

Jegyzőkönyv
Adatbázis rendszerek I.
Féléves feladat

Készítette: **Kovács László**
Neptunkód: **CIJ404**
Gyak: **Kedd 8-10**
Vezér: *Dr. Bednarik László*

A feladat leírása:

A feladatban egy GTA5 játékszervert kezdetleges adatbázisa van létrehozva, amely az alábbi dolgokat tartalmazza.

A legfontosabb az Account egyed, amely nyilvántartja minden játékos felhasználóját, ezen tulajdonságokkal rendelkezik:

- id: INT tulajdonság, amely a primary és egyedi kulcs is
- userName: egy UTF8-as varchar, amely a felhasználónevet tartalmazza
- adminLevel: a felhasználó adminszintjét tárolja el
- adminNick: a felhasználó adminnevét tárolja el
- passwordHashed: egy 64 karakter hosszú varchar, amelybe a Whirlpoolsal hashelt + saltolt jelszó mentődik
- email tulajdonság: magától értetődő, max hossza 64 karakter
- rememberToken: a megjegyzés rendszer által használt, 128 karakter hosszúságú primary tokeneket tartalmaz
- hwidHash: tulajdonság egy max 32 karakteres string, amely egyedileg azonosítja a felhasználó számítógépét
- hwidExHash: hasonló az előző hwidHash-hoz, csak mást néz a számítógép hardvere közül
- socialId: megadja, hogy a R* Social Clubban mi az ID-je a felhasználónak.
- characterPatternsSaved: lehetősége van a felhasználónak elmenteni a karakterének kinézetét, és később felhasználhatja azt, ha vissza akar váltani egy régebbi karakterére

A második legfontosabb egyed a Characters egyed, amely nyilvántartja egy felhasználónak a karaktereit. Egy felhasználónak több karaktere is lehet, így az Account és a Characters között 1:N kapcsolat van. A Characters egyed ezen tulajdonságokkal rendelkezik:

- id: INT tulajdonság, amely a primary és egyedi kulcs is
- ownerAccountId: INT tulajdonság, foreign key az Account-ból, megadja, hogy hova tartozik a karakter
- characterName: egyedi neve a karakternek, egy UTF8-as varchar
- posData: összetett tulajdonság, amely az x, y, z float pozíciókat tartalmazza a 3D-s térben
- rotData: összetett tulajdonság, amely az rx, ry, rz float rotációt tartalmazza a 3D-s térben
- dimension: megadja, hogy a karakter melyik dimenzióban lett elmentve (többvilágos rendszer)
- health: megadja, hogy a karakternek mennyi élete van
- armor: megadja, hogy a karakternek mennyi páncélja van
- equippedClothes: megadja, hogy mely ruhadarabok vannak rajta jelenleg a karakteren (itemId-k)
- moneyInCash: megadja, hogy a karakternél hány \$ van készpénzben
- bankMoney: származtatott tulajdonság, megadja, hogy hány \$-ja van a karakternek összesen a bankszámláin, amelyek az ő tulajdonában vannak.
- characterVisibleDescription: mindenki által látható karakterleírás
- gta5CharacterAttributes: max 4096 karakter hosszúságú JSON string, amely összefoglalva megadja a karakter kinézetét

- birthData: összetett tulajdonság, rendelkezik dateOfBirth string és placeOfBirth string tulajdonsággal

A Vehicles egyed a járműveket tartalmazza, tulajdonságai:

- dbID: INT, egyedi és primary kulcs
- ownerType: megadja, hogy mi a tulajdonos típusa (0 = játékos, 1 = frakció)
- ownerId: a tulajdonos ID-je, INT (karakterid vagy frakcióid)
- gta5model: megadja, hogy melyik modelt használja a jármű a játékban
- installedExtras: megadja, hogy milyen extra ID-ket kell rárakni lespawoláskor
- posData: x, y, z float a pozíció megadására
- rotData: rx, ry, rz float a rotáció megadására
- healthAndDamageData: GTA5-beli állapot lementése egy járműnek (törésadat, satöbbi)

A CurrentModsVehicles egyed megadja, hogy egyes járműveken milyen típusú és beállítottságú tuningok találhatóak:

- vehicledbId: foreign key, a vehicles dbID-je
- modType: szám azonosító (pld. 11 => engine)
- modValue: a tuning mértéke (pld. 3 a legmagasabb a 11-nél)

Egy karakternek több járműve is lehet, viszont egy járműnek csak egy tulajdonosa lehet, így a Characters és a Vehicles egyed között 1:N kapcsolat van.

Egy autón több tuning is lehet, egyfajta tuningkombináció (pld. 11-es modType és 3-as modValue) több autohoz is tartozhat, így N:M kapcsolat van a Vehicles és a CurrentModsVehicles között.

A BankAccounts egyed rögzíti a játékszerveren található bankszámlákat:

- numericId: INT egyedi azonosító, amely egyben primary key is
- stringId: származtatott tulajdonság: 'JÁTÉKSZERVERNEVE-' + 'numericId-' + ownerId
- currentMoney: a jelenleg bankszámlán található pénzmennyiség \$-ban
- charactersHavingPermissionToUse: megadja, hogy ki férhet hozzá a bankszámlához
- bankAccountName: a szöveges elnevezése a bankszámlának (pld. cégnév)

A BankAccounts és a Characters egyed között N:M kapcsolat van, mivel egy bankszámlához több karakter is hozzáférhet, továbbá egy karakternek több bankszámlája is lehet. A kapcsolathoz tartozik egy „permissionType” tulajdonság ('o', mint tulajdonos, etc..) amely megadja, hogy a karakter a bankszámlához mennyire fér hozzá.

Az Items egyed rögzíti a játékszerveren található tárgyakat:

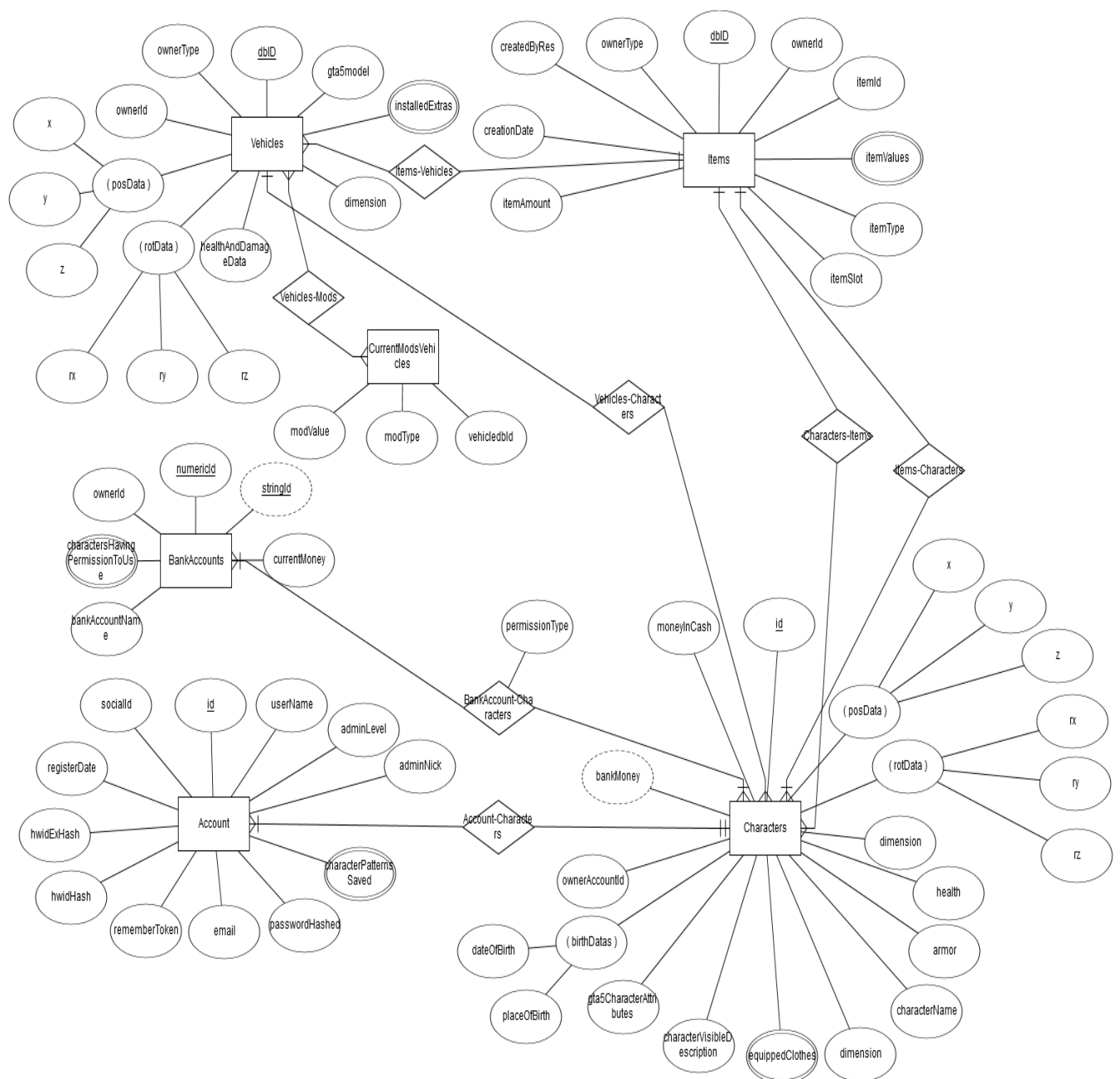
- dbID: INT egyedi azonosító, amely egyben primary key is
- ownerType: megadja, hogy milyen típusú tulajdonosnál van a tárgy (0 = player, 1 = autó, 2 = széf, stb.)
- ownerId: a tulajdonos egyedi azonosítója
- itemId: a tárgy (nem egyedi) azonosítója, pld. 1 = kocsi kulcs
- itemValues: a tárgyhoz tartozó értékek (pld. 5645, ekkor az 5645-ös kocsit nyitja)
- itemType: a tárgy típusa (0 = sima, 1 = szolgálati)

- itemSlot: hanyas számú slotra van rakva a tárgy az inventoryban
- itemAmount: hány darab van belőle
- creationDate: unix timestamp arról, hogy mikor lett létrehozva és odaadva a tárgy
- createdByRes: megmondja, hogy melyik script hozta létre a tárgyat

A Characters és Items egyed között 1:N kapcsolat van, mivel 1 karakternek lehet több tárgya, viszont 1 tárgynak csak és kizárólag egy tulajdonosa lehet.

A Vehicles és Items egyed között 1:N kapcsolat van, mivel 1 járműben lehet több tárgy, viszont 1 tárgynak csak és kizárólag egy tulajdonosa lehet.

Az adatbázis ER-modellje:



Az adatbázis konvertálása relációs modellre:

Az Account egyedből tábla lesz a characterPatternsSaved mező kihagyásával, a characterPatternsSaved tulajdonságából külön tábla lesz gta5CharacterAttributes és savedByAccountId mezőkkel (a savedByAccountId egy foreign key, amely az Account tábla id-jére mutat).

A Characters egyedből tábla lesz az equippedClothes tulajdonság kihagyásával, továbbá a posData és a rotData tulajdonságok szétbontásával, amelyekből x, y, z, rx, ry, rz mező adódik.

Az Account-Characters 1:N kapcsolata miatt az ownerAccountId a Characters táblában foreign key lesz, és az Account tábla id-jére hivatkozik.

A Characters equippedClothes tulajdonságából egy külön tábla lesz, amelyben benne van a Characters tábla id-jére mutató foreign key, továbbá az Items tábla dbID-jére mutató foreign key.

A BankAccounts egyedből tábla lesz, kihagyva a stringId és a charactersHavingPermissionToUse tulajdonságot.

A BankAccounts egyed és a Characters egyed közötti N:M kapcsolat eredményeként létrejön az charactersHavingPermissionToUseBA tábla, amely tartalmaz kettő foreign keyt: numericId-t, amely a BankAccounts id numericId-jára mutat, továbbá a characterId-t, amely a Characters tábla id-jére mutat; és még egy permissionType mezőt.

A Vehicles egyedből tábla lesz, az installedExtras tulajdonság kihagyásával, továbbá a posData és rotData tulajdonságok szétbontásával, amelyekből x, y, z, rx, ry, rz mező adódik.

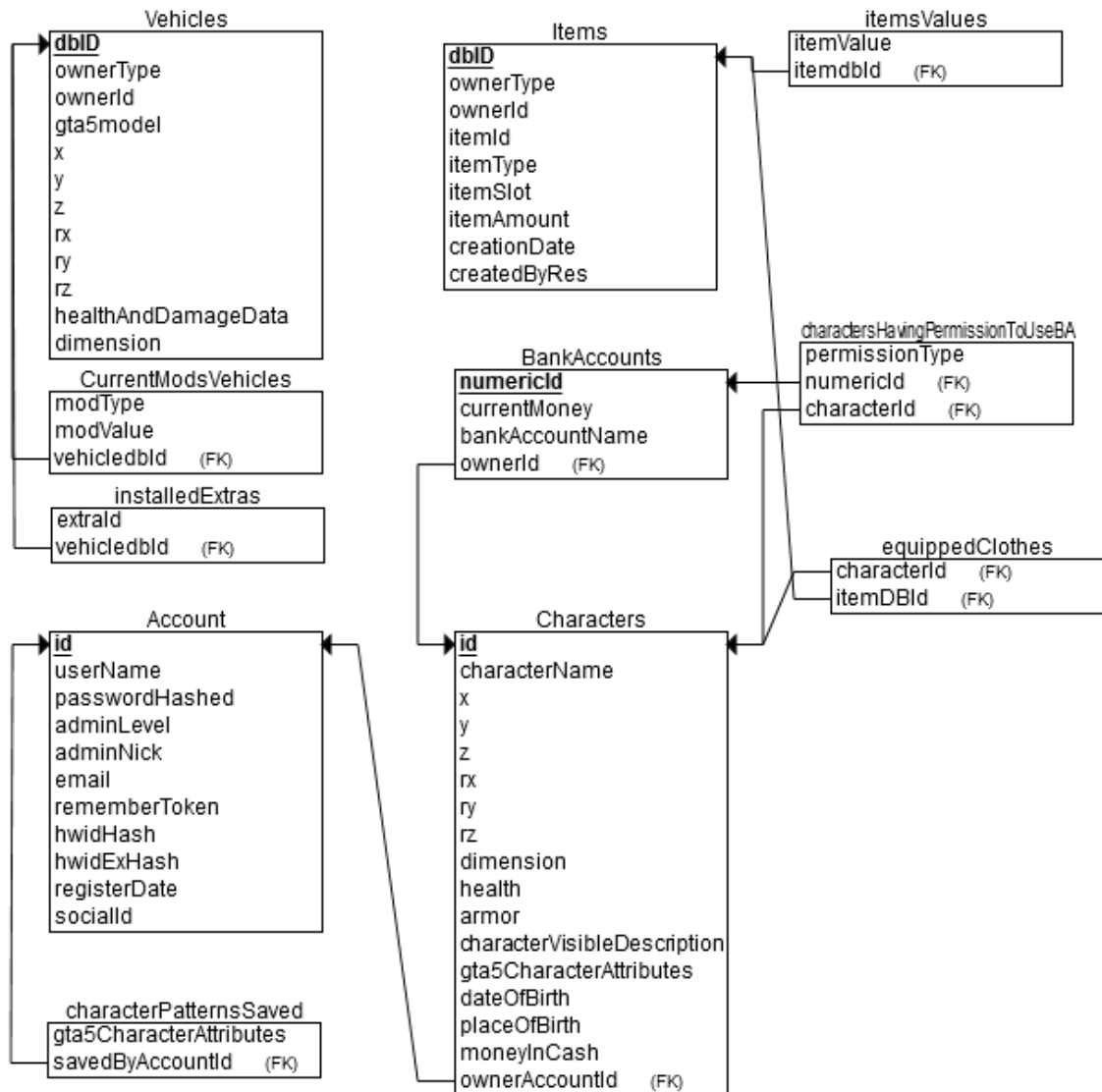
A Vehicles egyed installedExtras tulajdonságából külön tábla lesz, vehicledbId foreign keyvel, amely a Vehicles mező dbID-jára mutat, továbbá extraId mezővel.

A CurrentModsVehicles egyedből tábla lesz, a vehicledbId foreign key lesz, amely rámutat a Vehicles mező dbID-jére.

A Characters és a Vehicles egyed között 1:N kapcsolat van, ami azt jelenti, hogy egy karakternek lehet több autója, azonban egy autónak nem lehet több tulajdonosa.

Az Items egyedből az itemsValue tulajdonság kivételével tábla lesz. Az itemsValue többértékű tulajdonság miatt létrejön egy itemsValue tábla, amely tartalmazza az itemdbId foreign keyt, amely az Items tábla dbID-jére mutat, továbbá az itemValue mezőt. Az itemAmount mezőre integrity check kerül, amely kimondja, hogy az itemAmountnak kötelezően nagyobbnak kell lennie, mint 0.

Az adatbázis relációs modellje:



A táblák létrehozása:

```
CREATE TABLE `Account` (`id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY, `userName`  
varchar(128) CHARACTER SET utf8 COLLATE utf8_hungarian_ci NOT NULL, `passwordHashed`  
VARCHAR(64) NOT NULL, `adminLevel` INT DEFAULT 0, `adminNick` VARCHAR(64) CHARACTER SET  
utf8 COLLATE utf8_hungarian_ci DEFAULT 0, `email` VARCHAR(64) CHARACTER SET utf8 COLLATE  
utf8_hungarian_ci NOT NULL, `rememberToken` VARCHAR(128), `hwidHash` VARCHAR(32) NOT  
NULL, `hwidExHash` VARCHAR(32) NOT NULL, `registerDate` TIMESTAMP DEFAULT  
CURRENT_TIMESTAMP, `socialId` INT NOT NULL);
```

```
CREATE TABLE `characterPatternsSaved` (`gta5CharacterAttributes` VARCHAR(4096),  
`savedByAccountId` INT NOT NULL, FOREIGN KEY (savedByAccountId) REFERENCES Account(id));
```

```
CREATE TABLE `Characters` (`id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY, `characterName`  
varchar(128) CHARACTER SET utf8 COLLATE utf8_hungarian_ci NOT NULL, `x` FLOAT DEFAULT 0, `y`  
FLOAT DEFAULT 0, `z` FLOAT DEFAULT 0, `dimension` INT DEFAULT 0, `health` FLOAT DEFAULT 200,  
`armor` FLOAT DEFAULT 0, `characterVisibleDescription` VARCHAR(256) CHARACTER SET utf8  
COLLATE utf8_hungarian_ci, `gta5CharacterAttributes` VARCHAR(4096), `dateOfBirth` VARCHAR(32)  
NOT NULL, `placeOfBirth` VARCHAR(32) CHARACTER SET utf8 COLLATE utf8_hungarian_ci NOT NULL,  
`moneyInCash` INT NOT NULL, `ownerAccountId` INT NOT NULL, FOREIGN KEY (ownerAccountId)  
REFERENCES Account(id));
```

```
CREATE TABLE `Vehicles` (`dbID` INT NOT NULL AUTO_INCREMENT PRIMARY KEY, `ownerType` INT  
NOT NULL, `ownerId` INT NOT NULL, `gta5model` VARCHAR(16) NOT NULL, `x` FLOAT DEFAULT 0, `y`  
FLOAT DEFAULT 0, `z` FLOAT DEFAULT 0, `rx` FLOAT DEFAULT 0, `ry` FLOAT DEFAULT 0, `rz` FLOAT  
DEFAULT 0, `healthAndDamageData` VARCHAR(512), `dimension` INT DEFAULT 0);
```

```
CREATE TABLE `BankAccounts` (`numericId` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
`currentMoney` INT DEFAULT 0, `bankAccountName` VARCHAR(32) CHARACTER SET utf8 COLLATE  
utf8_hungarian_ci NOT NULL, `ownerId` INT NOT NULL, FOREIGN KEY (ownerId) REFERENCES  
Characters(id));
```

```
CREATE TABLE `Items` (`dbID` INT NOT NULL AUTO_INCREMENT PRIMARY KEY, `ownerType` INT NOT  
NULL, `ownerId` INT NOT NULL, `itemId` INT NOT NULL, `itemType` INT NOT NULL, `itemSlot` INT NOT  
NULL, `itemAmount` INT NOT NULL CHECK (itemAmount > 0), `creationDate` TIMESTAMP DEFAULT  
CURRENT_TIMESTAMP, `createdByRes` VARCHAR(64) NOT NULL);
```

```
CREATE TABLE `itemsValues` (`itemValue` VARCHAR(256) CHARACTER SET utf8 COLLATE  
utf8_hungarian_ci NOT NULL, `itemdbId` INT NOT NULL, FOREIGN KEY (itemdbId) REFERENCES  
Items(dbID));
```

```
CREATE TABLE `CurrentModsVehicles` (`modType` INT NOT NULL, `modValue` INT NOT NULL,  
`vehicledbId` INT NOT NULL, FOREIGN KEY (vehicledbId) REFERENCES Vehicles(dbID));
```

```
CREATE TABLE `installedExtras` (`extraId` INT NOT NULL, `vehicledbId` INT NOT NULL, FOREIGN KEY  
(vehicledbId) REFERENCES Vehicles(dbID));
```

```
CREATE TABLE `charactersHavingPermissionToUseBA` (`permissionType` VARCHAR(8) NOT NULL,
`numericId` INT NOT NULL, `characterId` INT NOT NULL, FOREIGN KEY (numericId) REFERENCES
BankAccounts(numericId), FOREIGN KEY (characterId) REFERENCES Characters(id));
```

```
CREATE TABLE `equippedClothes` (`characterId` INT NOT NULL, `itemDBId` INT NOT NULL, FOREIGN
KEY (characterId) REFERENCES Characters(id), FOREIGN KEY (itemDBId) REFERENCES Items(dbID));
```

A táblák feltöltése:

```
INSERT INTO `account` (`id`, `userName`, `passwordHashed`, `adminLevel`, `adminNick`, `email`,
`rememberToken`, `hwidHash`, `hwidExHash`, `registerDate`, `socialId`) VALUES
```

```
(1, 'teszt1234', 'WhirlpoolHashVagyNem ', 0, '0', 'kovacs128@iit.uni-miskolc.hu ', NULL, '123456',
'43321', '2020-11-30 20:10:13', 1337);
```

```
INSERT INTO `characters` (`id`, `characterName`, `x`, `y`, `z`, `dimension`, `health`, `armor`,
`characterVisibleDescription`, `gta5CharacterAttributes`, `dateOfBirth`, `placeOfBirth`,
`moneyInCash`, `ownerAccountId`) VALUES
```

```
(1, 'Üdv_Özlet', 0, 0, 0, 0, 200, 0, 'Miskolci Egyetem hallgatója', '{"eyebrowType":"test"}',
'1999.12.12', 'Miskolc', 0, 1),
```

```
(2, 'Üdv_Özlet2', 0, 0, 0, 0, 200, 0, 'Miskolci Egyetem hallgatója', '{"eyebrowType":"test"}',
'1999.12.12', 'Miskolc', 0, 1);
```

```
INSERT INTO `vehicles` (`dbID`, `ownerType`, `ownerId`, `gta5model`, `x`, `y`, `z`, `rx`, `ry`, `rz`,
`healthAndDamageData`, `dimension`) VALUES
```

```
(1, 0, 1, 'zentorno', 0, 0, 0, 0, 0, 0, 'A===', 0),
```

```
(2, 0, 2, 'pfister811', 0, 0, 0, 0, 0, 0, 'A===', 0),
```

```
(3, 0, 1, 'schafter2', 0, 0, 0, 0, 0, 0, 'A===', 0);
```

```
INSERT INTO `CurrentModsVehicles` (`modType`, `modValue`, `vehicledbId`) VALUES (11, 3, 1);
```

```
INSERT INTO `installedExtras` (`extraId`, `vehicledbId`) VALUES (1, 1);
```

```
INSERT INTO `items` (`dbID`, `ownerType`, `ownerId`, `itemId`, `itemType`, `itemSlot`, `itemAmount`,
`creationDate`, `createdByRes`) VALUES
```

```
(1, 0, 1, 1, 0, 30, 1, '2020-11-30 20:10:13', 'teszt'),
```

```
(2, 0, 1, 2, 0, 30, 1, '2020-11-30 20:10:13', 'teszt'),
```

```
(3, 0, 1, 2, 0, 30, 1, '2020-11-30 20:27:11', 'teszt');
```

```
INSERT INTO `Items` (`ownerType`, `ownerId`, `itemId`, `itemType`, `itemSlot`, `itemAmount`,
`createdByRes`) VALUES (0, 1, 2, 0, 30, 1, 'teszt');
```

```
INSERT INTO `itemsValues` (`itemValue`, `itemdbId`) VALUES ('1', 1);
```



```
INSERT INTO `equippedClothes` (`characterId`, `itemDBId`) VALUES (1, 2);
```

```
INSERT INTO `bankaccounts` (`numericId`, `currentMoney`, `bankAccountName`, `ownerId`) VALUES  
(1, 123456, 'BBM Protection', 1),  
(3, 43215, 'BBM Protection 2', 2),  
(4, 696969, 'BBM Protection 3', 2),  
(5, 123456, 'BBM Protection 4', 1);
```

```
INSERT INTO `charactershavingpermissiontouseba` (`permissionType`, `numericId`, `characterId`)
VALUES
```

```
('own', 1, 1),
```

```
('fri', 1, 2),
```

```
('own', 3, 2),
```

```
('fri', 3, 1);
```

Lekérdezések:

1. Hány autója van Üdv_Özlet karakternek?

```
SELECT COUNT(`dbID`) FROM `Vehicles` WHERE `ownerType` = 0 AND `ownerId` = (SELECT `id` FROM
`Characters` WHERE `characterName` = 'Üdv_Özlet');
```

2. Hány tárgy található Üdv_Özlet karakternél?

```
SELECT COUNT(`dbID`) FROM `Items` WHERE `ownerType` = 0 AND `ownerId` = (SELECT `id` FROM
`Characters` WHERE `characterName` = 'Üdv_Özlet');
```

3. Hány éves az Üdv_Özlet karakter?

```
SELECT (YEAR(CURRENT_TIMESTAMP)-YEAR(dateOfBirth)) AS `age` FROM `Characters` WHERE
`characterName` = 'Üdv_Özlet';
```

4. Mikor kapta meg a legutolsó tárgyát az Üdv_Özlet karakter?

```
SELECT `creationDate` FROM `Items` WHERE `ownerType` = 0 AND `ownerId` = (SELECT `id` FROM
`Characters` WHERE `characterName` = 'Üdv_Özlet') ORDER BY `creationDate` DESC LIMIT 1;
```

5. Hogy néz ki Üdv_Özlet karaktere?

```
SELECT `gta5CharacterAttributes` FROM `Characters` WHERE `characterName` = 'Üdv_Özlet';
```

6. Mely járműveken nincs egyetlen tuning sem ID szerint?

```
SELECT vehicles.dbID FROM `vehicles` WHERE `dbID` NOT IN (SELECT DISTINCT `vehicledbId` FROM
`currentmodsvehicles`);
```

7. Mennyi pénze van összesen a barátjai bankszámláit is beleszámolva bankban Üdv_Özlet karakternek?

```
SELECT SUM(currentMoney) FROM `bankaccounts` WHERE `numericId` IN (SELECT `numericId` FROM
`charactershavingpermissiontouseba` WHERE `characterId` = (SELECT `id` FROM `Characters` WHERE
`characterName` = 'Üdv_Özlet'));
```

8. Kérdezzük le, hogy egy-egy karakternek mennyi pénze van a bankszámláján!

```
SELECT SUM(currentMoney), CONCAT('teszt-', numericId, '-', ownerId) AS `stringId` ,  
Characters.characterName FROM `bankaccounts` INNER JOIN `Characters` ON characters.id =  
bankaccounts.ownerId GROUP BY `ownerId`;
```

9. Kérdezzük le a járművek tulajdonosának nevét, majd irassuk ki a dbID-ját, továbbá a gta5-ös modelljét, és a tulajdonos nevét!

```
SELECT `dbID`, `gta5model`, Characters.characterName FROM `vehicles` INNER JOIN `Characters` ON  
characters.id = vehicles.ownerId;
```

10. Kérdezzük le, hogy kiknek van hozzáférése név és karakterid szerint a BBM Protection-ös bankszámlákhoz!

```
SELECT `id`, `characterName` FROM `characters` WHERE `id` IN (SELECT DISTINCT `characterId` FROM  
`charactershavingpermissiontouseba` WHERE `numericId` IN (SELECT `numericId` FROM  
`bankaccounts` WHERE `bankAccountName` LIKE '%BBM Protection%'));
```