

# Videospēļu platforma

Markuss Birznieks, mb20075

## Biznesa pārskats

Ar vien mazāk cilvēku vēlas pirkt fiziskas kopijas videospēlēm, tā vietā izvēlas ērtāku risinājumu pirkt un lejupielādēt spēles digitāla izplatītāja platformā. Videospēļu spēlētāji vēlas vienuviet gan iegādāties spēles, gan arī komunicēt ar citiem spēles spēlētājiem, tāpēc platformai jādarbojas gan kā interneta veikalam, gan kā sociālajam mēdijam. Šādai platformai ir nepieciešama datu bāze, kurā tiek turēti dati par videospēlēm, pirkumiem, lietotājiem un citām aktivitātēm.

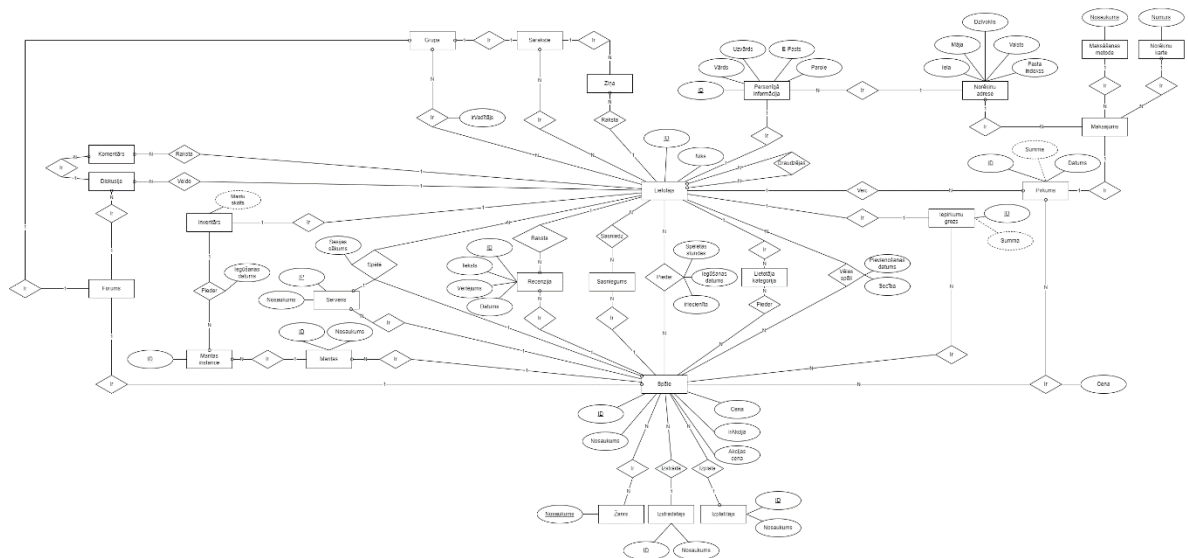
Videospēļu platforma ļauj tās lietotājiem iegādāties un lejupielādēt spēles, pievienojot tās virtuālajai bibliotēkai, no kuras tās var lejupielādēt un instalēt. Lietotāji šādā platformā jaspēj sazināties ar citiem spēlētājiem, pievienoties grupām un piedalīties tiešsaistes kopienās. Veikalā lietotāji var meklēt spēles pēc dažādiem parametriem kā nosaukums, cena, ražotājs, žanrs. Katram lietotājam arī ir savs vēlamu spēļu saraksts. Pēc spēles iegādes spēle tiek pievienota lietotāja virtuālajai bibliotēkai, kurā spēles var kārtot pēc nosaukuma, žanra, nospēlētajām stundām un var atzīmēt mīļākās spēles.

## Biznesa prasības un noteikumi

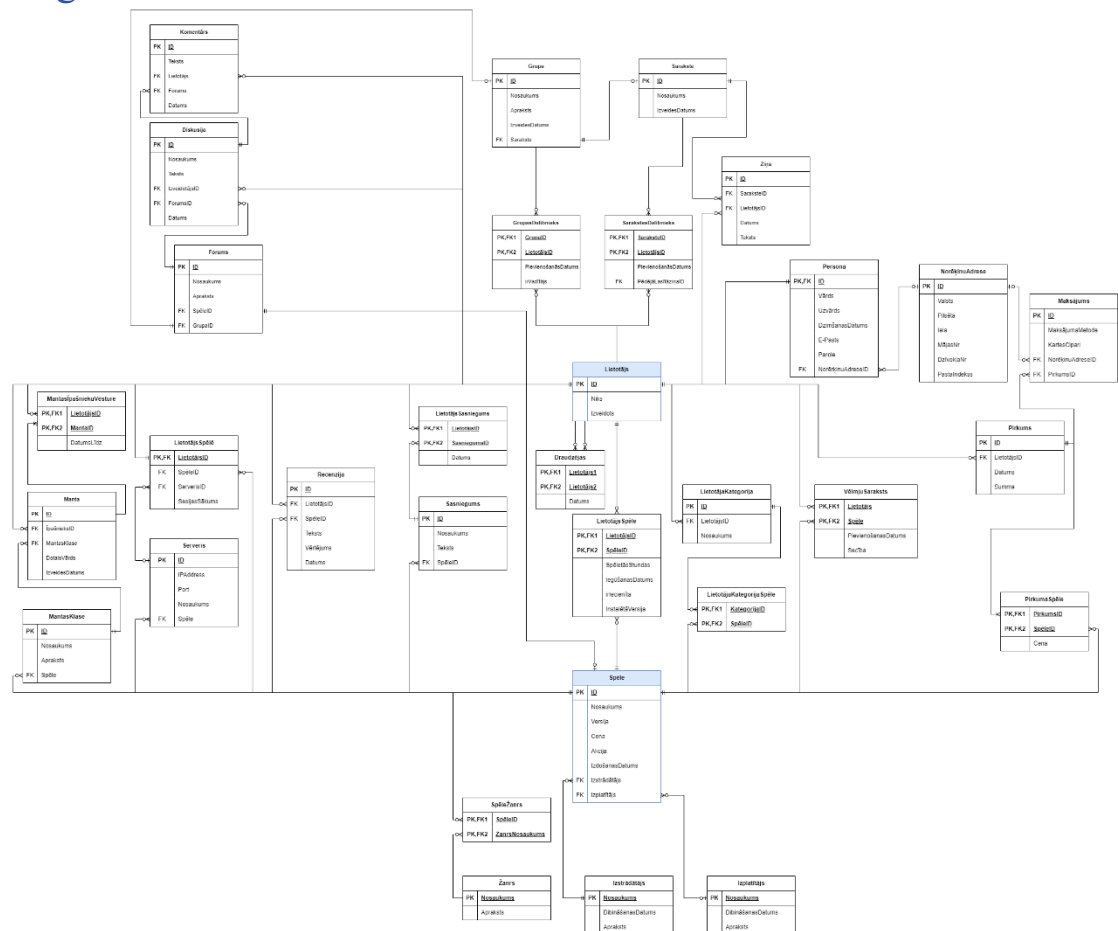
- Lietotāji var iegādāties videospēles, pēc iegādes tās tiek pievienotas lietotāja virtuālajai bibliotēkai, no kuras lietotājs var spēles lejupielādēt neierobežota skaita reizi.
- Datubāzē būs iekļauta informācija par videospēlēm, tostarp to nosaukumi, izdevēji, žanri, izdošanas datumi, vērtējumi un recenzijas.
- Datubāzē tiek saglabāta arī informācija par lietotāju veiktajiem pirkumiem, tostarp par iegādātajām spēlēm, pirkuma datumu un samaksāto summu.
- Veikalā ir iespējams meklēt spēles pēc nosaukuma, izdevēja, žanra un izdošanas datuma, kā arī filtrēt meklēšanas rezultātus pēc dažādiem kritērijiem, tostarp cenas, lietotāju vērtējuma un sistēmas prasībām.
- Lietotājs var kārtot savu bibliotēku pēc tādiem pašiem kritērijiem kā veikalā, tostarp pēc nospēlētajām stundām.
- Virtuālajā bibliotēkā var atzīmēt mīļākās spēles.
- Lietotāji varēs arī atstāt atsauksmes un vērtējumus par spēlēm, kas būs redzami citiem lietotājiem.
- Lietotājs var atstāt atsauksmi un vērtējumu tikai spēlēm kuras lietotājs ir iegādājies un katrai spēlei tikai vienu vērtējumu un atsauksmi.
- Lietotāji varēs novērtēt spēles skalā no 1 līdz 10, kur 1 ir zemākais vērtējums, bet 10 - augstākais.
- Katram lietotājam ir savs vēlmju saraksts kurā ir lietotāja secība sakārtots kuras spēles lietotājs vēlas iegādāties.
- Lietotājiem tiek nodrošināti atjauninājumi spēlēm.
- Lietotāji šādā platformā jaspēj sazināties ar citiem spēlētājiem, pievienoties grupām un piedalīties tiešsaistes kopienās.

- Divi lietotāji var sarakstīties ar viens otru tikai ja tie ir draugi.
- Datubāzē tiek glabāta arī lietotāja informācija, tostarp lietotājvārds, parole, e-pasta adrese.
- Platforma ļaus lietotājiem jebkurā laikā dzēst savus kontus, kā rezultātā no datubāzes tiks dzēsti visi saistītie dati.
- Lietotājs spēlējot spēles spēj nopelnīt sasniegumus(achievements) attiecīgajai spēlei.
- Dažām spēlēm ir savas mantas, kuras spēlētājs var iegūt spēlējot spēli. Lietotājs šīs mantas var apskatīt savā inventārā.
- Lietotāji var apskatīt citu lietotāju bibliotēku un inventāru.

# Konceptuālais modelis



# Loģiskais modelis



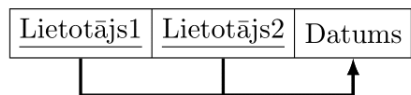
# Funkcionālo atkarību grafi

Visas tabulas ir vismaz 3NF.

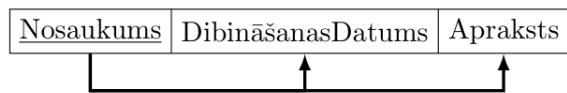
## Lietotājs



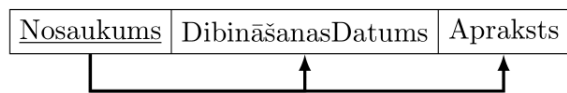
## Draudzējas



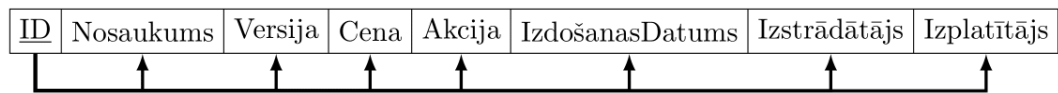
## Izstrādātājs



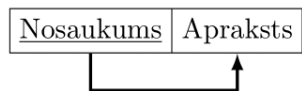
## Izplatītājs



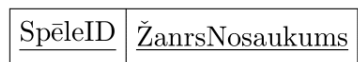
## Spēle



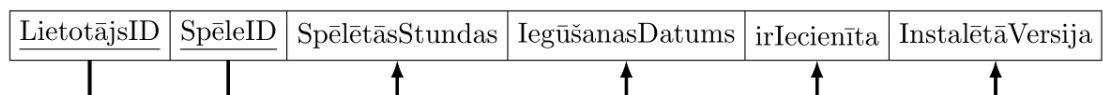
## Žanrs



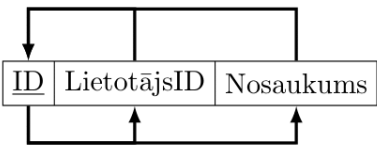
## SpēleŽanrs



## LietotājsSpēle



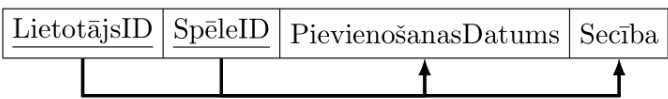
**LietotājaKategorija**



**LietotājaKategorijaSpēle**



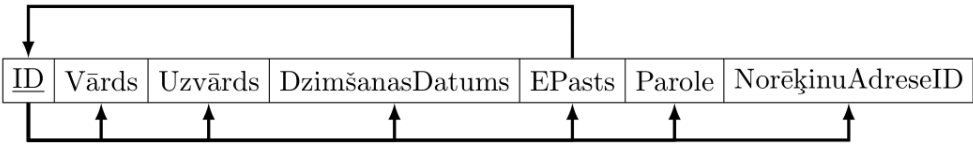
**VēlmjuSaraksts**



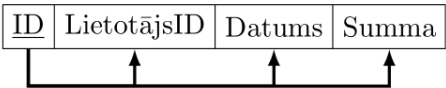
**NorēķinuAdrese**



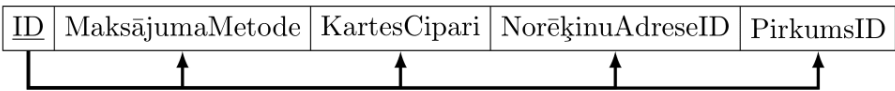
**Persona**



**Pirkums**



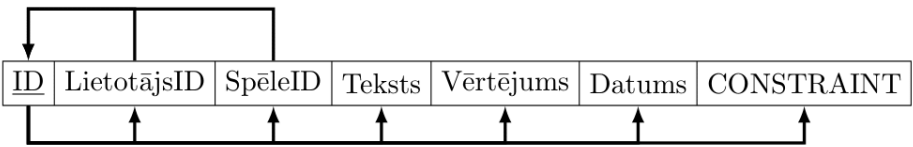
**Maksājums**



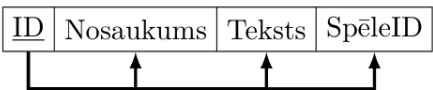
**PirkumaSpēle**



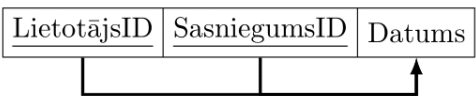
**Recenzija**



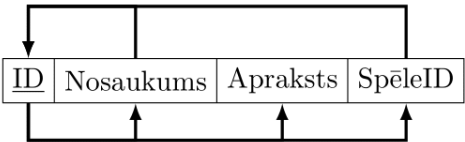
**Sasniegums**



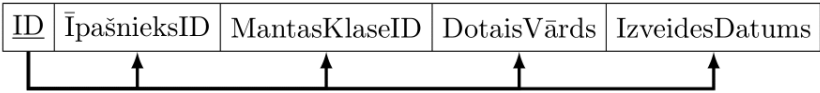
**LietotājsSasniegums**



**MantasKlase**



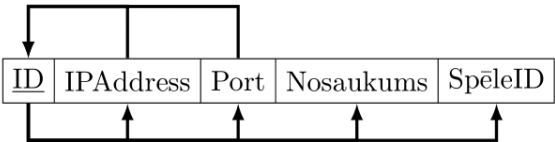
**Manta**



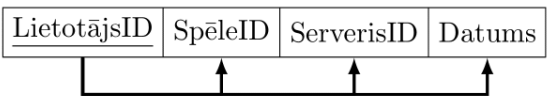
**MantasĪpašniekuVēsture**



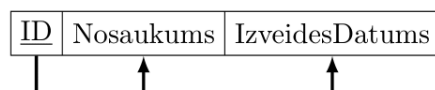
**Serveris**



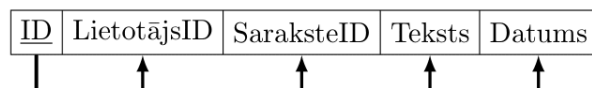
**LietotājsSpēlē**



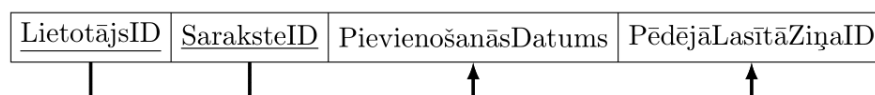
### Sarakste



### Ziņa



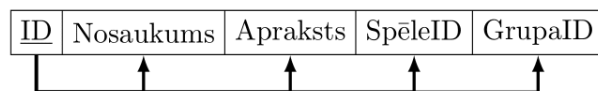
### SarakstesDalībnieks



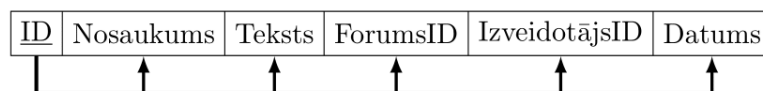
### Grupa



### Forums



### Diskusija



### Komentārs



# Skripti

## Tabulu izveidošana

```
CREATE TABLE Lietotājs (  
    ID int IDENTITY(1000,1) PRIMARY KEY,  
    Niks nvarchar(50) NOT NULL,  
    Izveidots date DEFAULT getUTCDate()  
);  
  
CREATE TABLE Draudzējas (  
    Lietotājs1 int NOT NULL FOREIGN KEY REFERENCES Lietotājs(ID),  
    Lietotājs2 int NOT NULL FOREIGN KEY REFERENCES Lietotājs(ID),  
    Datums date  
    PRIMARY KEY (Lietotājs1, Lietotājs2)  
);  
  
CREATE TABLE Izstrādātājs (  
    Nosaukums nvarchar(50) NOT NULL PRIMARY KEY,  
    DibināšanasDatums date DEFAULT getUTCDate(),  
    Apraksts nvarchar(MAX)  
);  
  
CREATE TABLE Izplatītājs (  
    Nosaukums nvarchar(50) NOT NULL PRIMARY KEY,  
    DibināšanasDatums date DEFAULT getUTCDate(),  
    Apraksts nvarchar(MAX)  
);  
  
CREATE TABLE Spēle (  
    ID int IDENTITY(1,1) PRIMARY KEY,  
    Nosaukums nvarchar(100) NOT NULL,  
    Versija nvarchar(50) DEFAULT '1.0.0',  
    Cena smallmoney NOT NULL,  
    Akcija int DEFAULT 0,  
    IzdošanasDatums date DEFAULT getUTCDate(),  
    Izstrādātājs nvarchar(50) NOT NULL FOREIGN KEY REFERENCES  
Izstrādātājs(Nosaukums),  
    Izplatītājs nvarchar(50) NOT NULL FOREIGN KEY REFERENCES  
Izplatītājs(Nosaukums),  
);  
  
CREATE TABLE Žanrs (  
    Nosaukums nvarchar(50) NOT NULL PRIMARY KEY,  
    Apraksts nvarchar(MAX)  
);  
  
CREATE TABLE SpēleŽanrs (  
    SpēleID int NOT NULL FOREIGN KEY REFERENCES Spēle(ID),
```



```

        ŽanrsNosaukums nvarchar(50) NOT NULL FOREIGN KEY REFERENCES
Žanrs(Nosaukums),
        PRIMARY KEY (SpēleID, ŽanrsNosaukums)
    );

CREATE TABLE LietotājsSpēle (
    LietotājsID int NOT NULL FOREIGN KEY REFERENCES Lietotājs(ID),
    SpēleID int NOT NULL FOREIGN KEY REFERENCES Spēle(ID),
    SpēlētāsStundas int DEFAULT 0,
    IegūšanasDatums date DEFAULT getUTCDate(),
    irIecienīta bit DEFAULT 0,
    InstalētāVersija nvarchar(50),
    PRIMARY KEY (LietotājsID, SpēleID)
);

CREATE TABLE LietotājaKategorija (
    ID int IDENTITY(1,1) PRIMARY KEY,
    LietotājsID int NOT NULL FOREIGN KEY REFERENCES Lietotājs(ID),
    Nosaukums nvarchar(50) NOT NULL,
    UNIQUE (LietotājsID, Nosaukums)
);

CREATE TABLE LietotājaKategorijaSpēle (
    LietotājsKategorijaID int NOT NULL FOREIGN KEY REFERENCES
LietotājaKategorija(ID),
    SpēleID int NOT NULL FOREIGN KEY REFERENCES Spēle(ID),
    PRIMARY KEY (LietotājsKategorijaID, SpēleID)
);

CREATE TABLE VēlmjuSaraksts (
    LietotājsID int NOT NULL FOREIGN KEY REFERENCES Lietotājs(ID),
    SpēleID int NOT NULL FOREIGN KEY REFERENCES Spēle(ID),
    PievienošanasDatums smalldatetime DEFAULT getUTCDate(),
    Secība int NOT NULL,
    PRIMARY KEY (LietotājsID, SpēleID)
);

CREATE TABLE NorēķinuAdrese (
    ID int IDENTITY(1,1) PRIMARY KEY,
    Valsts nvarchar(50) NOT NULL,
    Pilsēta nvarchar(50) NOT NULL,
    Iela nvarchar(50) NOT NULL,
    MājasNr nvarchar(10) NOT NULL,
    DzīvokļaNr nvarchar(10) NOT NULL,
    PastaIndekss nvarchar(10) NOT NULL
);

CREATE TABLE Persona (

```

```

        ID int PRIMARY KEY FOREIGN KEY REFERENCES Lietotājs(ID),
        Vārds nvarchar(50) NOT NULL,
        Uzvārds nvarchar(50) NOT NULL,
        DzimšanasDatums date NOT NULL,
        EPasts nvarchar(100) NOT NULL UNIQUE,
        Parole nvarchar(100) NOT NULL,
        NorēķinuAdreseID int NOT NULL FOREIGN KEY REFERENCES
NorēķinuAdrese(ID)
    );

CREATE TABLE Pirkums (
    ID int IDENTITY(1,1) PRIMARY KEY,
    LietotājsID int NOT NULL FOREIGN KEY REFERENCES Lietotājs(ID),
    Datums smalldatetime DEFAULT getUTCDate(),
    Summa smallmoney NOT NULL
);

CREATE TABLE Maksājums (
    ID int IDENTITY(1,1) PRIMARY KEY,
    MaksājumaMetode nvarchar(50) NOT NULL,
    KartesCipari char(4),
    NorēķinuAdreseID int NOT NULL FOREIGN KEY REFERENCES
NorēķinuAdrese(ID),
    PirkumsID int NOT NULL FOREIGN KEY REFERENCES Pirkums(ID)
);

CREATE TABLE PirkumaSpēle (
    PirkumsID int NOT NULL FOREIGN KEY REFERENCES Pirkums(ID),
    SpēleID int NOT NULL FOREIGN KEY REFERENCES Spēle(ID),
    Cena smallmoney NOT NULL,
    PRIMARY KEY (PirkumsID, SpēleID)
);

CREATE TABLE Recenzija (
    ID int IDENTITY(1,1) PRIMARY KEY,
    LietotājsID int NOT NULL FOREIGN KEY REFERENCES Lietotājs(ID),
    SpēleID int NOT NULL FOREIGN KEY REFERENCES Spēle(ID),
    Teksts nvarchar(MAX),
    Vērtējums int NOT NULL,
    Datums smalldatetime DEFAULT getUTCDate(),
    CONSTRAINT CK_Recenzija_Vērtējums CHECK (Vērtējums >= 1 AND
Vērtējums <= 10),
    UNIQUE (LietotājsID, SpēleID)
);

CREATE TABLE Sasniegums (
    ID int IDENTITY(1,1) PRIMARY KEY,
    Nosaukums nvarchar(50) NOT NULL,

```

```

        Teksts nvarchar(255),
        SpēleID int NOT NULL FOREIGN KEY REFERENCES Spēle(ID)
    );

CREATE TABLE LietotājsSasniegums (
    LietotājsID int NOT NULL FOREIGN KEY REFERENCES Lietotājs(ID),
    SasniegumsID int NOT NULL FOREIGN KEY REFERENCES Sasniegums(ID),
    Datums smalldatetime DEFAULT getUTCDate(),
    PRIMARY KEY (LietotājsID, SasniegumsID)
);

CREATE TABLE MantasKlase (
    ID int IDENTITY(1,1) PRIMARY KEY,
    Nosaukums nvarchar(50) NOT NULL,
    Apraksts nvarchar(255),
    SpēleID int NOT NULL FOREIGN KEY REFERENCES Spēle(ID),
    UNIQUE (SpēleID, Nosaukums)
);

CREATE TABLE Manta (
    ID int IDENTITY(1,1) PRIMARY KEY,
    ĪpašnieksID int NOT NULL FOREIGN KEY REFERENCES Lietotājs(ID),
    MantasKlaseID int NOT NULL FOREIGN KEY REFERENCES MantasKlase(ID),
    DotaisVārds nvarchar(50),
    IzveidesDatums smalldatetime NOT NULL DEFAULT getUTCDate()
);

CREATE TABLE MantasĪpašniekuVēsture (
    LietotājsID int NOT NULL FOREIGN KEY REFERENCES Lietotājs(ID),
    MantaID int NOT NULL FOREIGN KEY REFERENCES Manta(ID),
    DatumsLīdz smalldatetime NOT NULL DEFAULT getUTCDate(),
    PRIMARY KEY (MantaID, LietotājsID)
);

CREATE TABLE Serveris (
    ID int IDENTITY(1,1) PRIMARY KEY,
    IPAddress nvarchar(15) NOT NULL,
    Port int NOT NULL,
    Nosaukums nvarchar(50) NOT NULL,
    SpēleID int NOT NULL FOREIGN KEY REFERENCES Spēle(ID),
    UNIQUE (IPAddress, Port)
);

CREATE TABLE LietotājsSpēlē (
    LietotājsID int PRIMARY KEY FOREIGN KEY REFERENCES Lietotājs(ID),
    SpēleID int NOT NULL FOREIGN KEY REFERENCES Spēle(ID),
    ServerisID int FOREIGN KEY REFERENCES Serveris(ID),
    Datums smalldatetime DEFAULT getUTCDate()
);

```

```

);

CREATE TABLE Sarakste (
    ID int IDENTITY(1,1) PRIMARY KEY,
    Nosaukums nvarchar(50) NOT NULL,
    IzveidesDatums smalldatetime DEFAULT getUTCDate()
);

CREATE TABLE Ziņa (
    ID int IDENTITY(1,1) PRIMARY KEY,
    LietotājsID int NOT NULL FOREIGN KEY REFERENCES Lietotājs(ID),
    SaraksteID int NOT NULL FOREIGN KEY REFERENCES Sarakste(ID),
    Teksts nvarchar(MAX) NOT NULL,
    Datums smalldatetime DEFAULT getUTCDate()
);

CREATE TABLE SarakstesDalībnieks (
    LietotājsID int NOT NULL FOREIGN KEY REFERENCES Lietotājs(ID),
    SaraksteID int NOT NULL FOREIGN KEY REFERENCES Sarakste(ID),
    PievienošanāsDatums smalldatetime DEFAULT getUTCDate(),
    PēdējāLasītāZiņaID int FOREIGN KEY REFERENCES Ziņa(ID),
    PRIMARY KEY (LietotājsID, SaraksteID)
);

CREATE TABLE Grupa (
    ID int IDENTITY(1,1) PRIMARY KEY,
    Nosaukums nvarchar(50) NOT NULL,
    Apraksts nvarchar(255),
    IzveidesDatums smalldatetime DEFAULT getUTCDate(),
    SaraksteID int FOREIGN KEY REFERENCES Sarakste(ID)
);

CREATE TABLE Forums (
    ID int IDENTITY(1,1) PRIMARY KEY,
    Nosaukums nvarchar(50) NOT NULL,
    Apraksts nvarchar(255) NOT NULL,
    SpēleID int FOREIGN KEY REFERENCES Spēle(ID),
    GrupaID int FOREIGN KEY REFERENCES Grupa(ID)
);

CREATE TABLE Diskusija (
    ID int IDENTITY(1,1) PRIMARY KEY,
    Nosaukums nvarchar(50) NOT NULL,
    Teksts nvarchar(MAX) NOT NULL,
    ForumsID int NOT NULL FOREIGN KEY REFERENCES Forums(ID),
    IzveidotājsID int NOT NULL FOREIGN KEY REFERENCES Lietotājs(ID),
    Datums smalldatetime DEFAULT getUTCDate()
);

```

```
CREATE TABLE Komentārs (
    ID int IDENTITY(1,1) PRIMARY KEY,
    Teksts nvarchar(MAX) NOT NULL,
    DiskusijaID int NOT NULL FOREIGN KEY REFERENCES Diskusija(ID),
    LietotājsID int NOT NULL FOREIGN KEY REFERENCES Lietotājs(ID),
    Datums smalldatetime DEFAULT getUTCDate()
);
```

### Tabulu dzēšana

```
DROP TABLE Komentārs;
DROP TABLE Diskusija;
DROP TABLE Forums;
DROP TABLE Grupa;

DROP TABLE SarakstesDalībnieks;
DROP TABLE Ziņa;
DROP TABLE Sarakste;

DROP TABLE LietotājsSpēlē;
DROP TABLE Serveris;

DROP TABLE MantasĪpašniekuVēsture;
DROP TABLE Manta;
DROP TABLE MantasKlase;

DROP TABLE LietotājsSasniegums;
DROP TABLE Sasniegums;

DROP TABLE Recenzija;

DROP TABLE PirkumaSpēle;
DROP TABLE Maksājums;
DROP TABLE Pirkums;
DROP TABLE Persona;
DROP TABLE NorēķinuAdrese;

DROP TABLE VēlmjuSaraksts;

DROP TABLE LietotājaKategorijaSpēle;
DROP TABLE LietotājaKategorija;

DROP TABLE LietotājsSpēle;

DROP TABLE SpēleŽanrs;
DROP TABLE Žanrs;
DROP TABLE Spēle;
DROP TABLE Izplatītājs;
DROP TABLE Izstrādātājs;

DROP TABLE Draudzējas;
DROP TABLE Lietotājs;
```

### Datu ievietošana

Tikai piemēri, pilnie insert skripti atrodami failā insert.sql

Ievieto žanrus

```
INSERT INTO Žanrs VALUES
    ('Asa sižeta', 'Strauji ritējošas spēles, kas prasa ātrus refleksus un
    stratēģisko domāšanu.'),
```

```

('Piedzīvojumu', 'Spēles, kas koncentrējas uz izpēti, stāstu un mīklu
risināšanu.'),
('Casual', 'Viegli spēlējamās spēles, kas piemērotas plašam auditorijam.'),
('Indie', 'Spēles, ko izstrādā mazas neatkarīgas izstrādātājfirmas vai
studijas.'),
...

```

#### Ievieto izstrādātājus

```
INSERT INTO Izstrādātājs VALUES
```

```

('Valve', '1996-08-24', 'Amerikāņu videospēļu izstrādātāju kompānija, kas
izveidota 1996. gadā.'),
('Rockstar Games', '1998-12-01', 'Amerikāņu videospēļu izstrādātāju
kompānija, kas izveidota 1998. gadā.'),
('Ubisoft', '1986-03-12', 'Francijas videospēļu izstrādātāju kompānija, kas
izveidota 1986. gadā.'),
...

```

#### Ievieto izplatītājus

```
INSERT INTO Izplatītājs VALUES
```

```

('Electronic Arts', '1982-05-27', 'Amerikāņu videospēļu izdevējs un
distribūtors, kas ir pazīstams ar spēlēm, piemēram, FIFA, Need for Speed, Mass
Effect un Battlefield.'),
('Ubisoft', '1986-03-12', 'Francijas videospēļu izdevējs un distribūtors,
kas izstrādā un izplatā spēles, piemēram, Assassin's Creed, Far Cry, Just
Dance un Tom Clancy's sērijas.'),
('Activision', '1979-10-01', 'Amerikāņu videospēļu izdevējs un
distribūtors, pazīstams ar spēlēm, piemēram, Call of Duty, Skylanders, Destiny
un Tony Hawk's Pro Skater.'),
...

```

#### Ievieto spēles

```
INSERT INTO Spēle (Nosaukums, Versija, Cena, Akcija, IzdošanasDatums,
Izstrādātājs, Izplatītājs) VALUES
```

```

('Counter-Strike', '1.6.5', 9.99, 0, '2000-11-01', 'Valve', 'Valve'),
('Team Fortress Classic', '2.8.7', 4.99, 0, '1999-04-01', 'Valve',
'Valve'),
('Ricochet', '1.0.0', 4.99, 0, '2000-11-01', 'Valve', 'Valve'),
...

```

#### Pievieno spēlēm žanrus

```
INSERT INTO SpēleŽanrs (SpēleID, ŽanrsNosaukums) VALUES
```

```

(1, 'Asa sižeta'),
(1, 'FPS'),
(2, 'Co-op'),
(4, 'FPS'),
...

```

#### Ievieto lietotājus

```
INSERT INTO Lietotājs (Niks, Izveidots) VALUES
```

```

('Lāčplēsis', '2014-07-25'),
('RīgasRaiders', '2012-03-15'),
('BaltijasBlasters', '2015-06-20'),
...

```

#### Ievieto lietotāju draudzības

```
INSERT INTO Draudzējas (Lietotājs1, Lietotājs2, Datums) VALUES
```

```
(1003, 1068, '2019-08-28'),  
(1068, 1003, '2019-08-28'),  
(1062, 1080, '2022-07-15'),
```

...

Ievieto lietotāju spēles

```
INSERT INTO LietotājsSpēle VALUES  
(1001, 105, 1487, '2012-08-08', 0, '1.0.0'),  
(1001, 28, 1101, '2010-03-18', 0, '1.0.0'),  
(1001, 64, 962, '2012-12-30', 1, '1.0.0'),
```

...

Ievieto lietotāja definētas kategorijas

```
INSERT INTO LietotājaKategorija VALUES  
(1001, 'Galvaslauzīšana'),  
(1003, 'Vecie Labie'),  
(1003, 'Neparastie Pērles'),
```

...

Ievieto spēles lietotāju definētās kategorijās

```
INSERT INTO LietotājaKategorijaSpēle VALUES  
(1, 47),  
(1, 109),  
(1, 56),  
(4, 90),  
(4, 110),  
(4, 53),  
(3, 112),
```

...

Ievieto spēles lietotāju vēlmju sarakstā

```
INSERT INTO VēlmjuSaraksts VALUES  
(1003, 67, '2018-08-01', 1),  
(1003, 85, '2014-05-03', 2),  
(1004, 55, '2021-04-13', 1),  
(1004, 46, '2019-07-04', 2),  
(1005, 58, '2021-01-11', 1),  
(1005, 105, '2016-08-21', 2),
```

...

Ievieto spēlēm sasniegumus

```
INSERT INTO Sasniegums (Nosaukums, Teksts, SpēleID) VALUES  
( 'First Blood', ' Get the first kill in a round.', 12),  
( 'Dominating', ' Kill a player four times without them killing you.',  
12),  
( 'Hat Trick', ' Capture three capture points in a single life.', 12),
```

...

Ievieto lietotāju sasniegtos sasniegumus

```
INSERT INTO LietotājsSasniegums (LietotājsID, SasniegumsID, Datums) VALUES  
(1001, 20, '2016-12-31'),  
(1001, 13, '2022-11-03'),  
(1001, 53, '2019-01-07'),  
(1001, 55, '2023-05-09'),  
(1002, 5, '2022-12-31'),
```

...

Ievieto norēķinu adreses

```
INSERT INTO NorēķinuAdrese (Valsts, Pilsēta, Iela, MājasNr, DzīvokļaNr,
PastaIndekss) VALUES
('Latvija', 'Rīga', 'Ēvalds Caurceļs', '2864', '5', 'LV-1000'),
('Latvija', 'Liepāja', 'Krišjānis Curve', '079', '90', 'LV-3401'),
('Latvija', 'Daugavpils', 'Sandra Rapid', '9495', '79', 'LV-5401'),
```

...

Ievieto lietotāju personas

```
INSERT INTO Persona (ID, Vārds, Uzvārds, DzimšanasDatums, EPasts, Parole,
NorēķinuAdreseID) VALUES
(1000, 'Toms', 'Vilks', '1994-07-12', 'caunemaza@example.org',
'UsCdode5#9', 28),
(1001, 'Rita', 'Siliņš', '1999-01-24', 'sprogisvalentina@example.com',
'_6CUzxc($R', 4),
(1002, 'Augusts', 'Bite', '1966-09-03', 'rvilks@example.net',
'4N(Neh_R&z', 12),
```

...

Ievieto lietotāju pirkumus

```
INSERT INTO Pirkums (LietotājsID, Datums, Summa) VALUES
(1001, '2007-11-27', 2.99),
(1001, '2010-03-18', 19.99),
(1001, '2011-12-11', 39.97),
```

...

Ievieto pirkumu spēles

```
INSERT INTO PirkumaSpēle (PirkumsID, SpēleID, Cena) VALUES
(1, 27, 2.99),
(2, 47, 19.99),
(3, 24, 9.99),
(3, 33, 19.99),
```

...

Ievieto maksājumus par parikumiem

```
INSERT INTO Maksājums (MaksājumaMetode, KartesCipari, NorēķinuAdreseID,
PirkumsID) VALUES
('PayPal', NULL, 4, 1),
('MasterCard', '6908', 4, 2),
('Visa', '4988', 4, 3),
('Visa', '9506', 4, 4),
```

...

Ievieto recenzijas

```
INSERT INTO Recenzija (LietotājsID, SpēleID, Teksts, Vērtējums, Datums) VALUES
(1001, 12, 'Spēle ir interesanta, bet ir daži tehniski jautājumi, kas
jāatrisina.', 8, '2007-12-09'),
(1001, 17, NULL, 6, '2012-07-19'),
(1001, 24, 'Šī spēle ir absolūti brīnišķīga. Es nevaru pārtraukt to
spēlēt.', 9, '2011-12-14'),
(1001, 33, NULL, 2, '2012-02-10'),
```

...

Ievieto spēļu serverus

```
INSERT INTO Serveris (IPAddress, Port, Nosaukums, SpēleID) VALUES
('34.111.182.81', 8324, 'Burvju Karalienes Cīņas', 1),
('153.1.29.217', 1927, 'Lāčplēša Vētras Pirātu', 2),
```



```
( '122.156.201.36', 5461, 'Lielās Pilsētas Pirātu', 2),
( '219.216.244.237', 3256, 'Melnās Ziemeļu Izdzīvošana', 2),
( '87.222.251.246', 9071, 'Senču Burvju Pirātu', 2),
( '98.65.181.192', 7883, 'Leģendārās Pilsētas Cīņas', 3),
```

...

Ievieto lietotāju pašreizējo spēles aktivitāti

```
INSERT INTO LietotājsSpēle (LietotājsID, SpēleID, ServerisID, Datums) VALUES
(1001, 12, NULL, NULL),
...
(1009, 56, 140, DATEADD(hour, -6, getUTCDate())),
...
(1021, 112, NULL, DATEADD(hour, -4, getUTCDate())),
...
```

Ievieto mantas spēlēm

```
INSERT INTO MantasKlase (Nosaukums, Apraksts, SpēleID) VALUES
( 'Dvēseļu Lāčplēša Krekls', NULL, 36),
( 'Dzelzs', 'Dzelzs ir būtiska materiāla sastāvdaļa daudzu ieroču un
bruņu ražošanā.', 36),
( 'Akadēmijas Mistērijas Jaka', NULL, 36),
( 'Zelta Lāčplēša Izdzīvošanas Iešana', NULL, 36),
...
```

Ievieto mantas instances

```
INSERT INTO Manta (ĪpašnieksID, MantasKlaseID, DotaisVārds, IzveidesDatums)
VALUES
(1001, 183, 'Gudrības Pulveris', '2022-09-14 13:10:21'),
(1001, 180, 'Ierocis Lāse', '2021-11-18 18:33:26'),
(1001, 178, NULL, '2012-10-18 10:01:21'),
(1001, 176, NULL, '2018-07-02 12:56:37'),
...
```

Ievieto mantām īpašnieku vēsturi

```
INSERT INTO MantasĪpašniekuVēsture (LietotājsID, MantaID, DatumsLīdz) VALUES
(1049, 4, '2019-03-23 20:53:52'),
(1039, 5, '2023-05-25 16:00:29'),
(1014, 6, '2023-05-25 03:56:35'),
...
```

## SELECT

Visi selecti pieejami failā select.sql. Šeit esmu atlasījis, kas man liekas interesantākie.

```
-- Spēļu skaits lietotāja definētā kategorijā
SELECT LietotājsID, Niks, LietotājaKategorija.ID as KategorijasID,
LietotājaKategorija.Nosaukums, COUNT(*) as SpēļuSkaits
FROM LietotājaKategorija
JOIN LietotājaKategorijaSpēle on LietotājaKategorijaSpēle.LietotājsKategorijaID
= LietotājaKategorija.ID
JOIN Lietotājs on Lietotājs.ID = LietotājsID
GROUP BY LietotājsID, Niks, LietotājaKategorija.ID, Nosaukums

-- 2.2 Plašāks bibliotēkas apraksts.
-- Lietotājs, spēļu skaits, mīļāko spēļu skaits, vispār kopējās stundas pa
visām spēlēm, spēle ar lielāko stundu skaitu, pēdējā iegūtā spēle, lietotāja
populārākais žanrs un cik spēles ar to žanru
SELECT top(10)
main.*,
stund.SpēlētāsStundas LielākaisStunduSkaitsSpēlei,
stund.Nosaukums VissspēlētākāSpēle,
sub.IegūšanasDatums PēdējāSpēleIegūta,
sub.SpēleID PēdējāsIegūtasSpēlesID,
```

```

sub.Nosaukums PēdējāsIegūtasSpēlesNosaukums,
žanrs.ŽanrsNosaukums PopulārākaisŽanrs,
žanrs.SpēlesArŽanru SpēlesŽanrā,
izstr.Izstrādātājs PopulārākaisIzstrādātājs,
izstr.SpēļuSkaitis SpēļuSkaitisNoIzstrādātāja
FROM
    (SELECT
        Lietotājs.ID,
        Lietotājs.Niks,
        count(SpēleID) as SpēļuSkaitis,
        sum(cast(irIecienīta as int)) IecienītuSpēļuSkaitis,
        sum(SpēlētāsStundas) as KopāNospēlētāsStundas
    FROM LietotājsSpēle
    RIGHT JOIN Lietotājs on LietotājsSpēle.LietotājsID = Lietotājs.ID
    GROUP BY Lietotājs.ID, Lietotājs.Niks) main
LEFT JOIN
    (SELECT
        LietotājsID,
        SpēleID,
        Nosaukums,
        IegūšanasDatums
    FROM
        (
            SELECT *, ROW_NUMBER() over (partition by LietotājsID order by
IegūšanasDatums desc) vieta
            FROM LietotājsSpēle) s
        JOIN Spēle on Spēle.ID = SpēleID
        WHERE vieta = 1) sub on sub.LietotājsID = main.ID
LEFT JOIN
    (SELECT *
    FROM
        (SELECT
            LietotājsID,
            ŽanrsNosaukums,
            count(*) SpēlesArŽanru,
            row_number() over (partition by LietotājsID order by
count(*) desc) VietaPēcSpēļuSkaita
            FROM LietotājsSpēle
            JOIN Spēle on Spēle.ID = LietotājsSpēle.SpēleID
            JOIN SpēleŽanrs on SpēleŽanrs.SpēleID = LietotājsSpēle.SpēleID
            GROUP BY LietotājsID,ŽanrsNosaukums) žr
        WHERE žr.VietaPēcSpēļuSkaita = 1) žanrs on žanrs.LietotājsID = main.ID
LEFT JOIN
    (SELECT
        LietotājsID,
        SpēleID,
        Spēle.Nosaukums,
        SpēlētāsStundas
    FROM
        (
            SELECT *, row_number() over (partition by LietotājsID order by
SpēlētāsStundas desc) vieta
            FROM LietotājsSpēle
            ) as StunduVietas
        JOIN Lietotājs on Lietotājs.ID = LietotājsID
        JOIN Spēle on SpēleID = Spēle.ID
        WHERE vieta = 1) stund on stund.LietotājsID = main.ID
LEFT JOIN
    (SELECT *
    FROM
        (SELECT LietotājsID, Izstrādātājs, count(*) SpēļuSkaitis,
DENSE_RANK() over (partition by LietotājsID order by count(*) desc) vieta

```

```

        FROM LietotājsSpēle
        JOIN Spēle on Spēle.ID = LietotājsSpēle.SpēleID
        GROUP BY LietotājsID, Izstrādātājs) apakšpieprasījums
WHERE apakšpieprasījums.vieta = 1) izstr on izstr.LietotājsID = main.ID

```

-- 3. Draugi

-- Kopīgo draugu skaits, ja lietotājs un tā draugs neparādas tad tiem nav draugu

```

SELECT
    dr1.Lietotājs1 as ID,
    l.Niks as Lietotājs,
    dr1.Lietotājs2 as DraugsID,
    d.Niks as Draugs,
    count(*) KopīgoDrauguSkaitis
FROM Draudzējas dr1
join Draudzējas dr2 on dr2.Lietotājs1 = dr1.Lietotājs2
join Lietotājs l on l.ID = dr1.Lietotājs1
join Lietotājs d on d.ID = dr1.Lietotājs2
join Lietotājs dd on dd.ID = dr2.Lietotājs2
where dr2.Lietotājs2 in (select ws.Lietotājs2 from Draudzējas ws where
ws.Lietotājs1 = dr1.Lietotājs1)
and dr1.Lietotājs1 != dr2.Lietotājs2
group by
    dr1.Lietotājs1,
    l.Niks,
    dr1.Lietotājs2,
    d.Niks
order by dr1.Lietotājs1

```

-- 3.3 Draugu kopīgās spēles

```

SELECT Lietotājs.ID,
    Lietotājs.Niks,
    Draugs.ID,
    Draugs.Niks,
    Spēle.ID,
    Spēle.Nosaukums
FROM Draudzējas
JOIN LietotājsSpēle as ls1 on ls1.LietotājsID = Lietotājs1
JOIN LietotājsSpēle as ls2 on ls2.LietotājsID = Lietotājs2
JOIN Lietotājs on Lietotājs.ID = Lietotājs1
JOIN Lietotājs Draugs on Draugs.ID = Lietotājs2
JOIN Spēle on Spēle.ID = ls1.SpēleID
WHERE ls1.SpēleID = ls2.SpēleID
ORDER BY Lietotājs.ID, Spēle.Nosaukums

```

-- 4. Izstrādātāji

-- Izstrādātāja žanrs ar visvairāk spēlēm

```

select *
from (
select Izstrādātājs.Nosaukums, SpēleŽanrs.ŽanrsNosaukums, count(Spēle.ID)
SpēlesArŽanru, dense_rank() over (partition by Izstrādātājs.Nosaukums order by
count(Spēle.ID) desc) vieta from Izstrādātājs
left join Spēle on Spēle.Izstrādātājs = Izstrādātājs.Nosaukums
left join SpēleŽanrs on SpēleŽanrs.SpēleID = Spēle.ID
group by Izstrādātājs.Nosaukums, ŽanrsNosaukums) ss
where ss.vieta = 1

```

```

-- 5. Spēles
-- 5.1 Spēles spēlētāju informācija t.i. cik lietotājiem spēle pieder, visu
spēlētāju kopā nospēlētās stundas spēlē un cik šobrīd spēli spēlē
CREATE VIEW SpēlesLietotājuInfo as
SELECT main.*, Spēlētāji.Spēlē FROM
    (
        SELECT
            Spēle.ID,
            Spēle.Nosaukums,
            count(ls.LietotājsID) Lietotāji,
            sum(ls.SpēlētāsStundas) KopāNospēlētāsStundas
        FROM Spēle
        LEFT JOIN LietotājsSpēle ls on ls.SpēleID = Spēle.ID
        GROUP BY Spēle.ID, Spēle.Nosaukums) main
LEFT JOIN
    (SELECT
        SpēleID,
        count(LietotājsID) Spēlē
    FROM LietotājsSpēlē
    GROUP BY SpēleID) Spēlētāji on Spēlētāji.SpēleID = main.ID

-- 5.2 Žanra vispopulārākā spēle pēc lietotāju skaita
SELECT * FROM
    (
        SELECT
            ŽanrsNosaukums,
            sli.*,
            DENSE_RANK() over (partition by ŽanrsNosaukums order by Lietotāji
desc) vieta
        FROM SpēlesLietotājuInfo sli
        LEFT JOIN SpēleŽanrs sz on sz.SpēleID = sli.ID) ss
WHERE ss.vieta = 1
ORDER BY ŽanrsNosaukums

-- 6. Mantas
-- Lietotāja inventārs pēc spēlēm
SELECT
    Lietotājs.ID,
    Lietotājs.Niks,
    Spēle.ID SpēleID,
    Spēle.Nosaukums SpēleNosaukums,
    MantasKlase.ID MantasKlaseID,
    MantasKlase.Nosaukums MantasKlaseNosaukums,
    Manta.ID MantaID,
    Manta.DotaisVārds MantasĪpašniekaDotaisVārds,
    Manta.IzveidesDatums,
    MantasKlase.Apraksts
FROM Lietotājs
LEFT JOIN Manta on Manta.ĪpašnieksID = Lietotājs.ID
LEFT JOIN MantasKlase on MantasKlase.ID = Manta.MantasKlaseID
LEFT JOIN Spēle on Spēle.ID = MantasKlase.SpēleID
ORDER BY Lietotājs.ID, Spēle.ID
-- cik klases instances eksistē
select MantasKlase.ID, MantasKlase.Nosaukums, count(Manta.ID) Instances from
MantasKlase
left join Manta on Manta.MantasKlaseID = MantasKlase.ID
group by MantasKlase.ID, MantasKlase.Nosaukums
order by Instances desc

-- 7. Serveri
-- Servera informācija
-- Servera ip adrese, spēles nosaukums un spēlētāju skaits

```

```

SELECT
    Serveris.IPAddress,
    Serveris.Port,
    Serveris.Nosaukums,
    Spēle.ID,
    Spēle.Nosaukums,
    count(LietotājsID) Spēlētāji
FROM Serveris
LEFT JOIN Spēle on Spēle.ID = Serveris.SpēleID
LEFT JOIN LietotājsSpēlē on LietotājsSpēlē.SpēleID = Spēle.ID
GROUP BY
    Serveris.IPAddress,
    Serveris.Port,
    Serveris.Nosaukums,
    Spēle.ID,
    Spēle.Nosaukums
ORDER BY Spēlētāji desc

-- Konkrēta servera spēlētāji
SELECT
    Lietotājs.ID,
    Lietotājs.Niks
FROM Serveris
LEFT JOIN LietotājsSpēlē on LietotājsSpēlē.ServerisID = Serveris.ID
LEFT JOIN Lietotājs on Lietotājs.ID = LietotājsSpēlē.LietotājsID
where IPAddress like '152.137.14.103' and port like '9638'

-- 8. Pirkumi
-- Lietotāja pirkumu informācija, pirkumuskaitis, nopirkto spēļu skaits,
-- iztērētā summa un lielākais pirkums
SELECT
    LietotājsID,
    count(*) PirkumuSkaits,
    sum(SpēļuSkaits) SpēļuSkaits,
    sum(Cena) IztērētāSumma,
    max(Cena) LielākaisPirkums
FROM
    (
        SELECT
            Pirkums.ID,
            Pirkums.LietotājsID,
            Pirkums.Datums,
            PirkumaSpēle.Cena,
            count(*) SpēļuSkaits
        FROM Pirkums
        LEFT JOIN PirkumaSpēle on PirkumaSpēle.PirkumsID = Pirkums.ID
        GROUP BY Pirkums.ID, Pirkums.LietotājsID, Pirkums.Datums,
        PirkumaSpēle.Cena) sub
GROUP BY LietotājsID
-- Spēles ienākumi
SELECT
    Spēle.ID,
    Spēle.Nosaukums,
    sum(PirkumaSpēle.Cena) Ienākumi
FROM Spēle
LEFT JOIN PirkumaSpēle on PirkumaSpēle.SpēleID = Spēle.ID
GROUP BY Spēle.ID, Spēle.Nosaukums
ORDER BY Ienākumi desc

-- 10.1 žanra vislabāk novērtētā spēle
SELECT * FROM
    (

```

```

SELECT
    ŽanrsNosaukums,
    svv.*,
    DENSE_RANK() over (partition by ŽanrsNosaukums ORDER BY
VidējaisVērtējums desc) vieta -- rank pēc vidējā vērtējuma žanrā
FROM
    (
        SELECT Spēle.ID, Spēle.Nosaukums, avg(Recenzija.Vērtējums)
VidējaisVērtējums
        FROM Spēle
        LEFT JOIN Recenzija on Recenzija.SpēleID = Spēle.ID
        GROUP BY Spēle.ID, Spēle.Nosaukums) svv -- spēļu vidējais
vērtējums
    LEFT JOIN SpēleŽanrs sz on sz.SpēleID = svv.ID) ss -- spēļu žanri
WHERE ss.vieta = 1 -- žanra vislabāk novērtētā spēle

```

Citi selecti pieejami failā select.sql kopā ar šiem.