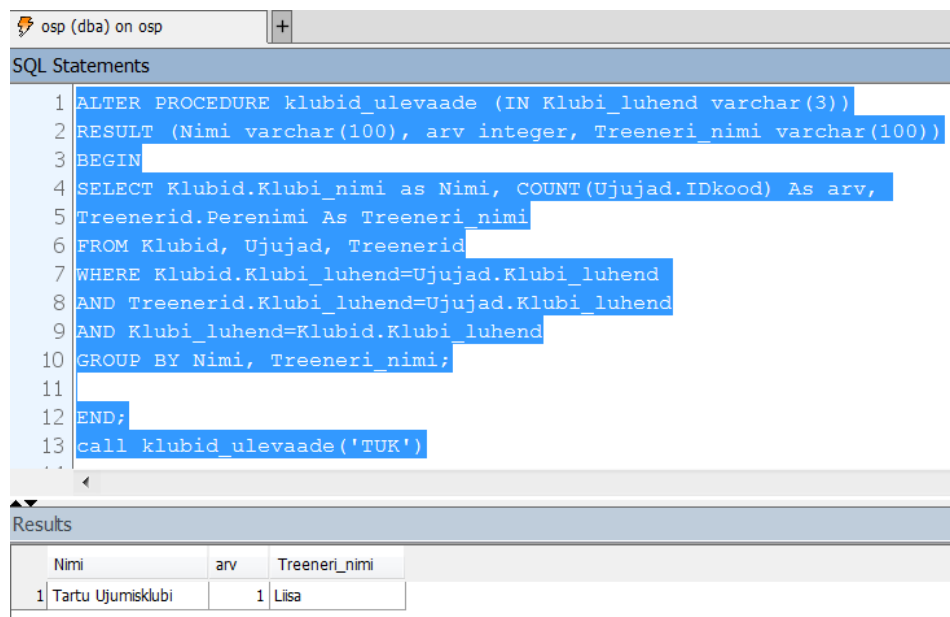
1111	Nordic Swimming Championships	2019-10-16	Pirita	Marina	Spordikeskus sportbassein
1112	Briis Open	2019-08-22	Nõmme Ujula		sportbassein
1113	EUL Noortesari	2018-05-23	Aura	Veekeskuse	võistlusbassein
1114	Laste Karikas	2019-03-24	Võru Ujula		suur bassein
1115	EUL Noortesari	2018-10-16	Võru Ujula		suur bassein

Päringud andmebaasi

1. Luua protseduur klubid_ulevaade, mis väljastab klubi nime, sealsete ujujate arvu ja treeneri nime.

```
CREATE PROCEDURE klubid_ulevaade (IN Klubi_luhend varchar(3))
RESULT (Nimi varchar(100), arv integer, Treeneri_nimi varchar(100))
BEGIN
SELECT Klubid.Klubi_nimi as Nimi, COUNT(Ujujad.IDkood) As arv,
Treenerid.Perenimi As Treeneri_nimi
FROM Klubid, Ujujad, Treenerid
WHERE Klubid.Klubi_luhend=Ujujad.Klubi_luhend
AND Treenerid.Klubi_luhend=Ujujad.Klubi_luhend
AND Klubid_luhend=Klubid.Klubi_luhend
GROUP BY Nimi, Treeneri_nimi;
END;
```

CALL klubid_ulevaade('TUK')



The screenshot shows a database management tool interface. At the top, there's a tab labeled 'osp (dba) on osp'. Below it, the 'SQL Statements' section displays the following SQL code:

```
1 ALTER PROCEDURE klubid_ulevaade (IN Klubi_luhend varchar(3))
2 RESULT (Nimi varchar(100), arv integer, Treeneri_nimi varchar(100))
3 BEGIN
4 SELECT Klubid.Klubi_nimi as Nimi, COUNT(Ujujad.IDkood) As arv,
5 Treenerid.Perenimi As Treeneri_nimi
6 FROM Klubid, Ujujad, Treenerid
7 WHERE Klubid.Klubi_luhend=Ujujad.Klubi_luhend
8 AND Treenerid.Klubi_luhend=Ujujad.Klubi_luhend
9 AND Klubid_luhend=Klubid.Klubi_luhend
10 GROUP BY Nimi, Treeneri_nimi;
11
12 END;
13 call klubid_ulevaade('TUK')
```

Below the SQL statements, the 'Results' section shows a table with the following data:

	Nimi	arv	Treeneri_nimi
1	Tartu Ujumisklubi	1	Liisa

Joonis 3 Väljund 1. päringule

2. Leida, millised on Katrin Viirpalu (IDkoodiga: 49804253672) ajad eri võistlustel koos alanimetusega. (protseduur peab toimima ka teiste ujujate tulemuste vaatamiseks)

```
CREATE PROCEDURE tulemused_ujuja (IN IDkood varchar(11))
RESULT (ajad time, alad varchar(50), võistlused varchar(100))
BEGIN
SELECT Tulemused.Aeg as ajad, Alad.Distants_stiil as alad,
      Võistlused.nimi as võistlused
FROM Tulemused, Alad, Ujujad, Võistlused
WHERE Tulemused.IDkood = Ujujad.IDkood
AND Tulemused.Võistluse_id = Võistlused.Id
AND Tulemused.Alanr = Alad.Alanr AND Ujujad.IDkood=IDkood;
END;
```

```
CALL tulemused_ujuja('49804253672');
```

SQL Statements			
1	CREATE	PROCEDURE	tulemused_ujuja (IN IDkood varchar(11))
2	RESULT	(ajad time, alad varchar(50), võistlused varchar(100))	
3	BEGIN		
4	SELECT	Tulemused.Aeg as ajad, Alad.Distants_stiil as alad,	
5		Võistlused.nimi as võistlused	
6	FROM	Tulemused, Alad, Ujujad, Võistlused	
7	WHERE	Tulemused.IDkood = Ujujad.IDkood	
8	AND	Tulemused.Võistluse_id = Võistlused.Id	
9	AND	Tulemused.Alanr = Alad.Alanr AND Ujujad.IDkood=IDkood	
10	;		
11	END;		
12	call	tulemused_ujuja('49804253672');	

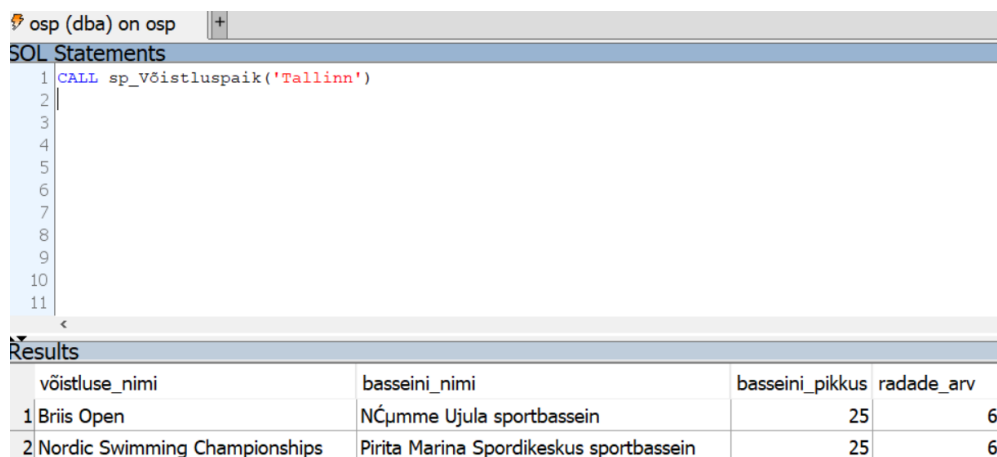
Results			
	ajad	alad	võistlused
1	4:32	400 m vabalt	EUL Noortesari

Joonis 4 Väljund 2. päringule

3. Leida kasutaja poolt sisestatud asukohas/linnas toimunud võistluste nimed ning tuua välja, mis nimega, kui pikas ja mitme rajaga basseinis võistlus peeti.

```
CREATE PROCEDURE sp_Võistluspaik (in Asula varchar (100))
RESULT (võistluse_nimi varchar (100),
        basseini_nimi varchar (100),
        basseini_pikkus integer,
        radade_arv integer)
BEGIN
SELECT Võistlused.Nimi AS võistluse_nimi,
        Basseinid.Nimi AS basseini_nimi,
        Basseinid.Basseinipikkus AS basseini_pikkus,
        Basseinid.Radade_arv AS radade_arv
FROM basseinid, võistlused
WHERE Asula = Basseinid.Asukoht AND
Basseinid.nimi = Võistlused.Bassein
END;
```

```
CALL sp_Võistluspaik('Tallinn')
```



The screenshot shows a database query tool interface. At the top, there's a tab labeled 'osp (dba) on osp'. Below it, the 'SQL Statements' pane contains the following SQL code:

```
1 CALL sp_Võistluspaik('Tallinn')
2
3
4
5
6
7
8
9
10
11
```

Below the SQL statements, the 'Results' pane displays a table with the following data:

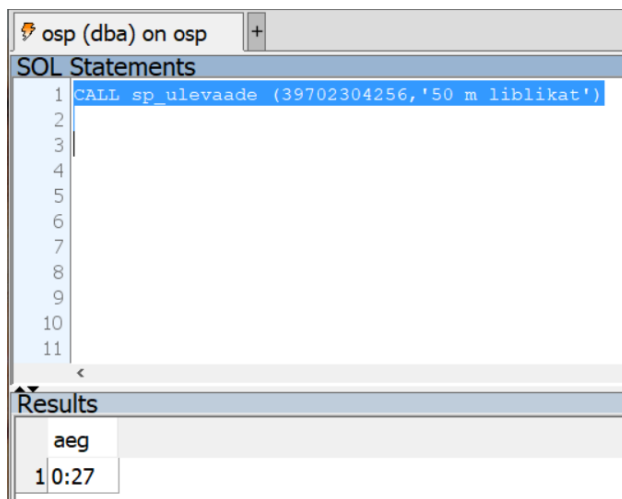
	võistluse_nimi	basseini_nimi	basseini_pikkus	radade_arv
1	Briis Open	Nõmme Ujula sportbassein	25	6
2	Nordic Swimming Championships	Pirita Marina Spordikeskus sportbassein	25	6

Joonis 5 Väljund 3. päringule

4. Leida ujuja aeg, mingi kindla ala ja id-koodi alusel. Kasutaja sisestab ujuja id-koodi ja ala nimetuse (distsants ja stiil kujul nt 50m liblikat) ja saab vastuseks ujuja parima aja sellel alal.

```
CREATE PROCEDURE sp_ulevaade (IN idkood varchar(11), IN ala varchar (50))  
RESULT (aeg TIME)  
BEGIN  
SELECT Tulemused.Aeg as aeg  
FROM Tulemused key join Alad  
WHERE Ala = Alad.Distsants_stiil AND  
idkood = Tulemused.IDkood  
END;
```

```
CALL sp_ulevaade(39702304256, '50 m liblikat')
```

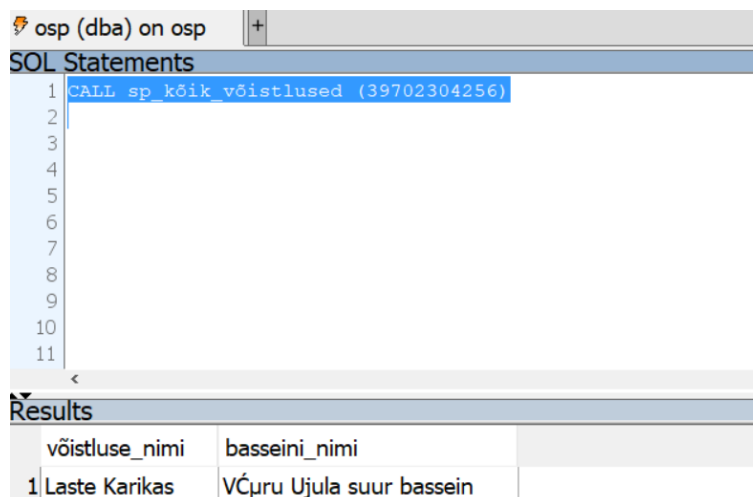


Joonis 6 Väljund 4. päringule

5. Leida kõik võistlused, kus ujuja on osalenud. Kasutaja sisestab ujuja id koodi ja väljundiks on tabel võistluse nimega ja basseini, kus see võistlus toimus, nimega.

```
CREATE PROCEDURE sp_kõik_võistlused (IN Idkood varchar(11))
RESULT (võistluse_nimi varchar (200),
        basseini_nimi varchar (200))
BEGIN
SELECT Võistlused.Nimi as võistluse_nimi,
        Basseinid.Nimi as basseini_nimi
FROM Võistlused, Basseinid, Ujujad, Tulemused
WHERE
Idkood = Ujujad.IDkood AND
Ujujad.IDkood = Tulemused.IDkood AND
Võistlused.bassein = Basseinid.Nimi AND
Tulemused.Võistluse_id=Võistlused.Id;
END;
```

CALL sp_kõik_võistlused (39702304256)



The screenshot shows a database query tool interface. At the top, there's a tab labeled 'osp (dba) on osp'. Below it, the 'SQL Statements' section contains a single line of SQL code: 'CALL sp_kõik_võistlused (39702304256)'. The 'Results' section below shows a table with two columns: 'võistluse_nimi' and 'basseini_nimi'. The first row of data shows 'Laste Karikas' under the first column and 'Võõru Ujula suur bassein' under the second column.

	võistluse_nimi	basseini_nimi
1	Laste Karikas	Võõru Ujula suur bassein

Joonis 7 Väljund 5. päringule