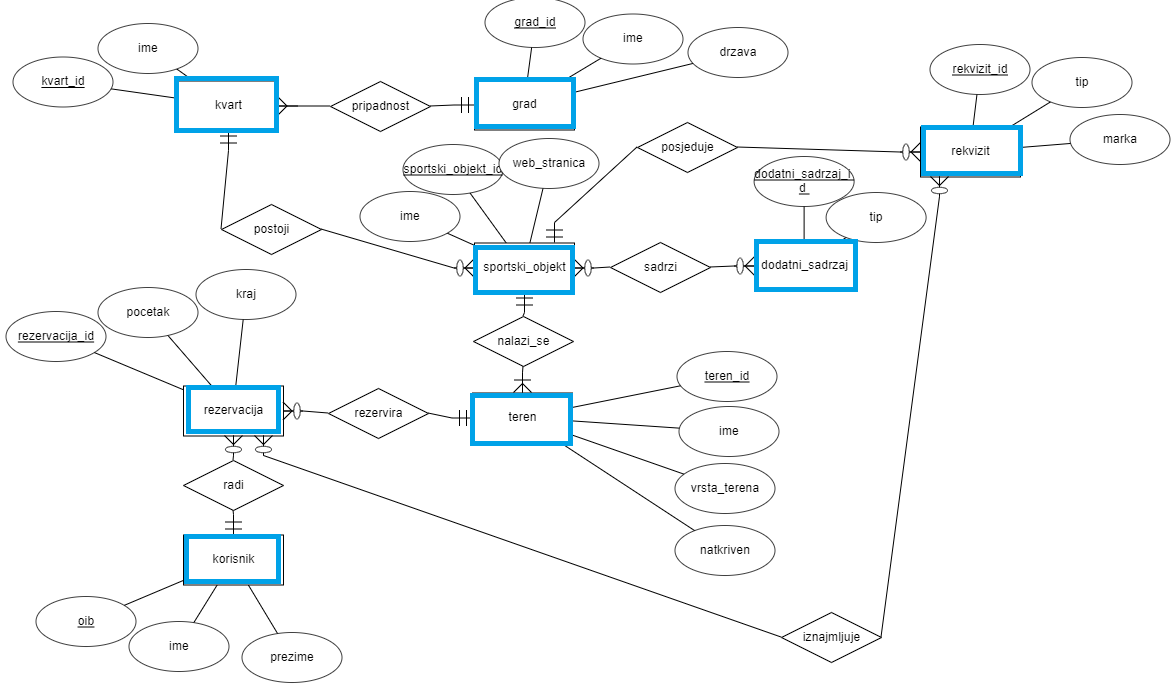
**ER dijagram**



**Entiteti**

korisnik – osoba koja želi rezervirati teren

rezervacija – informacije o rezervaciji

teren – teren koji se želi rezervirati

sportski\_objekt – mjesto u kojem se nalaze različiti sportski tereni

dodatni\_sadrzaj – informacije o dodatnim sadržajima koje sportski objekt može imati (kafić, tuševi, teretana i sl.)

rekvizit – sportski rekviziti (nogometne lopte, košarkaške lopte, reketi, mreže za stolni tenis i sl.) koji se mogu iznajmiti u sportskom objektu

kvart – kvart u kojem se sportski objekt nalazi

grad – grad u kojem se sportski objekt nalazi

**Relacije**

radi – određeni korisnik može napraviti nula ili više rezervacija, određena rezervacija može biti napravljena od jednog i samo jednog korisnika

rezervira – određenom rezervacijom se može rezervirati jedan i samo jedan teren, nad određenim terenom se može napraviti nula ili više rezervacija

iznajmljuje – određenom rezervacijom se iznajmljuje nula ili više rekvizita, određeni rekvizit može biti iznajmljen u nula ili više rezervacija

nalazi\_se – u određenom sportskom objektu se može nalaziti jedan ili više terena, određeni teren se nalazi u jednom i samo jednom sportskom objektu

sadrzi – određeni sportski objekt može sadržavati nula ili više dodatnih sadržaja, dodatni sadržaj može biti sadržan u nula ili više sportskih objekata

posjeduje – određeni sportski objekt posjeduje nula ili više rekvizita, određeni rekvizit može biti posjedovan od jednog i samo jednog sportskog objekta

ima – određeni grad može imati jedan ili više sportskih objekata, određeni sportski objekt može pripadati jednom i samo jednom gradu

pripadnost – određeni kvart pripada jednom i samo jednom gradu, gradu može pripadati više kvartova

postoji – u jednom kvartu postoji nula ili više sportskih objekata, jedan sportski objekt postoji u samo jednom kvartu

**Kod za stvaranje tablica**

CREATE TABLE system.grad

(

grad\_id INT NOT NULL,

ime VARCHAR(60) NOT NULL,

drzava VARCHAR(60) NOT NULL,

PRIMARY KEY (grad\_id)

);

CREATE TABLE system.korisnik

(

oib VARCHAR(60) NOT NULL,

ime VARCHAR(60) NOT NULL,

prezime VARCHAR(60) NOT NULL,

PRIMARY KEY (oib)

);

CREATE TABLE system.dodatni\_sadrzaj

(

dodatni\_sadrzaj\_id INT NOT NULL,

tip VARCHAR(60) NOT NULL,

PRIMARY KEY (dodatni\_sadrzaj\_id)

);

CREATE TABLE system.kvart

(

kvart\_id INT NOT NULL,

ime VARCHAR(60) NOT NULL,

grad\_id INT NOT NULL,

PRIMARY KEY (kvart\_id),

FOREIGN KEY (grad\_id) REFERENCES system.grad(grad\_id)

);

CREATE TABLE system.sportski\_objekt

(

sportski\_objekt\_id INT NOT NULL,

ime VARCHAR(60) NOT NULL,

web\_stranica VARCHAR(60) NOT NULL,

kvart\_id INT NOT NULL,

PRIMARY KEY (sportski\_objekt\_id),

FOREIGN KEY (kvart\_id) REFERENCES system.kvart(kvart\_id)

);

CREATE TABLE system.teren

(

teren\_id INT NOT NULL,

vrsta\_terena VARCHAR(60) NOT NULL,

natkriven INT NOT NULL,

ime VARCHAR(60) NOT NULL,

sportski\_objekt\_id INT NOT NULL,

PRIMARY KEY (teren\_id),

FOREIGN KEY (sportski\_objekt\_id) REFERENCES system.sportski\_objekt(sportski\_objekt\_id)

);

CREATE TABLE system.rezervacija

(

pocetak DATE NOT NULL,

kraj DATE NOT NULL,

rezervacija\_id INT NOT NULL,

teren\_id INT NOT NULL,

oib VARCHAR(11) NOT NULL,

PRIMARY KEY (rezervacija\_id),

FOREIGN KEY (teren\_id) REFERENCES system.teren(teren\_id),

FOREIGN KEY (oib) REFERENCES system.korisnik(oib)

);

CREATE TABLE system.rekvizit

(

rekvizit\_id INT NOT NULL,

tip VARCHAR(60) NOT NULL,

marka VARCHAR(60) NOT NULL,

sportski\_objekt\_id INT NOT NULL,

PRIMARY KEY (rekvizit\_id),

FOREIGN KEY (sportski\_objekt\_id) REFERENCES system.sportski\_objekt(sportski\_objekt\_id)

);

CREATE TABLE system.rezervacija\_rekvizit

(

rezervacija\_id INT NOT NULL,

rekvizit\_id INT NOT NULL,

PRIMARY KEY (rezervacija\_id, rekvizit\_id),

FOREIGN KEY (rezervacija\_id) REFERENCES system.rezervacija(rezervacija\_id),

FOREIGN KEY (rekvizit\_id) REFERENCES system.rekvizit(rekvizit\_id)

);

CREATE TABLE system.objekt\_sadrzaj

(

sportski\_objekt\_id INT NOT NULL,

dodatni\_sadrzaj\_id INT NOT NULL,

PRIMARY KEY (sportski\_objekt\_id, dodatni\_sadrzaj\_id),

FOREIGN KEY (sportski\_objekt\_id) REFERENCES system.sportski\_objekt(sportski\_objekt\_id),

FOREIGN KEY (dodatni\_sadrzaj\_id) REFERENCES system.dodatni\_sadrzaj(dodatni\_sadrzaj\_id)

);

**Funkcije i triggeri**

CREATE OR REPLACE TRIGGER insertTrigger

BEFORE INSERT ON system.rezervacija

FOR EACH ROW

DECLARE

threeDaysException EXCEPTION;

someNum INTEGER;

BEGIN

SELECT COUNT(\*) INTO someNum

FROM KORISNIK

JOIN REZERVACIJA ON KORISNIK.OIB = REZERVACIJA.OIB

WHERE SUBSTR(CURRENT\_DATE,0,9) = SUBSTR(REZERVACIJA.pocetak,0,9)

AND :new.OIB = KORISNIK.OIB;

IF (someNum>2) THEN

RAISE threeDaysException;

END IF;

EXCEPTION

WHEN threeDaysException THEN

RAISE\_APPLICATION\_ERROR(-20005, 'Only 3 reservations per day per user allowed');

END;

/

CREATE OR REPLACE TRIGGER insertTrigger2

BEFORE INSERT ON system.rezervacija

FOR EACH ROW

DECLARE

timeAlreadyReserverd EXCEPTION;

someNum INTEGER;

BEGIN

SELECT COUNT(\*) INTO someNum

FROM REZERVACIJA

WHERE :new.TEREN\_ID = TEREN\_ID AND

:new.POCETAK = rezervacija.pocetak;

IF (someNum>0) THEN

RAISE timeAlreadyReserverd;

END IF;

EXCEPTION

WHEN timeAlreadyReserverd THEN

RAISE\_APPLICATION\_ERROR(-20005, 'Time slot is already reserved');

END;

/

CREATE OR REPLACE TRIGGER insertTrigger3

BEFORE INSERT ON system.rezervacija

FOR EACH ROW

DECLARE

alreadyPlayingElsewhere EXCEPTION;

someNum INTEGER;

BEGIN

SELECT COUNT(\*) INTO someNum

FROM REZERVACIJA

WHERE :new.POCETAK = rezervacija.pocetak

AND :new.OIB = rezervacija.oib;

IF (someNum>0) THEN

RAISE alreadyPlayingElsewhere;

END IF;

EXCEPTION

WHEN alreadyPlayingElsewhere THEN

RAISE\_APPLICATION\_ERROR(-20005, 'You are already plaing at another field');

END;

/

CREATE OR REPLACE FUNCTION get\_rand\_start\_time

RETURN VARCHAR2

IS

rand\_start\_time integer;

rand\_end\_time integer;

date\_only varchar2(30);

full\_start\_date varchar2(30);

full\_end\_date varchar2(30);

BEGIN

rand\_start\_time := dbms\_random.value(10,20);

rand\_end\_time := rand\_start\_time + 1;

date\_only:= SUBSTR(CURRENT\_DATE,0,9);

date\_only:= CONCAT(date\_only,' ');

full\_start\_date:= CONCAT(date\_only,rand\_start\_time);

full\_start\_date:= CONCAT(full\_start\_date,':00');

RETURN full\_start\_date;

END get\_rand\_start\_time;

declare

ukupno\_terena integer;

rand\_teren integer;

res\_id integer;

rand\_start\_time date;

rand\_end\_time date;

begin

SELECT COUNT(\*) INTO ukupno\_terena

FROM teren;

SELECT COUNT(\*) INTO res\_id

FROM rezervacija;

rand\_teren := dbms\_random.value(1,ukupno\_terena);

rand\_start\_time:= TO\_DATE(get\_rand\_start\_time(),'dd-MON-yy hh24:mi');

rand\_end\_time:= TO\_DATE(rand\_start\_time+1/24,'dd-MON-yy hh24:mi');

INSERT INTO system.REZERVACIJA(REZERVACIJA\_ID,POCETAK,KRAJ,TEREN\_ID,OIB) VALUES (

res\_id+1,

rand\_start\_time,

rand\_end\_time,

rand\_teren,

'0311'

);

end;

/

**Postavljanje okruženja**

git clone <https://github.com/defilippomattia/uprpod>

cd uprpod/

Start Docker

Start SQLDeveloper

docker-compose up -d --force-recreate

Paste and Run functions\_and\_triggers.sql in SQLDeveloper

alter session set nls\_date\_format = 'dd-MON-yy hh24:mi'

**Početni zahtjevi**

**„Korisnici moraju vidjeti sve dostupne terene u njihovoj blizini (po kvartovima) te vidjeti postoji li slobodan termin.”**

SELECT TEREN.VRSTA\_TERENA, TEREN.NATKRIVEN, SPORTSKI\_OBJEKT.IME SPORTSKI\_OBJEKT\_IME, KVART.IME KVART\_IME

FROM TEREN

JOIN SPORTSKI\_OBJEKT ON teren.sportski\_objekt\_id = sportski\_objekt.sportski\_objekt\_id

JOIN KVART ON sportski\_objekt.kvart\_id = kvart.kvart\_id

SELECT pocetak, kraj, teren.vrsta\_terena, teren.ime TEREN\_IME, kvart.ime KVART\_IME FROM REZERVACIJA

JOIN TEREN ON teren.teren\_id = rezervacija.teren\_id

JOIN SPORTSKI\_OBJEKT ON sportski\_objekt.sportski\_objekt\_id = teren.sportski\_objekt\_id

JOIN KVART ON kvart.kvart\_id = sportski\_objekt.kvart\_id

WHERE CURRENT\_DATE < pocetak

**“Rezervacija je moguća ako postoji slobodan termin. Jedna osoba može rezervirati jedan termin u određenom trenutku (ne može se dogoditi da osoba rezervira 2 termina u isto vrijeme na različitim terenima).”**

Testiranje dodavanjem istog termina:

INSERT INTO system.REZERVACIJA(REZERVACIJA\_ID,POCETAK,KRAJ,TEREN\_ID,OIB) VALUES (

'9',

TO\_DATE('27-NOV-22 14:00', 'dd-MON-yy hh24:mi'),

TO\_DATE('27-NOV-22 15:00', 'dd-MON-yy hh24:mi'),

'4',

'0515');

Testiranje rezerviranja istog termina u različitim terenima

INSERT INTO system.REZERVACIJA(REZERVACIJA\_ID,POCETAK,KRAJ,TEREN\_ID,OIB) VALUES (

'9',

TO\_DATE('27-NOV-22 14:00', 'dd-MON-yy hh24:mi'),

TO\_DATE('27-NOV-22 15:00', 'dd-MON-yy hh24:mi'),

'1',

'0515');

**“Korisnik može vidjeti i povijest svojih rezervacija.”**

SELECT korisnik.oib,pocetak, kraj, teren.vrsta\_terena, teren.ime TEREN\_IME, kvart.ime KVART\_IME FROM REZERVACIJA

JOIN TEREN ON teren.teren\_id = rezervacija.teren\_id

JOIN SPORTSKI\_OBJEKT ON sportski\_objekt.sportski\_objekt\_id = teren.sportski\_objekt\_id

JOIN KVART ON kvart.kvart\_id = sportski\_objekt.kvart\_id

JOIN KORISNIK ON korisnik.oib = rezervacija.oib

WHERE pocetak < CURRENT\_DATE

**“Treba postojati i mogućnost rezerviranja termina na način “I’m feeling lucky” gdje će se rezervirati nasumični (slobodni) termin u nasumičnom kvartu za nasumični sport.”**

declare

ukupno\_terena integer;

rand\_teren integer;

res\_id integer;

rand\_start\_time date;

rand\_end\_time date;

begin

SELECT COUNT(\*) INTO ukupno\_terena

FROM teren;

SELECT COUNT(\*) INTO res\_id

FROM rezervacija;

rand\_teren := dbms\_random.value(1,ukupno\_terena);

rand\_start\_time:= TO\_DATE(get\_rand\_start\_time(),'dd-MON-yy hh24:mi');

rand\_end\_time:= TO\_DATE(rand\_start\_time+1/24,'dd-MON-yy hh24:mi');

INSERT INTO system.REZERVACIJA(REZERVACIJA\_ID,POCETAK,KRAJ,TEREN\_ID,OIB) VALUES (

res\_id+1,

rand\_start\_time,

rand\_end\_time,

rand\_teren,

'0311'

);

end;

/

**“Za svaki objekt treba postojati i više informacija o dostupnim sadržajima (kafić u sklopu kompleksa, tuševi, je li moguće iznajmljivanje reketa, sportskih lopti i sl.).”**

SELECT SPORTSKI\_OBJEKT.ime, dodatni\_sadrzaj.tip FROM SPORTSKI\_OBJEKT JOIN OBJEKT\_SADRZAJ

ON sportski\_objekt.sportski\_objekt\_id = objekt\_sadrzaj.sportski\_objekt\_id

JOIN DODATNI\_SADRZAJ ON dodatni\_sadrzaj.dodatni\_sadrzaj\_id = objekt\_sadrzaj.dodatni\_sadrzaj\_id

**Zahtjevi programera**

**“Želim dobiti popis svih rezervacija za određenog korisnika u određenom vremenskom razdoblju.”**

SELECT SPORTSKI\_OBJEKT.IME SPORTSKI\_OBJEKT\_IME, TEREN.VRSTA\_TERENA, TEREN.TEREN\_ID, REZERVACIJA.POCETAK,REZERVACIJA.KRAJ FROM REZERVACIJA

JOIN KORISNIK ON REZERVACIJA.OIB = KORISNIK.OIB

JOIN TEREN ON REZERVACIJA.TEREN\_ID = TEREN.TEREN\_ID

JOIN SPORTSKI\_OBJEKT ON teren.sportski\_objekt\_id = sportski\_objekt.sportski\_objekt\_id

WHERE REZERVACIJA.POCETAK BETWEEN

TO\_DATE('09-NOV-22 14:00', 'dd-MON-yy hh24:mi') AND

TO\_DATE('24-NOV-22 14:00', 'dd-MON-yy hh24:mi')

AND KORISNIK.OIB IN ('0541');

**“Moguće je napraviti maksimalno 3 rezervacije u danu po osobi.”**

Testiranje:

INSERT INTO system.REZERVACIJA(REZERVACIJA\_ID,POCETAK,KRAJ,TEREN\_ID,OIB) VALUES (

'9',

TO\_DATE('15-DEC-22 19:00', 'dd-MON-yy hh24:mi'),

TO\_DATE('15-DEC-22 20:00', 'dd-MON-yy hh24:mi'),

'4',

'0515');

**Želim dobiti popis svih rezervacija za određeni kvart.**

SELECT KVART.IME KVART\_IME,SPORTSKI\_OBJEKT.IME SPORTSKI\_OBJEKT\_IME, TEREN.VRSTA\_TERENA, TEREN.TEREN\_ID, REZERVACIJA.POCETAK, REZERVACIJA.KRAJ FROM

REZERVACIJA

INNER JOIN TEREN

ON REZERVACIJA.TEREN\_ID = TEREN.TEREN\_ID

INNER JOIN SPORTSKI\_OBJEKT

ON SPORTSKI\_OBJEKT.SPORTSKI\_OBJEKT\_ID = TEREN.SPORTSKI\_OBJEKT\_ID

INNER JOIN KVART

ON SPORTSKI\_OBJEKT.KVART\_ID = KVART.KVART\_ID

WHERE KVART.IME IN ('Trnje')

**Želim dobiti popis svih objekata u kojima je moguće iznajmiti određeni rekvizit i u kojoj je količini trenutno taj rekvizit raspoloživ.**

SELECT TIP, SPORTSKI\_OBJEKT.IME, COUNT(\*) UKUPNO\_REKVIZITA FROM REKVIZIT

LEFT OUTER JOIN SPORTSKI\_OBJEKT

ON rekvizit.sportski\_objekt\_id = SPORTSKI\_OBJEKT.SPORTSKI\_OBJEKT\_ID

GROUP BY TIP, SPORTSKI\_OBJEKT.IME