

ÓBUDAI EGYETEM
Neumann János Informatikai Kar
Nappali Tagozat

ADATBÁZIS-KEZELÉS FÉLÉVES FELADAT

NÉV: **Kovács Patrik Gábor**

NEPTUN KÓD: Y9Z4XX

Beadási határidő:
13. oktatási hét

A DOLGOZAT CÍME:
**Magyar Labdarugó Bajnokság
játékos adatbázisa**

1.Környezet

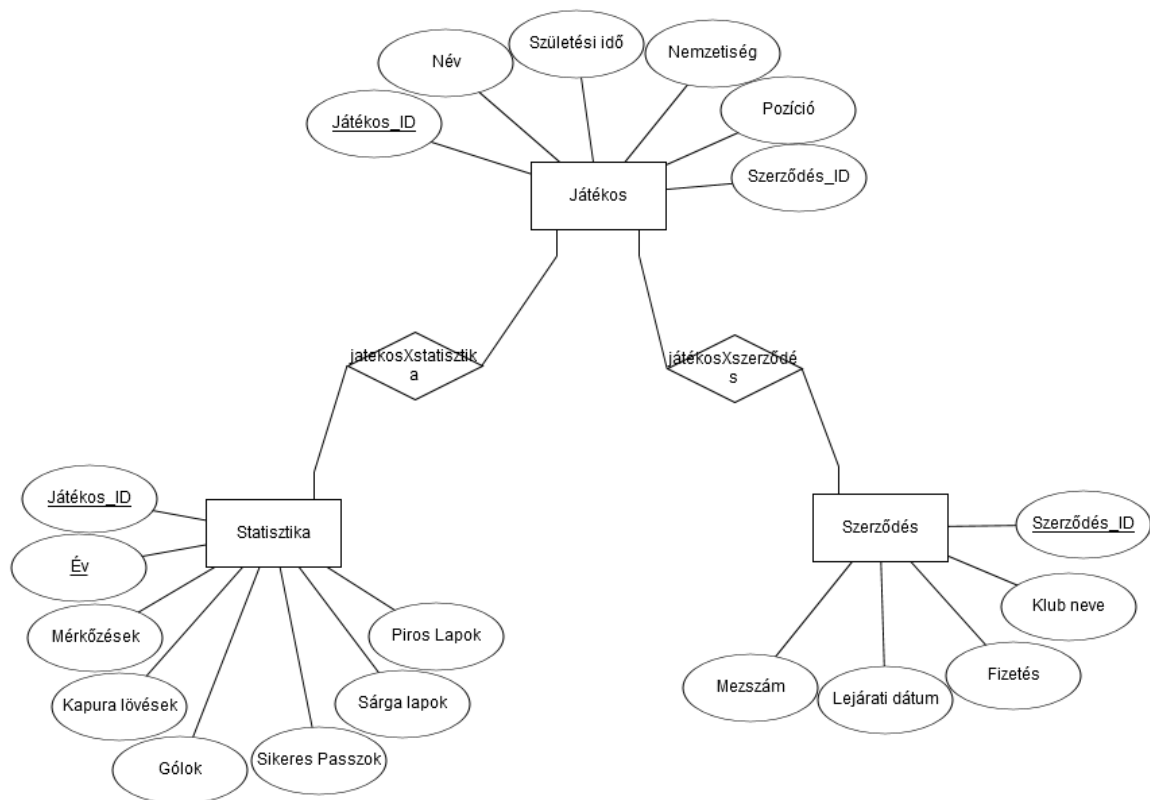
A magyar labdarúgó bajnokság játékosainak adatbázisát fogom megtervezni. A futballisták személyes adatain kívül az egyéni statisztikáikat illetve szerződésük adatait fogja tartalmazni az adatbázis.

Játékos: Minden a játékossal kapcsolatos információt eltárolunk. Ilyen a név, születési dátum, nemzetiség, pozíció és a labdarúgó szerződésének száma.

Szerződés: A játékos aktuális szerződésének adatait tartalmazza. Ilyen például a játékos szerződésének száma, fizetése, klubjának neve, a szerződésnek a lejárat dátuma illetve, hogy a játékos hányas számú mezben játszik.

Statisztika: Ez a tábla a játékosok egyéni teljesítményét mutató statisztikákat tartalmazza idényekre bontva. Többek között, hogy hány mérkőzésen lépett pályára, hány gólt szerzett, sárga illetve piros lapjainak számát és egyéb adatokat.

2.E/K modell:



3.E/K modell átalakítása relációkká:

Játékos(Játékos_ID, Név, Születési idő, Nemzetiség, Pozíció, Szerződés_ID)

Egy játékost egyértelműen a Játékos_ID-je határoz meg, ezért ez az elsődleges kulcs.

JÁTÉKOS_ID: a játékos egyedi id-je az adatbázisban

NÉV: a játékos neve

SZÜLETÉSI IDŐ: a játékos születési ideje

NEMZETISÉG: a játékos nemzetisége

POZÍCIÓ: a játékos posztja(kapus, védő, középpályás, támadó)

SZERZŐDÉS_ID: idegen kulcs, a játékos egyedi számozású nyilvántartott szerződése

Statisztika(Játékos_ID, Év, Mérkőzések, Kapura lövések, Gólok, Sikeres passzok, Sárga lapok, Piros Lap)

Itt egy összetett kulcsot használunk, amelyet az adott év és a játékos ID-je ad meg.

JÁTÉKOS_ID: a játékos egyedi id-je az adatbázisban

ÉV: az adott szezon éve

MÉRKŐZÉSEK: hány mérkőzésen lépett pályára

KAPURA LÖVÉS: hány darab kísérlete volt az adott szezonban

LÖTT GÓL: ebből hány gólt szerzett

PASSZ: sikeres passzok száma

SÁRGA LAP: sárga lapok száma egy idény alatt

PIROS LAP: piros lapok száma egy szezonban

Szerződés(Szerződés_ID, Klub neve, Fizetés, Lejárat dátum, Mezsám)

SZERZŐDÉS_ID: a játékos egyedi számozású nyilvántartott szerződése

KLUB NEVE: melyik klubbal áll szerződésben

FIZETÉS: a játékos heti fizetése euróban

LEJÁRATI DÁTUM: a szerződés végét jelentő dátum

MEZSZÁM: a játékos mezszáma csapaton belül

4.Normalizálás:

Játékos(Játékos_ID, Név, Születési idő, Nemzetiség, Pozíció, Szerződés_ID)

- 1NF, mivel nincs benne összetett attribútum
- 2 NF, a kulcs halmaz egy elemű
- 3 NF, csak kulcsfüggőség van így teljesül
- BCNF, is teljesül

Statisztika(Játékos_ID, Év, Mérkőzések, Kapura lövések, Gólok, Sikeres passzok, Sárga lapok, Piros Lap)

- 1NF, mivel nincs benne összetett attribútum
- 2 NF, a kulcs halmaz egy elemű
- 3 NF, csak kulcsfüggőség van így teljesül
- BCNF, is teljesül

Szerződés(Szerződés_ID, Klub neve, Fizetés, Lejáratí Dátum, Mezsám)

- 1NF, mivel nincs benne összetett attribútum
- 2 NF, a kulcs halmaz egy elemű
- 3 NF, csak kulcsfüggőség van így teljesül
- BCNF, is teljesül

Mindegyik tábla 1NF-ban van, hiszen az attribútumok egyszerű adatok és van kulcs.

Mindegyik tábla 2NF-ben van.

Ezen kívül, a relációkban csak kulcsfüggőség van, ezért az 3NF, sőt BCNF is.

5. DDL

Táblák létrehozása és megszorítások:

```
CREATE TABLE Jatekos(  
jatekos_id NUMBER(5),  
nev VARCHAR2(20),  
szuletesi_ido DATE,  
nemzetiseg VARCHAR2(20),  
pozicio VARCHAR2(20),  
szerzodes_id NUMBER(5),  
CONSTRAINT jatekos_pk PRIMARY KEY(jatekos_id),  
CONSTRAINT jatekos_fk FOREIGN KEY(szerzodes_id)  
REFERENCES Szerzodes(szerzodes_id),  
CONSTRAINT jatekos_jatekosid_uk UNIQUE(jatekos_id)  
);
```

```
CREATE TABLE Szerzodes(  
szerzodes_id NUMBER(5),  
klubnev VARCHAR2(20),  
fizetes NUMBER(10),  
lejarati_datum DATE,  
mezzsam NUMBER(2),  
CONSTRAINT szerzodes_pk PRIMARY KEY(szerzodes_id),  
CONSTRAINT szerzodes_szerzodesid_uk UNIQUE(szerzodes_id)  
);
```

```
CREATE TABLE Statisztika(  
jatekos_id NUMBER(5),  
ev NUMBER(4),  
merkozések NUMBER(2),  
kapuralovesek NUMBER(3),  
golok NUMBER(2),  
sikeress passzok NUMBER(4),  
sargalapok NUMBER(2),  
piroslapok NUMBER(2),  
CONSTRAINT statisztika_pk PRIMARY KEY(jatekos_id,ev)  
);
```

Táblák feltöltése:

```
INSERT INTO Jatekos VALUES(1,'Davide  
Lanzafame',TO_DATE('1987.02.09','YYYY.MM.DD'),'olasz','támadó',100);
```

```
INSERT INTO Jatekos VALUES(2,'Eppel  
Márton',TO_DATE('1991.11.20','YYYY.MM.DD'),'magyar','támadó',101);
```

```
INSERT INTO Jatekos VALUES(3,'Ikenne  
King',TO_DATE('1991.10.29','YYYY.MM.DD'),'nigériai','védő',102);
```

```
INSERT INTO Jatekos VALUES(4,'Böde  
Dániel',TO_DATE('1986.10.24','YYYY.MM.DD'),'magyar','támadó',103);
```

```
INSERT INTO Jatekos VALUES(5,'Gera  
Zoltán',TO_DATE('1979.04.22','YYYY.MM.DD'),'magyar','középpályás',104);
```

```
INSERT INTO Jatekos VALUES(6,'Botka  
Endre',TO_DATE('1994.08.25','YYYY.MM.DD'),'magyar','védő',105);
```

```
INSERT INTO Jatekos VALUES(7,'Kovácsik  
Ádám',TO_DATE('1991.04.04','YYYY.MM.DD'),'magyar','kapus',106);
```

```
INSERT INTO Jatekos  
VALUES(8,'Stopira',TO_DATE('1988.05.20','YYYY.MM.DD'),'zöld-foki-  
szigeteki','védő',107);
```

```
INSERT INTO Jatekos VALUES(9,'Varga  
József',TO_DATE('1988.06.06','YYYY.MM.DD'),'magyar','középpályás',108);
```

```
INSERT INTO Jatekos VALUES(10,'Manjrekar  
James',TO_DATE('1993.08.05','YYYY.MM.DD'),'kanadai','középpályás',109);
```

```

INSERT INTO Szerzodes VALUES(100,'Budapest
Honvéd',600000,TO_DATE('2021.06.31','YYYY.MM.DD'),7);
INSERT INTO Szerzodes VALUES(101,'Budapest
Honvéd',450000,TO_DATE('2020.06.31','YYYY.MM.DD'),9);
INSERT INTO Szerzodes VALUES(102,'Budapest
Honvéd',230000,TO_DATE('2018.06.31','YYYY.MM.DD'),8);
INSERT INTO Szerzodes
VALUES(103,'Ferencváros',280000,TO_DATE('2019.06.30','YYYY.MM.DD'),1
3);
INSERT INTO Szerzodes
VALUES(104,'Ferencváros',150000,TO_DATE('2018.06.30','YYYY.MM.DD'),2
0);
INSERT INTO Szerzodes
VALUES(105,'Ferencváros',300000,TO_DATE('2022.06.30','YYYY.MM.DD'),1
6);
INSERT INTO Szerzodes
VALUES(106,'Videoton',420000,TO_DATE('2020.06.30','YYYY.MM.DD'),1);
INSERT INTO Szerzodes
VALUES(107,'Videoton',220000,TO_DATE('2021.06.30','YYYY.MM.DD'),3);
INSERT INTO Szerzodes
VALUES(108,'Videoton',200000,TO_DATE('2021.06.30','YYYY.MM.DD'),6);
INSERT INTO Szerzodes
VALUES(109,'Vasas',350000,TO_DATE('2019.06.30','YYYY.MM.DD'),10);

```

```

INSERT INTO Statisztika VALUES(1,2016,38,321,23,3463,5,1);
INSERT INTO Statisztika VALUES(1,2015,20,127,10,2248,2,0);
INSERT INTO Statisztika VALUES(1,2014,50,362,36,6341,8,2);
INSERT INTO Statisztika VALUES(2,2016,39,116,23,2175,1,2);
INSERT INTO Statisztika VALUES(3,2016,38,42,3,3675,9,0);
INSERT INTO Statisztika VALUES(4,2014,56,154,3,4002,12,5);
INSERT INTO Statisztika VALUES(5,2016,26,10,1,3953,5,1);
INSERT INTO Statisztika VALUES(6,2015,22,334,26,3412,1,3);
INSERT INTO Statisztika VALUES(7,2013,32,0,0,1264,3,0);
INSERT INTO Statisztika VALUES(8,2016,30,16,2,463,7,1);
INSERT INTO Statisztika VALUES(9,2016,42,42,7,2463,5,0);
INSERT INTO Statisztika VALUES(10,2015,12,71,20,1563,9,0);

```

Megjegyzés: Annyi változást eszközöltem hogy a játékos táblában az összetett kulcs helyett egy egyedi játékos_ID-t használok illetve a statisztika táblában a játékos_ID és az adott év kombinációja lesz az összetett kulcs.

EGYSZERŰ LEKÉRDEZÉSEK:

Listazzuk ki a az 1991-ben született labdarúgókat

```
select nev as "Név",to_char(szuletesi_ido,'YYYY.MM.DD') as "Születési idő"
from jatekos
where szuletesi_ido between to_date('1991.01.01','YYYY.MM.DD') and
to_date('1991.12.31','YYYY.MM.DD');
```

Listázza ki azokat a játékosokat akiknek a pozíciója „védő”

```
select nev as "Név",pozicio as "Pozíció"
from jatekos
where upper(pozicio) like upper('%védő%')
order by nev;
```

Listázza ki az idegenlégiós játékosokat

```
select nev as "Név",nemzetiseg as "Nemzetiség"
from jatekos
where nemzetiseg not like 'magyar'
order by nev;
```

Számítsa ki a játékosok átlagfizetését!

```
select avg(fizetes) as "Átlagfizetés"
from szerzodes;
```

CSOPORTOSÍTÓ LEKÉRDEZÉSEK:

Listázza ki hogy származásuk szerint hány játékos játszik a bajnokságban

```
select nemzetiseg as "Nemzetiségek" , count(*) as db  
from jatekos  
group by nemzetiseg  
order by db desc;
```

Listázza ki hogy az egyes kluboknak hány játékosal van szerződésük

```
select klubnev as "Klubnév" , count(*) as db  
from szerzodes  
group by klubnev  
order by db desc ,klubnev;
```

Listázza ki azokat a klubokat amelyeknek legalább 3 játékosal van szerződésük

```
select klubnev as "Klubnév" , count(*) as db  
from szerzodes  
group by klubnev  
having count(*)>2  
order by db desc ,klubnev;
```

TÖBBTÁBLÁS LEKÉRDEZÉSEK:

Irassa ki a játékosok nevét,klubjukat,fizetésüket illetve mezsámukat

```
select nev as "Név",klubnev as "Klub",fizetes as "Fizetés",mezsam as "Mezsám"  
from jatekos natural join szerzodes  
order by nev;
```

Irassa ki hogy a játékosoknak hány szezonnyi statisztikája van az adatbázisban

```
select *  
from jatekos natural join(  
select jatekos_id,count(*) as "Szezonok száma"  
from statisztika  
group by jatekos_id)  
order by nev;
```

Listázza ki hogy a játékosok a lejátszott szezonjaik során hány gólt szereztek összesen

```
select nev as "Név",sum(golok) as "Gólok száma"  
from jatekos natural join statisztika  
group by nev;
```

ALLEKÉRDEZÉSEK:

Irassuk ki a 2014 es szezon gólkirályát

```
select jatekos.nev
from jatekos,statisztika
where jatekos.jatekos_id=statisztika.jatekos_id and statisztika.ev=2014 and
golok in(select max(golok)
from statisztika
where ev=2014);
```

NÉZETEK:

```
create or replace view JatekosSzerzodes as
select nev as "Név",nemzetiseg as "Nemzetiség",pozicio as "Pozíció",
klubnev as "Klubnév",fizetes as "Fizetés", mezsam as "Mezszám"
from jatekos natural join szerzodes
order by nev;
select * from JatekosSzerzodes;
```

```
create or replace view JatekosStatisztika as
select *
from jatekos natural join statisztika;
select * from JatekosStatisztika;
```

```
create or replace view JatekosStatisztika2016 as
select *
from jatekos natural join statisztika
where ev=2016;
select * from JatekosStatisztika2016;
```

DML:

Töröljük azokat a játékosokat akiknek a pozíciója kapus

```
savepoint kapus_Torles;  
delete from jatekos where upper(pozicio) like upper('kapus');  
rollback;
```

Töröljük a 2015-ös szezon statisztikáit

```
savepoint szezon_Torles;  
delete from statisztika where ev=2015;  
rollback;
```

Töröljük az összes idegenlégiós játékost

```
savepoint legios_Torles;  
delete from jatekos where upper(nemzetiseg) not like upper('magyar');  
rollback;
```

Emeljük meg azon játékosok fizetését akik 300000 forint alatt keresnek

```
savepoint fizetes_UP;  
update szerzodes set fizetes=fizetes+100000 where fizetes<300000;  
rollback;
```

Vonultassuk vissza a 10-es mezsámot

```
savepoint mezsam_UP;  
update szerzodes set mezsam=mezsam+10 where mezsam=10;  
rollback;
```

Jogosultságok kezelése

Hozzunk létre két szerepkört , azokon belül pedig még két-két felhasználót

```
DROP role DBBuilder;
DROP role SimpleUser;
DROP user Emma;
DROP user Etelka;
DROP user Pisti;
DROP user Karcsi;
create role DBBuilder;
grant drop any table, create any table, create any view, backup any table,
alter any table, create session to DBBuilder;
create user Emma identified by "Emma";
grant DBBuilder to Emma;
create user Etelka identified by "Etelka";
grant DBBuilder to Etelka;
create role SimpleUser;
grant select any table,
select any sequence,
create procedure,
create sequence,
insert any table,
update any table,
create session to SimpleUser;
create user Pisti identified by "Pistike";
grant SimpleUser to Pisti;
create user Karcsi identified by "Karcsika";
grant SimpleUser to Karcsi;
```

Alprogramok

```
create or replace procedure SzemelyTorleseJatekos(JATEKOS_ID in
jatekos.jatekos__id%type)
is
  Elofordul integer;
begin
  select count(*) into elofordul
  from jatekos
  where jatekos_id=JATEKOS_ID;
  if Elofordul=0
  then
    dbms_output.put_line('A törlés engedélyezett');
  else
    raise_application_error(-20200,'Sajnos függés van');
  end if;
end;
```

```
create or replace procedure SzemelyTorleseSzerzodes(JATEKOS_ID in
jatekos.jatekos__id%type)
is
  Elofordul integer;
begin
  select count(*) into elofordul
  from szerzodes
  where jatekos_id=JATEKOS_ID;
  if Elofordul=0
  then
    dbms_output.put_line('A törlés engedélyezett');
  else
    raise_application_error(-20200,'Sajnos függés van');
  end if;
end;
```

```
create or replace procedure SzemelyTorleseStatisztika(JATEKOS_ID in
jatekos.jatekos__id%type)
is
  Elofordul integer;
begin
  select count(*) into elofordul
  from statisztika
  where jatekos_id=JATEKOS_ID;
  if Elofordul=0
  then
    dbms_output.put_line('A törlés engedélyezett');
  else
    raise_application_error(-20200,'Sajnos függés van');
  end if;end;
```


Triggerek

Készítsünk egy triggeret amely törlés előtt kiírja a játékos adatait

```
create or replace trigger JatekosTorolTrigger
before delete on jatekos
for each row
begin dbms_output.put_line('A játékos adatai:');
dbms_output.put_line('Neve: ' || old.jatekos.nev || 'Származása:' ||
old.jatekos.nemzetiseg);
exception when NO_DATA_FOUND then dbms_output.put_line('Nincs ilyen játékos');
end;
```

Készítsünk egy triggeret amely szerződés módosítás előtt megkérdezi hogy biztosan módosítani szeretnénk-e.

```
create or replace trigger SzerzodesFrissites
before update on Szerzodes
begin
dbms_output.put_line('Biztosan szeretné frissíteni a szerződést?');
exception when NO_DATA_FOUND then dbms_output.put_line('Nincs ilyen szerződés.');
```

```
end;

create or replace trigger UjSzerzodes
before insert on szerzodes
for each row
begin
dbms_output.put_line('> Az új szerződés lejárat dátuma: ' ||
:new.szerzodes.lejarati_datum')
VoltMarSzerzodes(:new.jatekos_id);
end;
```

PL/SQL

Szerződés törlése

```
declare
cursor kurzor is select * from szerzodes;
rekord kurzor%rowtype;
szerzodesLetezik boolean :=false;
NincsIlyenSzerzodes exception;
begin for rekord in kurzor loop
if rekord.szerzodes_id=2 then
szerzodesLetezik := true;
exit;
end if;
end loop;
if emailLetezik then
dbms_output.put_line('A törlés sikeres!');
delete from szerzodes where szerzodes_id=2;
else
raise NincsIlyenSzerzodes;
end if;
```

Frissítés az szerződésekben

```
DECLARE
CURSOR kurzor IS SELECT * FROM szerzodes;
rekord kurzor%rowtype;
SzerzodesLetezik BOOLEAN := false;
NincsIlyenSzerzodes EXCEPTION;
BEGIN
FOR rekord in kurzor loop
IF rekord.szerzodes_id = 2 THEN
SzerzodesLetezik := true;
EXIT;
END IF;
END LOOP;
IF SzerzodesLetezik THEN
dbms_output.put_line('A megadott szerződés létezik, a módosítás sikeres.');
```

```
UPDATE Szerzodes SET fizetes= fizetes+50000 WHERE szerzodes_id = 2;
ELSE
RAISE NincsIlyenSzerzodes;
END IF;
EXCEPTION
WHEN NincsIlyenSzerzodes THEN
dbms_output.put_line('Nem létezik az adatbázisban ilyen szerződés.');
```

```
END;
```

Lista a 300000 Ft-nál jobban kereső játékosokról

```
DECLARE
CURSOR kurzor IS SELECT * FROM szerzodes;
rekord kurzor%rowtype;
SzerzodesLetezik BOOLEAN := false;
NincsIlyenSzerzodes EXCEPTION;
BEGIN
dbms_output.put_line('300000-nél többet keresők listája');
OPEN kurzor ;
FETCH kurzor INTO rekord;
WHILE kurzor%found LOOP
IF rekord.fizetes > 300000 THEN
SzerzodesLetezik := true;
dbms_output.put_line('Klub: ' || rekord.klubneve || ', Fizetés: ' || rekord.fizetes ||
',Mezszám: ' || rekord.mezszam);
END IF;
FETCH kurzor into rekord;
END LOOP;
IF SzerzodesLetezik = false THEN
RAISE NincsIlyenSzerzodes;
END IF;
EXCEPTION
WHEN NincsIlyenEszkoz THEN
dbms_output.put_line('Nincs 300000-nél jobban kereső játékos');
END;
```