

Jõusaaliketi andmebaas

Rühmaliikmed: Stefan Nezgoda (10. rühm), Tristan Šaraškin (10. rühm)

I etapp

1. Valdkonna ja selle mõistete kirjeldus

Valitud valdkonnaks on jõusaalikett. Inimesed käivad jõusaalis, et teha trenni ning seeläbi arendada oma füüsilist tervist. Jõusaalikel on erinevates asukohtades klubid, mis on varustatud erinevate treeningvahenditega, mis on valdavalt orienteeritud kardiovaskulaarse ning jõutreeninguga tegelemisele. Jõutreeningu vahendeid saab jagada omakorda "masinateks" ja "vabadeks raskusteks". Masinad on mõeldud konkreetse lihase või lihasgrupi isoleerimiseks ning arendamiseks ning nad sisaldavad liikuvaid osi, näiteks reiesirutusmasin. Vabad raskused nagu kangid, hantlid ja pommid on mõeldud harjutuste jaoks, mis samuti arendavad konkreetsemat lihasgruppi, kuid kaasavad ka palju teisi lihaseid ning arendavad seeläbi kogu keha terviklikumalt.

Klubides tegutsevad ka treenerid, kes aitavad jõusaali klientidel seada eesmärgid, valida sobivaid harjutusi ning sooritada neid harjutusi korrektse tehnikaga, lisaks saab treener adekvaatsemalt hinnata kliendi progressi ning nõustada klienti toitumise osas.

Selleks, et klient saaks jõusaaliklubi külastada ning selle teenuseid kasutada, peab ta omama tellimust. Tellimus võimaldab kliendil külastada ükskõik millist jõusaaliklubi ning sealsete vahenditega treenida. Tellimus määrab, mis klubi on kliendi koduklubiks, kus ta treenib kõige sagedamini ning kus ta saab võtta endale treeneri, perioodi, mille jooksul ta saab jõusaaliklubisid külastada, ning rahalise summa, mille ta peab perioodi eest tasuma.

2. Valitud valdkonna probleem ning milliseid kitsaskohti likvideeritakse

Jõusaalikel on erinevaid klubisi, milles treenivad erinevad kliendid, kellel on erinevad tellimused ning treenerid, kes neid aitavad. See tähendab, et ühel jõusaalikel on hallata väga palju infot ning andmebaas aitaks jõusaalikel saada hea ülevaate oma klubide inventarist, klientuurist, klientide tellimustest ning treenerite tegevusest. Lisaks soovivad kliendid näha andmeid oma tellimuse kohta ning arvestada, millal nad peavad tegema järgmise makse. Andmebaasi kaudu saab klient vaadata, millised treenerid tegutsevad tema koduklubis ning seeläbi saab ta valida endale meelepärase treeneri. Treenerid saavad andmebaasi kaudu kontrollida, kes nende treenitavad kliendid on.

3. Andmebaasi kasutajad ning küsimused, millele nad vastuseid soovivad saada

Andmebaasi kasutajad oleksid jõusaalide administraatorid, juhid, raamatupidajad ning treenerid. Nad sooviksid teada näiteks, milline tellimus on kasutajate seas kõige populaarsem, millistes kohtades on inimeste poolt kõige ülekoormatud jõusaalid ja kus tuleb avada uusi, kes eelistab treenida personaaltreeneriga, millised treenerid on kõige populaarsemad ja kellel on rohkem kliente jpm. Klientidel oleks võimalus vaadata, millised vahendid on mis jõusaaliklubides olemas ning millised treenerid on mis jõusaaliklubides saadaval. Kliendid saaksid ka jälgida oma tellimuse staatust ning näha oma eelnevaid külastusi jõusaaliklubidesse.

4. Ärireeglid ja nendest tulenevad seoste aarsused

- Igal klubil on unikaalne nimi.
- Klient saab aktiivse tellimusega külastada ükskõik millist klubi.
- Kliendi tellimus peab olema tasutud, et ta saaks jõusaali külastada.
- Tellimusel on kindel kehtivusperiood.
- Klient võib külastada jõusaali piiramatult oma tellimuse perioodi jooksul.
- Ühel kliendil korraga võib olla ainult üks tellimus.
- Treener töötab ainult ühes jõusaaliklubis.
- Klient saab endale valida treeneri ainult sellest klubist, mis on tema tellimuses määratud tema koduklubiks.
- Erinevad klubid võivad sisaldada erinevaid vahendeid.
- Eeldame, et treenerid vabal ajal jõusaalis ei käi, sest nad saavad piisavalt treenida klientidega koos, seega nendel tellimust ei ole, seega nad ei ole kliendid ja nende külastuse ajad ei lähe arvesse.

Ühel kliendil võib olla üks treener ja üks treener võib treenida mitut klienti - 1:n seos.

Üks klient võib omada üht tellimust ja ühele tellimusele vastab üks klient - 1:1 seos.

Ühel treeneril võib olla üks koduklubi ja ühes klubis võib tegutseda mitu treenerit - 1:n seos.

Ühel tellimusel võib olla üks koduklubi ja üks klubi võib olla koduklubiks mitmele tellimusele - 1:n seos.

Üks klient saab külastada mitut erinevat klubi ning ühte klubi saab külastada mitu erinevat klienti - n:m seos

Üks vahend võib olla mitmes klubis ja ühes klubis võib olla mitu vahendit - m:n seos

Olemitüübid:

kliendid:

- eesnimi
- perenimi
- isikukood
- synniaeg

- sugu

klubid:

- nimi
- linn

vahendid:

- selles tabelis on toodud erinevate treeningvahendite tüübid
- tyyp
- nimetus

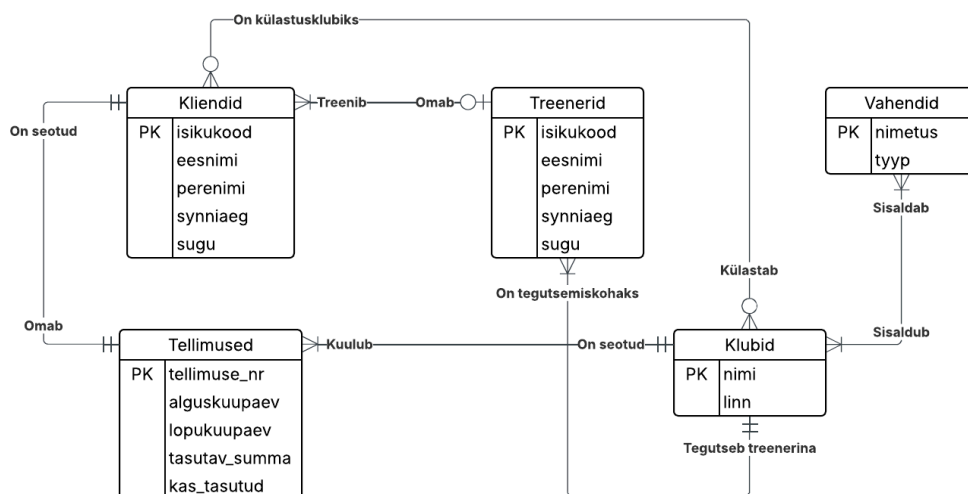
treenerid:

- eesnimi
- perenimi
- isikukood
- synniaeg
- sugu

tellimused:

- tellimuse_nr
- algusaeg
- lõppemisaeg
- tasutav_summa
- kas_tasutud

Andmebaasi ER diagramm



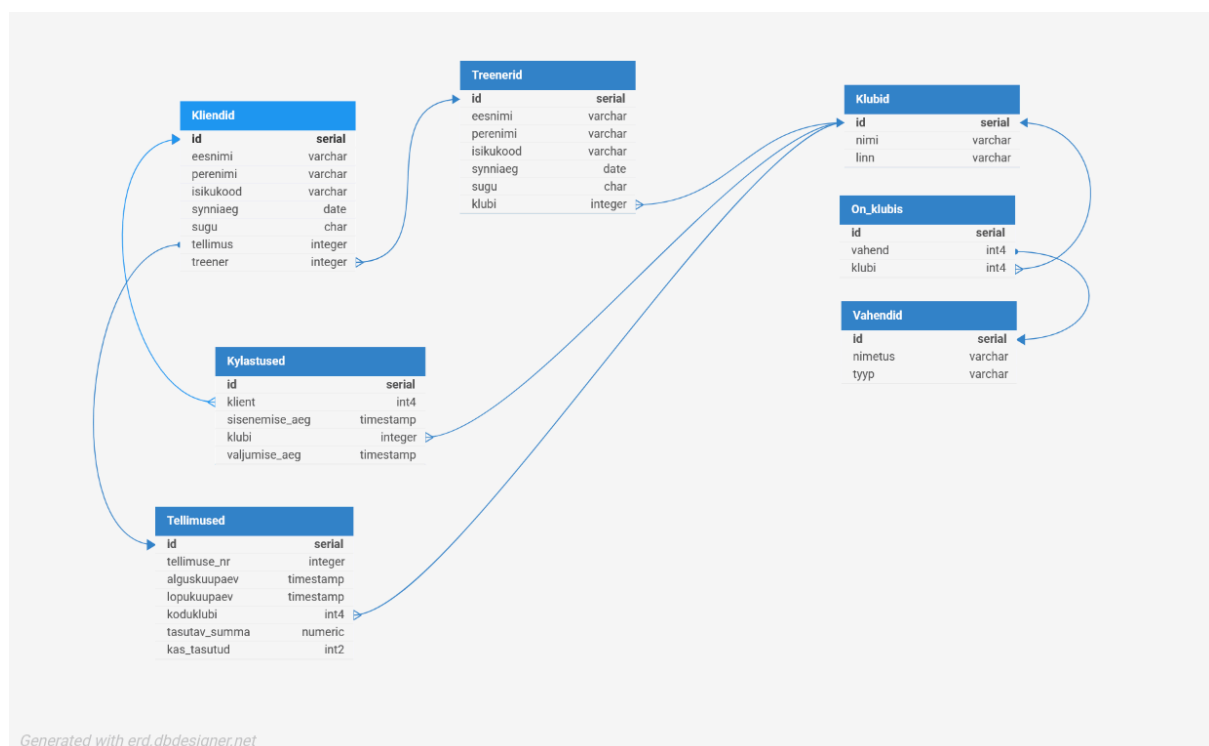
II etapp

1. Mudeli parandused ja täiendused saadud kommentaaride põhjal

Vastavalt tagasisidele on tehtud mudelis järgmised parandused-täiendused (märgitud rohelisega I etapi all):

- Lisatud ärireegel, mille kohaselt treenerid pole kliendid ning neil puudub tellimus ja külastusi nende puhul ei registreerita.
- Seosed “üks klient saab teha mitut külastust ja ühte külastust saab teha üks klient - 1:n seos” ja “üks külastus saab toimuda ühes klubis ja ühes klubis saab toimuda mitu külastust - 1:n seos” on koondatud seoseks “üks klient saab külastada mitut erinevat klubi ning ühte klubi saab külastada mitu erinevat klienti - n:m seos”.
- olemitüübile Tellimused on lisatud juurde atribuut tellimuse_nr, mis identifitseerib üheselt iga kliendi tellimuse ning eemaldatud on atribuut isik.
- olemitüübi Klubid atribuut asukoht on muudetud atribuudiks linn.
- olemitüüpide atribuudid, mis on tegelikkuses välisvõtmed, on I etapi alt eemaldatud.
- esialgne ER diagramm on asendatud korrektsema I etapile vastava ER diagrammiga.
- Külastused ja On_klubis on eemaldatud olemitüüpide alt, sest nad on olemuselt seosetabelid:

2. Andmemudeli relatsiooniline kuju



3. Relatsioonide atribuutide metaandmed

Ühelgi atribuudil ei ole NULL väärtus lubatud, v.a Kliendid relatsioonis treener atribuut

Kliendid:

- eesnimi - varchar(70)
- perenimi - varchar(100)
- isikukood - varchar(11)
- sünniaeg - date
- sugu - char(1)
- tellimus - integer
- treener - integer

Klubid:

- nimi - varchar(100)
- linn - varchar(50)

Vahendid:

- selles tabelis on toodud erinevate treeningvahendite tüübid
- tüüp - varchar(80)
- nimetus - varchar(80)

Treenerid:

- eesnimi - varchar(100)
- perenimi - varchar(70)
- isikukood - varchar(11)
- klubi - integer
- sünniaeg - date
- sugu - char(1)

Tellimused:

- tellimuse_nr - integer
- alguskuupaev - timestamp
- lopukuupaev - timestamp
- koduklubi - integer
- tasutav_summa - numeric(5, 2)
- kas_tasutud - int2 (0 kui tasumata, 1 kui tasutud)

Kylastused:

- klient - integer
- klubi - integer
- sisenemise_aeg - timestamp

- valjumise_aeg - timestamp

On_klubis:

- kirje tabelis näitab, kas antud masin on konkreetses klubis olemas
- vahend - integer
- klubi - integer

4. Olemitüübid relatsioonidena, ilma kunstlike võtmeteta

Kliendid(eesnimi, perenimi, isikukood, synniaeg, sugu, tellimus, treener)

Treenerid(eesnimi, perenimi, isikukood, synniaeg, sugu, klubi)

Tellimused(tellimuse_nr, alguskuupaev, lopukuupaev, koduklubi, tasutav_summa, kas_tasutud)

Kylastused(klient, sisenemise_aeg, klubi, valjumise_aeg)

Klubid(nimi, linn)

Vahendid(nimetus, tyyp)

On_klubis(vahend, klubi)

5. Andmebaasi relatsioonide kolmandale normaalkujule viimine

1. Kliendid(eesnimi, perenimi, isikukood, synniaeg, sugu, tellimus, treener)

funktsionaalsed sõltuvused:

isikukood → eesnimi, perenimi, synniaeg, sugu, tellimus, treener

Relatsioon on esimesel normaalkujul, sest kõik antud relatsiooni atribuudid sisaldavad atomaarseid väärtusi. Relatsiooni primaarvõti on üheatribuudiline, mistõttu on antud relatsioon teisel normaalkujul. Relatsioon Kliendid on ka kolmandal normaalkujul, sest ükski relatsiooni sekundaarne atribuut ei sõltu transitiivselt isikukoodist.

2. Treenerid(eesnimi, perenimi, isikukood, synniaeg, sugu, klubi)

funktsionaalsed sõltuvused:

isikukood → eesnimi, perenimi, synniaeg, sugu, klubi

Relatsioon treenerid on esimesel normaalkujul, sest kõik atribuudid võivad sisaldada ainult atomaarseid väärtuseid. Relatsioon on ka teisel normaalkujul, sest primaarvõti on üheatribuudiline. Relatsioon vastab kolmandale normaalkujule, sest ükski sekundaarne atribuut ei sõltu transitiivselt primaarvõtmest.

3. Tellimused(tellimuse_nr, alguskuupaev, lopukuupaev, koduklubi, tasutav_summa, kas_tasutud)

funktsionaalsed sõltuvused:

tellimuse_nr → alguskuupaev, lopukuupaev, koduklubi, tasutav_summa, kas_tasutud

Relatsioon on esimesel normaalkujul, sest kõik atribuudid võivad sisaldada ainult ühte väärtust. Relatsioon on ka teisel normaalkujul, sest primaarvõti koosneb ühest atribuudist. Kuna relatsioonis ei esine transitiivseid sõltuvusi primaarvõtmest, siis on relatsioon ka kolmandal normaalkujul.

4. Kylastused(klient, sisenemise_aeg, klubi, valjumise_aeg)

funktsionaalsed sõltuvused:

klient, sisenemise_aeg → klubi, valjumise_aeg

Relatsioon vastab esimesele normaalkujule, sest atribuudid sisaldavad atomaarseid väärtusi. Ükski relatsiooni sekundaarsetest atribuutidest ei sõltu primaarvõtme võimalikest pärisosahulkadest, mistõttu on relatsioon teisel normaalkujul. Transitiivseid sõltuvusi primaarvõtmest ei esine, mistõttu on relatsioon kolmandal normaalkujul.

5. Klubid(nimi, linn)

funktsionaalsed sõltuvused:

nimi → linn

Relatsioon vastab esimesele normaalkujule, sest kõik atribuudid saavad sisaldada korraga vaid ühte väärtust. Relatsioon on teisel normaalkujul, sest primaarvõti on üheatribuudiline. Relatsioon on ka kolmandal normaalkujul, sest transitiivseid sõltuvusi primaarvõtmest ei esine.

6. Vahendid(nimetus, tüüp)

funktsionaalsed sõltuvused:

nimetus → tüüp

Relatsioon vastab esimesele normaalkujule, sest atribuudid on atomaarsed. Relatsioon vastab üheatribuudilise primaarvõtme tõttu ka teisele normaalkujule. Transitiivsete sõltuvuste puudumisest tingituna on relatsioon vastavuses kolmanda normaalkujuga.

7. On_klubis(vahend, klubi)

Relatsioon on esimesel normaalkujul, sest iga atribuut saab sisaldada vaid atomaarseid väärtuseid. Relatsioon on ka teisel normaalkujul, sest primaarvõtmeks on mõlemad

atribuudid, mis teineteisest ei sõltu. Transitiivsete sõltuvuste puudumise tõttu on relatsioon ka kolmandal normaalkujul.

6. Normaliseeritud andmebaasi relatsioonid

Kliendid(id, eesnimi, perenimi, isikukood, sünniaeg, sugu, tellimus, treener)

Treenerid(id, eesnimi, perenimi, isikukood, sünniaeg, sugu, klubi)

Tellimused(id, tellimuse_nr, alguskuupäev, lõpukuupäev, koduklubi, tasutav_summa, kas_tasutud)

Kylastused(id, klient, sisenemise_aeg, klubi, valjumise_aeg)

Klubid(id, nimi, linn)

Vahendid(nimetus, tüüp)

On_klubis(id, vahend, klubi)

III etapp

1. II etapi parandused

Relatsioonilisest mudelist on eemaldatud relatsioon 'Isikud' ning ER-diagramm ja relatsiooniline mudel on viidud omavahel kooskõlla. Relatsioonis vahendid on atribuut tüüp jäetud tabelisse, sest treeningvahendite võimalikke liigitusi on palju ning nende viimine eraldi relatsioonideks komplitseeriks ülemääraselt andmebaasi.

2. Andmebaasi loomise laused koos dokumenteeritud parandustega

Parandused: Tabelite ja veeru nimede ümbert eemaldati skripti eksportimisel loodud jutumärgid, jutumärgid said eemaldatud ka kitsenduste nimedest. Kliendid tabelis sai atribuudile "tellimus" lisatud kitsendus UNIQUE, sest igal kliendiga on seotud vaid üks tellimus ning sama tellimuse sidumine erinevate klientidega ei tohiks olla võimalik. Kitsendusele fk_klient_tellimus sai lisatud tingimus ON DELETE CASCADE, et tellimuse kustutamisel kustutataks ka kliendi kirje klientide tabelist. Kitsendusele fk_klient_treener sai lisatud tingimus ON DELETE SET NULL, et treeneri kirje kustumisel treenerid tabelist seataks kliendi treener olematuks. Kitsendustele fk_on_klubis_klubi ja fk_on_klubis_vahend said lisatud tingimused ON DELETE CASCADE, et vahendi eemaldamisel vahendite tabelist või klubi eemaldamisel klubide tabelist kustuksid vastavad kirjed on_klubis tabelis.

Tabelite loomise ja kitsenduste rakendamiste laused:

```
CREATE TABLE Kliendid (  
    id serial NOT NULL PRIMARY KEY,  
    eesnimi varchar(70) NOT NULL,  
    perenimi varchar(100) NOT NULL,  
    isikukood varchar(11) NOT NULL UNIQUE,  
    synniaeg date NOT NULL,  
    sugu varchar(1) NOT NULL,  
    tellimus int NOT NULL UNIQUE,  
    treener int  
);
```

```
CREATE TABLE Treenerid (  
    id serial NOT NULL PRIMARY KEY,  
    eesnimi varchar(100) NOT NULL,  
    perenimi varchar(70) NOT NULL,  
    isikukood varchar(11) NOT NULL UNIQUE,  
    synniaeg date NOT NULL,  
    sugu varchar(1) NOT NULL,  
    klubi int NOT NULL  
);
```

```
CREATE TABLE Klubid (  
    id serial NOT NULL PRIMARY KEY,  
    nimi varchar(100) NOT NULL UNIQUE,  
    linn varchar(50) NOT NULL  
);
```

```
CREATE TABLE Tellimused (  
    id serial NOT NULL PRIMARY KEY,  
    tellimuse_nr int NOT NULL,  
    alguskuupaev timestamp NOT NULL,  
    lopukuupaev timestamp NOT NULL,  
    koduklubi int NOT NULL,  
    tasutav_summa numeric(5, 2) NOT NULL,  
    kas_tasutud bit(1) NOT NULL  
);
```

```
CREATE TABLE Kylastused (  
    id serial NOT NULL PRIMARY KEY,  
    klient int NOT NULL,  
    sisenemise_aeg timestamp NOT NULL,
```

```
        klubi int NOT NULL,  
        valjumise_aeg timestamp NOT NULL  
    );
```

```
CREATE TABLE Vahendid (  
    id serial NOT NULL PRIMARY KEY,  
    nimetus varchar(80) NOT NULL,  
    tyyp varchar(80) NOT NULL  
);
```

```
CREATE TABLE On_klubis (  
    id serial NOT NULL PRIMARY KEY,  
    vahend int NOT NULL,  
    klubi int NOT NULL  
);
```

```
ALTER TABLE Kliendid  
ADD CONSTRAINT fk_klient_tellimus  
FOREIGN KEY (tellimus)  
REFERENCES Tellimused(id)  
ON DELETE CASCADE;
```

```
ALTER TABLE Kliendid  
ADD CONSTRAINT fk_klient_treener  
FOREIGN KEY (treener)  
REFERENCES Treenerid(id)  
ON DELETE SET NULL;
```

```
ALTER TABLE Treenerid  
ADD CONSTRAINT fk_treener_klubi  
FOREIGN KEY (klubi)  
REFERENCES Klubid(id);
```

```
ALTER TABLE Tellimused  
ADD CONSTRAINT fk_tellimus_koduklubi  
FOREIGN KEY (koduklubi)  
REFERENCES Klubid(id);
```

```
ALTER TABLE Kylastused  
ADD CONSTRAINT fk_kylastus_klient  
FOREIGN KEY (klient)  
REFERENCES Kliendid(id);
```

```
ALTER TABLE Kylastused
```

```
ADD CONSTRAINT fk_kylastus_klubi
FOREIGN KEY (klubi)
REFERENCES Klubid(id);
```

```
ALTER TABLE On_klubis
ADD CONSTRAINT fk_on_klubis_vahend
FOREIGN KEY (vahend)
REFERENCES Vahendid(id)
ON DELETE CASCADE;
```

```
ALTER TABLE On_klubis
ADD CONSTRAINT fk_on_klubis_klubi
FOREIGN KEY (klubi)
REFERENCES Klubid(id)
ON DELETE CASCADE;
```

3. Fiktiivsete andmete sisestamise laused

```
INSERT INTO klubid(nimi, linn)
VALUES
('Tasku', 'Tartu'),
('Veeriku', 'Tartu'),
('Vanalinna', 'Tallinn'),
('Ülemiste', 'Tallinn'),
('Port Arturi', 'Pärnu')
;
```

```
INSERT INTO treenerid(eesnimi, perenimi, isikukood, synniaeg, sugu, klubi)
VALUES
('Mihkel', 'Mets', '38110084941', '1981-10-08', 'm', 1),
('Kevin', 'Kuusepuu', '39206222774', '1992-06-22', 'm', 3),
('Jaan', 'Oja', '38806105750', '1988-06-10', 'm', 5),
('Liis', 'Rand', '49604035752', '1996-04-03', 'n', 2),
('Maria', 'Ivanova', '49901087056', '1999-01-08', 'n', 4)
;
```

```
INSERT INTO tellimused(tellimuse_nr, alguskuupaev, lopukuupaev, koduklubi,
tasutav_summa, kas_tasutud)
VALUES
(202400001, '2024-08-21', '2025-06-01', 2, 130.99, '1'),
(202300082, '2023-01-01', '2026-01-01', 4, 550, '1'),
(202500102, '2025-03-01', '2026-02-01', 1, 215.99, '1'),
(202500113, '2025-05-01', '2026-01-01', 2, 139.99, '0'),
```

(202400002, '2024-09-01', '2025-08-31', 3, 215.99, '0')

;

INSERT INTO Kliendid(eesnimi, perenimi, isikukood, synniaeg, sugu, tellimus, treener)
VALUES

('Tarmo', 'Tamme', '39703016580', '1997-03-01', 'm', 2, 5),

('Madis', 'Kask', '50601282776', '2006-01-28', 'm', 3, 1),

('Susanna', 'Saar', '60302012277', '2003-02-01', 'n', 1, 4),

('Tiiu', 'Tihane', '48006286546', '1980-06-28', 'n', 4, 4),

('Peeter', 'Karu', '37111205723', '1971-11-20', 'm', 5, NULL)

;

INSERT INTO kylastused(klient, sisenemise_aeg, klubi, valjumise_aeg)
VALUES

(6, '2024-10-10 18:15:30', 4, '2024-10-10 20:30:00'),

(10, '2025-01-15 15:00:00', 3, '2025-01-15 18:17:00'),

(7, '2024-09-01 08:12:00', 2, '2024-09-01 09:30:00'),

(7, '2024-11-04 08:06:00', 1, '2024-11-04 09:23:00'),

(10, '2025-01-27 16:12:00', 3, '2025-01-27 18:53:00')

;

INSERT INTO vahendid(nimetus, tyyp)
VALUES

('Rinnalt surumise pink', 'pink'),

('Kükiraam', 'raam'),

('Jooksulint', 'masin'),

('Hantliriiul hantlitega', 'riiul'),

('Sõudeergomeeter', 'masin'),

('Reiesirutusmasin', 'masin'),

('Jõutõmbe platvorm', 'platvorm'),

('Võimlemisraam', 'raam')

;

INSERT INTO on_klubis(vahend, klubi)
VALUES

(1, 5),

(2, 5),

(3, 5),

(4, 5),

(1, 2),

(3, 2),

(5, 2),

(6, 2),

(8, 2),

(1, 1),
(2, 1),
(4, 1),
(6, 1),
(7, 1),
(1, 3),
(2, 3),
(3, 3),
(4, 3),
(5, 3),
(6, 3),
(7, 3),
(8, 3),
(1, 4),
(2, 4),
(5, 4),
(6, 4),
(7, 4),
(8, 4)
;

4. Päringud andmebaasi pealt

1. Vaade v_vahendid_klubides, mis näitab jõusaaliklubide kaupa, mis vahendid nendes klubides olemas on. Väljastatud kirjed on järjestatud tähestikuliselt jõusaaliklubide nimede põhjal. Selle vaate põhjal on klientidel võimalik vaadata, mis vahendid konkreetsetes klubides olemas on.

vaate loomine:

```
CREATE VIEW v_vahendid_klubides AS  
SELECT nimi AS "klubi", nimetus AS "treeningvahend"  
FROM klubid k, vahendid v, on_klubis o  
WHERE k.id = o.klubi AND o.vahend = v.id  
ORDER BY 1 ASC;
```

vaade:

```
SELECT * FROM v_vahendid_klubides;
```

SELECT * FROM v_vahendid_klubides

Grid	Az klubi	Az treeningvahend
1	Port Arturi	Kükiraam
2	Port Arturi	Jooksulint
3	Port Arturi	Hantliriul hantlitega
4	Port Arturi	Rinnalt surumise pink
5	Tasku	Rinnalt surumise pink
6	Tasku	Kükiraam
7	Tasku	Jõutõmbe platvorm
8	Tasku	Reiesirutusmasin
9	Tasku	Hantliriul hantlitega
10	Ülemiste	Võimlemisraam
11	Ülemiste	Rinnalt surumise pink
12	Ülemiste	Kükiraam
13	Ülemiste	Sõudeergomeeter
14	Ülemiste	Reiesirutusmasin
15	Ülemiste	Jõutõmbe platvorm
16	Vanalinna	Rinnalt surumise pink
17	Vanalinna	Jooksulint
18	Vanalinna	Hantliriul hantlitega
19	Vanalinna	Sõudeergomeeter
20	Vanalinna	Reiesirutusmasin
21	Vanalinna	Jõutõmbe platvorm
22	Vanalinna	Võimlemisraam
23	Vanalinna	Kükiraam
24	Veeriku	Võimlemisraam
25	Veeriku	Reiesirutusmasin
26	Veeriku	Sõudeergomeeter
27	Veeriku	Jooksulint
28	Veeriku	Rinnalt surumise pink


2. Vaade, mis näitab erinevate klubide treenereid, kes on järjestatud treenitavate arvu järgi. Võrdsete treenitavate arvu puhul järjestatakse kirjed tähestikuliselt treenerite nimede põhjal. Vaatest oleks kasu klientidele, kes soovivad näha, millised treenerid on populaarsed ehk potentsiaalselt head ning millistes klubides nad tegutsevad.

Vaate loomine:

```
CREATE VIEW v_treenerid_treenitavate_jargi AS
SELECT t.eesnimi || ' ' || t.perenimi AS "treener", kl.nimi AS "klubi", count(k.treener) AS
"treenitavate arv"
FROM treenerid t
JOIN klubid kl ON t.klubi = kl.id
LEFT JOIN kliendid k ON k.treener = t.id
GROUP BY t.eesnimi, t.perenimi, kl.nimi
ORDER BY 3 DESC, 1;
```

Vaade:

```
SELECT * FROM v_treenerid_treenitavate_jargi;
```



	treener	klubi	treenitavate arv
1	Liis Rand	Veeriku	2
2	Maria Ivanova	Ülemiste	1
3	Mihkel Mets	Tasku	1
4	Jaan Oja	Port Arturi	0
5	Kevin Kuusepuu	Vanalinna	0

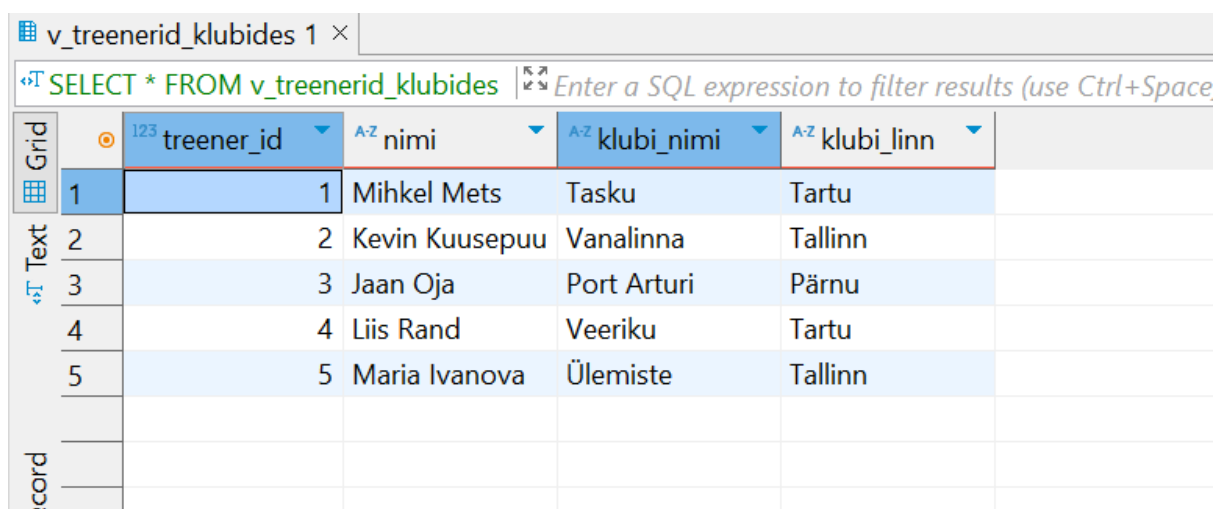
3. Vaade, mis näitab treenereid ja jõusaale, kus nad töötavad. Sellest vaatest oleks kasu klientidele, kes soovivad leida oma koduklubisse sobiva treeneri.

Vaate loomine:

```
CREATE VIEW v_treenerid_klubides AS  
SELECT t.id AS treener_id, t.eesnimi || ' ' || t.perenimi AS nimi, k.nimi AS klubi_nimi,  
k.linn AS klubi_linn  
FROM treenerid t  
JOIN klubid k ON t.klubi = k.id;
```

Vaade:

```
SELECT * FROM v_treenerid_klubides;
```



	treener_id	nimi	klubi_nimi	klubi_linn
1	1	Mihkel Mets	Tasku	Tartu
2	2	Kevin Kuusepuu	Vanalinna	Tallinn
3	3	Jaan Oja	Port Arturi	Pärnu
4	4	Liis Rand	Veeriku	Tartu
5	5	Maria Ivanova	Ülemiste	Tallinn

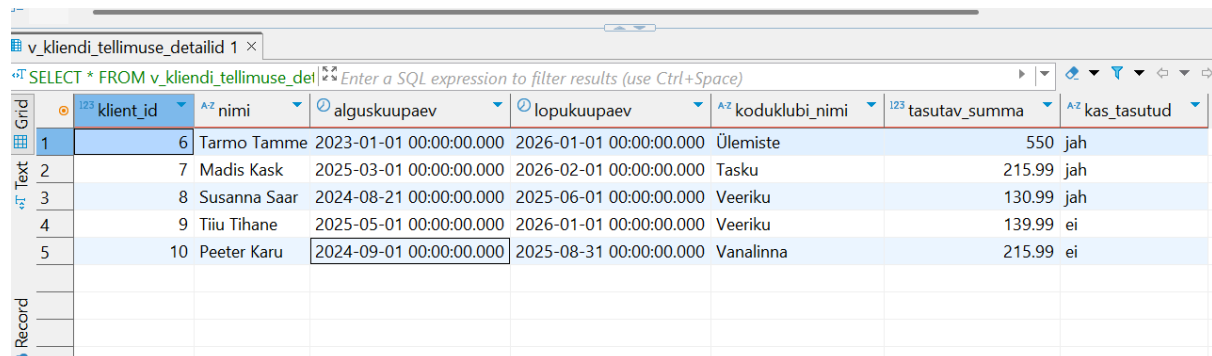
4. Vaade, mis näitab kõiki vajalikke detaile kliendi tellimuse kohta: alguskuupäev, lõppkuupäev, koduklubi, tasumisele kuuluv summa ning lisaks on näha, kas tellimus on tasutud või mitte. Vaade on väga oluline, kuna selle abil saab kontrollida ja otsustada, kas kliendile lubada sissepääs või mitte.

Vaate loomine:

```
CREATE VIEW v_kliendi_tellimuse_detailid AS
SELECT k.id AS klient_id, k.eesnimi || ' ' || k.perenimi AS nimi, t.alguskuupäev,
t.loppkuupäev, kl.nimi AS koduklubi_nimi, t.tasutav_summa,
CASE
WHEN t.kas_tasutud = B'1' THEN 'jah'
ELSE 'ei'
END AS kas_tasutud
FROM kliendid k
JOIN tellimused t ON k.tellimus = t.id
JOIN klubid kl ON t.koduklubi = kl.id;
```

Vaade:

```
SELECT * FROM v_kliendi_tellimuse_detailid;
```



	klient_id	nimi	alguskuupäev	lopukuupäev	koduklubi_nimi	tasutav_summa	kas_tasutud
1	6	Tarmo Tamme	2023-01-01 00:00:00.000	2026-01-01 00:00:00.000	Ülemiste	550	jah
2	7	Madis Kask	2025-03-01 00:00:00.000	2026-02-01 00:00:00.000	Tasku	215.99	jah
3	8	Susanna Saar	2024-08-21 00:00:00.000	2025-06-01 00:00:00.000	Veeriku	130.99	jah
4	9	Tiiu Tihane	2025-05-01 00:00:00.000	2026-01-01 00:00:00.000	Veeriku	139.99	ei
5	10	Peeter Karu	2024-09-01 00:00:00.000	2025-08-31 00:00:00.000	Vanalinna	215.99	ei

5. Tegevused andmebaasiga

1. Protseduur, mis eemaldab kõik aegunud tellimused (tellimused, mille lõpukuupäev on varem kui praegune kuupäev) tellimuste tabelist ning väljastab vastava teate. Kui aegunud tellimusi pole, siis väljastatakse vastav teade ja protseduur ei tee midagi. Tellimuse kustumisel kustub välisvõtme piirangu tõttu ka kliendi kirje klientide tabelist.

Protseduuri loomise lause:


```

CREATE OR REPLACE PROCEDURE sp_eemalda_aegunud_tellimused()
LANGUAGE plpgsql
AS $$
declare mitu_aegunud integer;
BEGIN
    select count(*) into mitu_aegunud
    from tellimused
    where lopukuupaev < current_date;
    if mitu_aegunud > 0 then
        delete from tellimused where lopukuupaev < current_date;
        raise notice 'Aegunud tellimused on eemaldatud.';
    else
        raise notice 'Aegunud tellimusi, mida eemaldada, ei ole.';
    end if;
END;
$$;

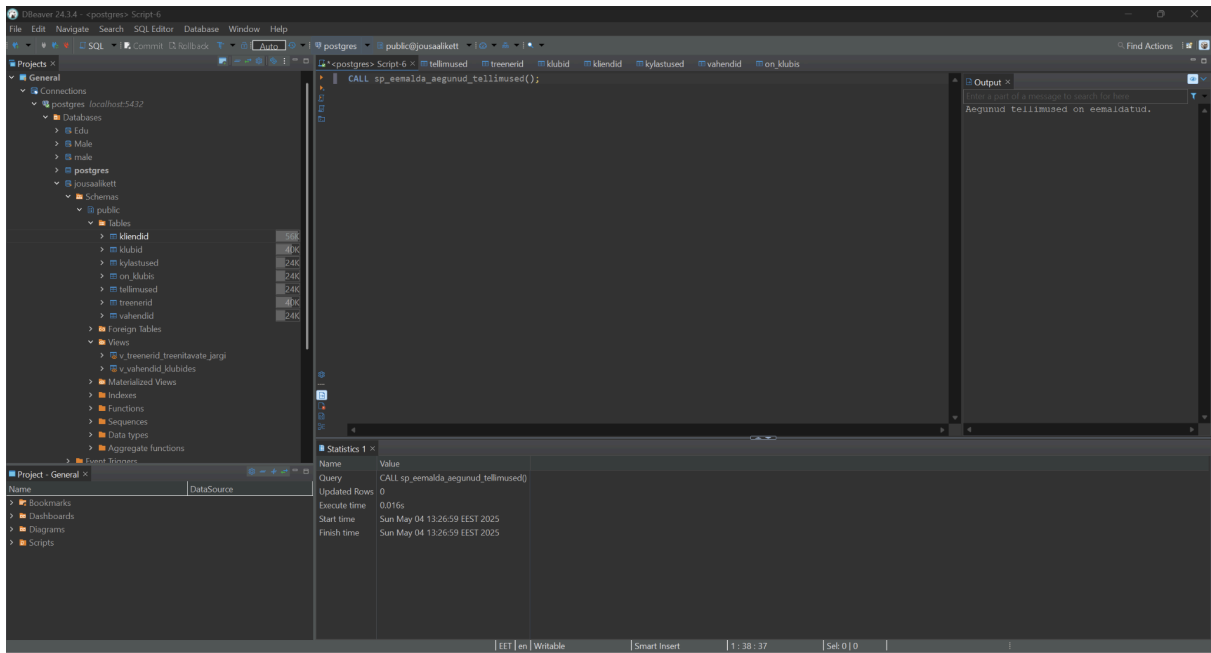
```

Tellimused ja kliendid tabelid enne protseduuri kutsumist:

tellimused								
Enter a SQL expression to filter results (use Ctrl+Space)								
	id	tellimuse_nr	alguskuupaev	lopukuupaev	koduklubi	tasutav_summa	kas_tasutud	
1	6	202,400,001	2024-08-21 00:00:00.000	2025-06-01 00:00:00.000	2	130.99		1
2	7	202,300,082	2023-01-01 00:00:00.000	2026-01-01 00:00:00.000	4	550		1
3	8	202,500,102	2025-03-01 00:00:00.000	2026-02-01 00:00:00.000	1	215.99		1
4	9	202,500,113	2025-05-01 00:00:00.000	2026-01-01 00:00:00.000	2	139.99		0
5	10	202,400,002	2024-09-01 00:00:00.000	2025-08-31 00:00:00.000	3	215.99		0
6	11	202,400,005	2024-09-01 00:00:00.000	2025-01-01 00:00:00.000	1	60		1

kliendid								
Enter a SQL expression to filter results (use Ctrl+Space)								
	id	eesnimi	perenimi	isikukood	synniaeg	sugu	tellimus	treener
1	11	Tarmo	Tamme	39703016580	1997-03-01	m	7	5
2	12	Madis	Kask	50601282776	2006-01-28	m	8	1
3	13	Susanna	Saar	60302012277	2003-02-01	n	6	4
4	14	Tiiu	Tihane	48006286546	1980-06-28	n	9	4
5	15	Peeter	Karu	37111205723	1971-11-20	m	10	[NULL]
6	16	Tiit	Sepp	50011150857	2000-11-15	m	11	[NULL]

Protseduuri väljakutse:



Tellimused ja kliendid tabelid peale protseduuri väljakutsumist:

Enter a SQL expression to filter results (use Ctrl+Space)									
	id	tellimuse_nr	alguskuupaev	lopukuupaev	koduklubi	tasutav_summa	kas_tasutud		
1	6	202,400,001	2024-08-21 00:00:00.000	2025-06-01 00:00:00.000	2	130.99	1		
2	7	202,300,082	2023-01-01 00:00:00.000	2026-01-01 00:00:00.000	4	550	1		
3	8	202,500,102	2025-03-01 00:00:00.000	2026-02-01 00:00:00.000	1	215.99	1		
4	9	202,500,113	2025-05-01 00:00:00.000	2026-01-01 00:00:00.000	2	139.99	0		
5	10	202,400,002	2024-09-01 00:00:00.000	2025-08-31 00:00:00.000	3	215.99	0		

Enter a SQL expression to filter results (use Ctrl+Space)									
	id	eesnimi	perenimi	isikukood	synniaeg	sugu	tellimus	treener	
1	11	Tarmo	Tamme	39703016580	1997-03-01	m	7	5	
2	12	Madis	Kask	50601282776	2006-01-28	m	8	1	
3	13	Susanna	Saar	60302012277	2003-02-01	n	6	4	
4	14	Tiiu	Tihane	48006286546	1980-06-28	n	9	4	
5	15	Peeter	Karu	37111205723	1971-11-20	m	10	[NULL]	

2. Protseduur, mis lisab uue kliendi külastuse tabelisse "külastused". Täpsemalt lisatakse: kliendi ID, sisenemise aeg, klubi ID ja väljumise aeg. Lisaks kontrollib protseduur, et väljumise aeg ei oleks varasem kui sisenemise aeg. Kui see tingimus ei ole täidetud, tagastatakse veateade.

Protseduuri loomise lause:

```
CREATE PROCEDURE sp_lisa_kylastus(  
k_klient_id INTEGER,  
k_sisenemise_aeg TIMESTAMP,  
k_klubi_id INTEGER,  
k_valjumise_aeg TIMESTAMP)  
LANGUAGE plpgsql  
AS $$  
BEGIN  
    IF k_valjumise_aeg < k_sisenemise_aeg THEN  
        RAISE EXCEPTION 'Väljumise aeg peab olema hiljem kui sisenemise aeg!';  
    END IF;  
    INSERT INTO kylastused (klient, sisenemise_aeg, klubi, valjumise_aeg)  
    VALUES (k_klient_id, k_sisenemise_aeg, k_klubi_id, k_valjumise_aeg);  
END;  
$$;
```

Külastused tabel enne protseduuri kutsumist:

Grid	id	klient	sisenemise_aeg	klubi	valjumise_aeg
1	1	6	2024-10-10 18:15:30.000	4	2024-10-10 20:30:00.000
2	2	10	2025-01-15 15:00:00.000	3	2025-01-15 18:17:00.000
3	3	8	2024-09-01 08:12:00.000	2	2024-09-01 09:30:00.000
4	4	8	2024-11-04 08:06:00.000	1	2024-11-04 09:23:00.000
5	5	10	2025-01-27 16:12:00.000	3	2025-01-27 18:53:00.000
6	6	7	2025-05-02 14:30:00.000	3	2025-05-02 16:00:00.000

Protseduuri väljakutse:

```
CALL sp_lisa_kylastus(8, '2025-05-03 10:15:00', 1, '2025-05-03 11:15:00')  
CALL sp_lisa_kylastus(6, '2025-05-03 12:30:00', 3, '2025-05-03 16:00:00')  
CALL sp_lisa_kylastus(10, '2025-05-03 11:30:00', 4, '2025-05-03 16:20:00')
```

Külastused tabel peale protseduuri väljakutsumist:

kylastused <i>Enter a SQL expression to filter results (use Ctrl+Space)</i>						
Grid		123 id	123 klient	sisenemise_aeg	123 klubi	valjumise_aeg
Text	1	1	6	2024-10-10 18:15:30.000	4	2024-10-10 20:30:00.000
	2	2	10	2025-01-15 15:00:00.000	3	2025-01-15 18:17:00.000
	3	3	8	2024-09-01 08:12:00.000	2	2024-09-01 09:30:00.000
	4	4	8	2024-11-04 08:06:00.000	1	2024-11-04 09:23:00.000
	5	5	10	2025-01-27 16:12:00.000	3	2025-01-27 18:53:00.000
	6	6	7	2025-05-02 14:30:00.000	3	2025-05-02 16:00:00.000
	7	7	8	2025-05-03 10:15:00.000	1	2025-05-03 11:15:00.000
	8	8	6	2025-05-03 12:30:00.000	3	2025-05-03 16:00:00.000
	9	9	10	2025-05-03 11:30:00.000	4	2025-05-03 16:20:00.000