

Andmebaaside alused kodutöö: andmebaaside projekt

Kristiina Miller

Valdkonna kirjeldus

Tegemist on koduse arvepidamise programmiga. Andmebaasis on kasutajad (ka kasutajanimed), sissetulekud, väljaminekud, sissetulekuallikad ja kulutüübid.

Sissetulekuallikad on jagatud põhigruppidesse: palk, toetus, renditulu, muu. Ülesande mahu pärast ei ole ma lisanud iga sissetulekuallika jaoks eraldi olemit. Sama ka kulutüüpide osas.

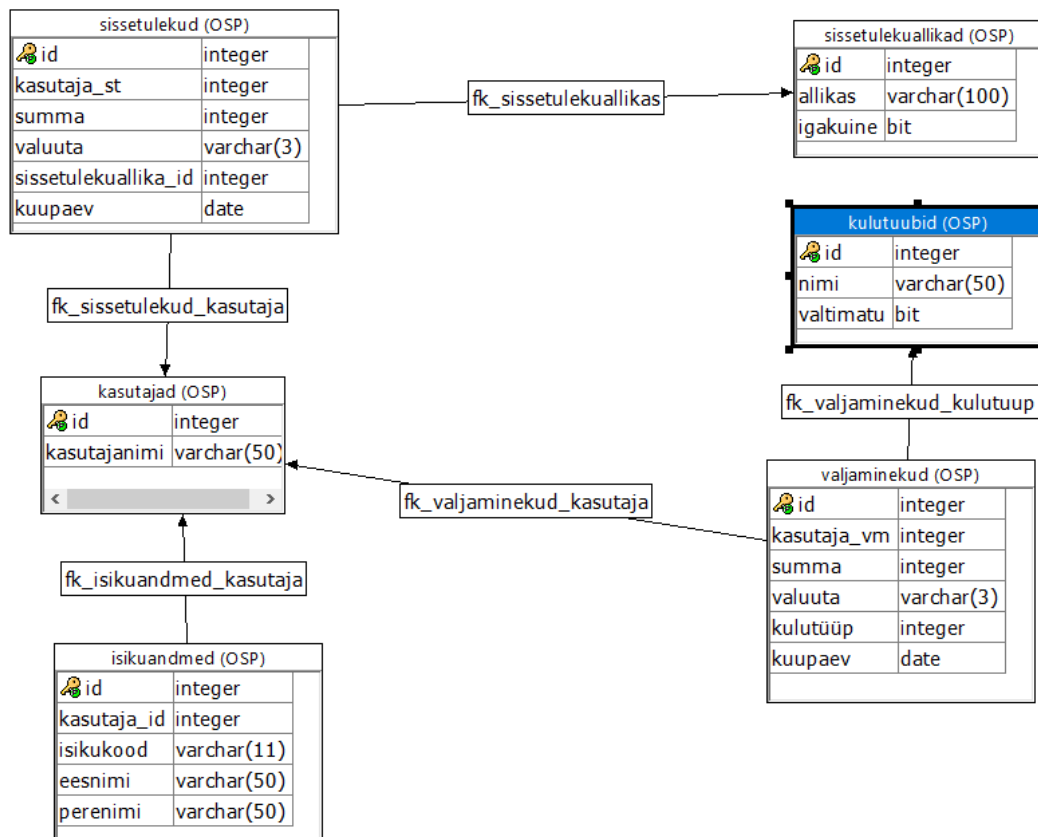
Ideaalis võiksid sissetulekud ja väljaminekud tulla andmebaasi automaatselt. Nt kui makstakse pangakaardiga, siis andmebaasis kajastub mis osteti, kuskohast osteti, palju maksis jne. Teine viis oleks ühendada kliendikaardid selle andmebaasiga – siis kajastuks sularahas maksmine samuti andmebaasis. See annaks kasutajale võimaluse võrrelda nt kus poes on nädala toidukorv kõige soodsam. Samuti annaks lisada nõ.hoiatusi kasutajale. Kui nt toidu peale on nädalas läinud üle 200 EURi, siis väljastatakse kasutajale selle kohane sõnum.

Antud valdkonna probleem (või lahendus) seisneb selles, et selline andmebaas teeb inimeste elu kergemaks. Selline andmebaas annab võimaluse vaadelda oma rahaasju üldiselt, planeerida sääste, otsida soodsamaid hindu poodides, paigaldada maksimum kulutushindasid tüüpidele jne.

Kasutaja võib olla igauks Üksikanimene, paar, perekond. Kui ka lastel on kontod, siis vanemad saaksid vaadata, mille peale nad raha kulutavad.

See andmebaas peaks lahendama inimeste rahaprobleeme ja õpetama neile, kuidas teha paremaid valikuid asju ostes. Kas osta iga nädal riideid või koguda reisiks raha?

Graafiline klassimudel



Mudelil on viis tabelit: sissetulekud, sissetulekuallikad, valjaminekud, kulutuubid, kasutajad ja isikuandmed. Kasutajad said loodud, et andmebaasis eksisteeriks ka nõrk olem.

Andmebaasi administraator saab sisestada andmebaasi oma (nt sisse logides oma kasutajaga) kulutusi ja sissetulekuid. Kuna sissetulekud ja valjaminekud on kõige mahukamad tabelid, siis sai loodud neile sissetulekuallikad ja valjaminekud tabelid, et informatsiooni kiiremini kätte saada ja ka sisestada.

Olemid ja seosed

Isikuandmed, sissetulekuallikad ja kulutüübid on tugevad olemid. Ülejäänud on nõrgemad kui mitte arvestada autoincrementiga loodud ID-sid.

1:1 seos on isikuandmed-kasutajad vahel.

1:n seos on sissetulekuallikad-sissetulekud, kulutuubid-valjaminekud, kasutajad-sissetulekud, kasutajad-valjaminekud vahel.

Välisvõtmed:

Sissetulekuallikad.ID=Sissetulekud.sissetulekuallika_ID

Kulutüübid.ID=Valjaminekud.kulutüüp

Kasutaja.ID=Sissetulekud.kasutaja_st

Kasutaja.ID=Valjaminekud.kasutaja_vm

Relatsioonide teisendus ja normaalkujud

Antud andmebaasist saab kolme relatsiooni:

1. Sissetulekud (sissetulekukande_id, kasutaja_ik, summa, valuuta, sissetulekuallikas, kuupäev, igakuine)
2. Valjaminekud (valjaminekukande_id, kasutaja_ik, summa, valuuta, kulutüüp, kuupäev, vältimatu)
3. Kasutajad (kasutajanimi, isikukood, esnimi, perenimi).

Kõik relatsioonid on kolmandal normaalkujul. Sissetulekukande_id on ainuke võti, millega saab teisi atribuute määrata. Teiste atribuutide kaudu ei saa teisi atribuute määrata. Sama on ka valjaminekukande_id-ga – ainult selle kaudu saab määrata teisi atribuute. Kasutajate isikukoodide kaudu saab määrata, mis kasutajanime, esnime ja perenimega on tegemist.

Päringud

1. Esimene paring on “isik_sissetulekud”. Kui kasutaja sisestab oma kasutajanime ID, siis saab ta kokkuvõtte oma sissetulekutest, kus on eraldi toodud välja sissetulekud allikate kaupa.

```
CREATE PROCEDURE "OSP"."isik_sissetulekud"( @a_id INTEGER )
AS
BEGIN
    SELECT "eesnimi","perenimi","allikas","summa","valuuta" FROM
    "isikuandmed","sissetulekud"
        JOIN "sissetulekuallikad" ON "sissetulekuallikad"."id" = "sissetulekud"."id"
        WHERE "isikuandmed"."kasutaja_id" = @a_id
        GROUP BY "eesnimi","perenimi","allikas","summa","valuuta"
        ORDER BY "allikas" ASC
END
```

```
CALL isik_sissetulekud (2);
```

```
eesnimi;perenimi;allikas;summa;valuuta
```

```
'Tõnu';'Mõgi';'muu';20;'EUR'
```

```
'Tõnu';'Mõgi';'palk';1200;'EUR'
```

```
'Tõnu';'Mõgi';'renditulu';400;'EUR'
```

```
'Tõnu';'Mõgi';'toetus';1600;'EUR'
```

2. Teine paring on “p_kokkuvõte”, kus antakse kasutaja ID sisestamisel kokkuvõtte sissetulekutest ja väljaminekutest. Saadud tulemus on summeeritud kokku.

```

CREATE PROCEDURE "OSP"."p_kokkuvote"( IN "a_kasutaja" INTEGER )
BEGIN
    SELECT "kasutajanimi", "SUM"("sissetulekud"."summa") AS
"sissetulekud", "SUM"("valjaminekud"."summa") AS "väljaminekud"
    FROM "kasutajad", "sissetulekud", "valjaminekud"
    WHERE "kasutajad"."id" = "a_kasutaja" AND "kasutajad"."id" =
"sissetulekud"."kasutaja_st"
    GROUP BY "kasutajanimi"
END

```

```
CALL p_kokkuvote (4);
```

```
kasutajanimi;sissetulekud;väljaminekud
```

```
'Kiisumiisu';13500;7066
```

3. Kolmas paring on “p_lisa_andmed”, mis lubab kasutajal lisada uue kasutaja andmebaasi. Lisaja peab valima kasutajanime ja lisama ka enda (või kellegi teise) isikuandmed.

```

CREATE PROCEDURE "OSP"."p_lisa_andmed"(
    IN "p_kasutajanimi" VARCHAR(50),
    IN "p_isikukood" VARCHAR(11),
    IN "p_eesnimi" VARCHAR(50),
    IN "p_perenimi" VARCHAR(50) )
BEGIN
    DECLARE "k_id" INTEGER;
    INSERT INTO "kasutajad"( "kasutajanimi" ) VALUES( "p_kasutajanimi" ) ;

```

```

INSERT INTO "isikuandmed"( "isikukood","eesnimi","perenimi" ) VALUES(
"p_isikukood","p_eesnimi","p_perenimi" );
SELECT @@identity INTO "k_id";
SET "kasutajad"."id" = "k_id"
END

```

CALL p_lisa_andmed ('VanaemaMai', '43322990876', 'Mai', 'Mägi');

Andmebaasis on nüüd ka VanaemaMai ID-ga 18 lisatud:

id,kasutajanimi

1,'Emme'

2,'Issi'

3,'Batman'

4,'Kiisumiisu'

5,'Tommi'

18,'VanaemaMai'

4. Neljas paring on “p_sissetulekud_kp”, kus kasutaja peab sisestama alguskuupäeva ja lõppkuupäeva, mille kohta ta tahab sissetulekute vahearuannet saada. Saadud tulemus on andmebaasi ülene.

```

CREATE PROCEDURE "OSP"."p_sissetulekud_kp"( IN "alguskp" DATE,IN "loppkp"
DATE )
BEGIN
SELECT "allikas","summa","valuuta" FROM "sissetulekud"
JOIN "sissetulekuallikad" ON "sissetulekuallikad"."id" =
"sissetulekud"."sissetulekuallika_id"
WHERE "sissetulekud"."kuupaev" BETWEEN "alguskp" AND "loppkp"

```

```

GROUP BY "allikas","summa","valuuta"
ORDER BY "allikas" ASC
END

```

```
CALL p_sissetulekud_kp ('2019-10-11', '2019-12-24');
```

```
allikas;summa;valuuta
```

```
'muu';20;'EUR'
```

```
'muu';50;'EUR'
```

```
'muu';100;'EUR'
```

```
'palk';1600;'EUR'
```

```
'palk';1400;'EUR'
```

```
'palk';1200;'EUR'
```

```
'renditulu';400;'EUR'
```

```
'toetus';300;'EUR'
```

5. Viies paring on “p_valjaminekud_kp”, kus kasutaja peab sisestama alguskuupäeva ja lõppkuupäeva, mille kohta tahab väljaminekute vahearuannet saada. Saadud tulemus on andmebaasi ülene. Näiteks kui ema tahab saada ülevaadet oma perekonna kulutuste kohta.

```

CREATE PROCEDURE "OSP"."p_valjaminekud_kp"( IN "alguskp" DATE,IN "loppkp"
DATE )
BEGIN
SELECT "nimi","summa","valuuta" FROM "valjaminekud"
JOIN "kulutuubid" ON "kulutuubid"."id" = "valjaminekud"."kulutuup"
WHERE "valjaminekud"."kuupaev" BETWEEN "alguskp" AND "loppkp"
GROUP BY "nimi","summa","valuuta"

```

END

CALL p_valjaminekud_kp ('2019-10-11', '2019-12-24');

nimi;summa;valuuta

'toit';88;'EUR'

'riided';300;'EUR'

'toit';50;'EUR'

'kommunaalid';100;'EUR'

'majapidamine';100;'EUR'

'toit';20;'EUR'

'hobid';5;'EUR'

'koolitarbed';80;'EUR'

'hobid';10;'EUR'

'sport';10;'EUR'

'toit';200;'EUR'

'lemmikloomad';100;'EUR'

'koolitarbed';50;'EUR'

'k¹/₄tus';100;'EUR'

'riided';50;'EUR'

'kommunaalid';200;'EUR'

'lemmikloomad';50;'EUR'

'k¹/₄tus';50;'EUR'

'riided';30;'EUR'

'perioodika';20;'EUR'

'perioodika';30;'EUR'

'hobid';100;'EUR'

'laen';800;'EUR'