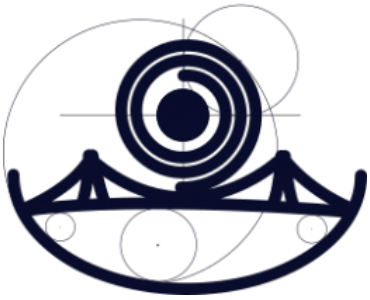
Baza podataka za Hrvatske željeznice

Seminarski rad

Dario Zavišić
predmet: Moderni sustavi baza podataka
profesor: Slobodan Jelić
asistenti: Ena Pribisalić, Mateja Đumić



Odjel za matematiku
Sveučilište Jurja Josipa Strossmayera Osijek
Rujan, 2020.

Sadržaj

1 Uvod

Ova baza podataka je napravljena kao ideja kako bi trebala izgledati baza podataka za Hrvatske željeznice.

Glavni cilj ove baze je da svaki kupac može kupiti sebi kartu. Da tu istu kartu može izdati, a kasnije i pregledati zaposlenik Hrvatskih željeznica.

Dodatno želimo da za svaku kartu imamo vlakove kojima upravljaju također zaposlenici Hrvatskih željenica na određenim relacijama u određeno vrijeme.

Za pravilno funkcioniranje ove baze, najbitniji dio jeste naravno odnos kupca i karte, ali pretežito odnos relacija i karata. Samom kupnjom karte osiguravamo da kupac na svojoj karti sadrži sve potrebne podatke o vremenu i datumu polaska, cijeni karte, rezervaciji sjedala te relaciji kojom se vozi.

Kako bismo bili sigurni da nam svaki putnik u vlaku ima kartu, osigurali smo da svaki vlak ima konduktere. Također kako bi i sami vlak mogao se voziti osigurali smo da ima i strojovođu.

Kroz ovaj seminar ćemo prikazati tablice koje su nam potrebne, njihovo kreiranje te relacije koje ih povezuju.
Također, nakon svih potrebnih unosa ćemo pokazati kako relacije djeluju na unose kada ih pokušamo prikazati.

2 Strukturiranje baze podataka

2.1 Dodavanje tablica

S obzirom da uz seminar je priložena SQL skripta koja sadrži sva stvaranja tablica, ovdje ću za primjer objasniti stvaranje tablica KUPAC i RELACIJA.

Za svakog kupca smo pretpostavili da ima svoju iskaznicu, pa simuliramo potrebne podatke koji su upisani za svakog kupca.

CREATE TABLE *kupac* (
kupac_id INTEGER NOT NULL,
ime VARCHAR2(15) NOT NULL,
prezime VARCHAR2(20) NOT NULL,
adresa VARCHAR2(50) NOT NULL,
tel INTEGER NOT NULL,
dob DATE NOT NULL,
popust_popust_id VARCHAR2(20) NOT NULL
);

Također, u stvaranju tablice za relaciju spremamo podatke poput cijene te relacije i vremenskog trajanja.

CREATE TABLE *relacija* (
rel_id INTEGER NOT NULL,
cijena_relacije NUMBER NOT NULL,
vremensko_trajanje VARCHAR2(9) NOT NULL,
vlak_vlak_id INTEGER NOT NULL
);

U prvoj tablici *kupac* imamo zanimljivi stupac *popust_popust_id* kojim povezujemo taj stupac s drugom tablicom
koja nam sadrži sve popuste, uključujući i popust *nema*, te njihove vrijednosti.

Dok u drugoj nam je zanimljivi stupac *vlak_vlak_id* koji se povezuje s drugom tablicom VLAK te nam govori koji vlak vozi tu relaciju.

ALTER TABLE *kupac*
ADD CONSTRAINT *kupac_popust_fk* FOREIGN KEY (*popust_popust_id*)
REFERENCES *popust* (*popust_id*);

ALTER TABLE *relacija*
ADD CONSTRAINT *relacija_vlak_fk* FOREIGN KEY (*vlak_vlak_id*)
REFERENCES *vlak* (*vlak_id*);

Dobar razlog zašto koristimo takve stupce za relacije je taj što ne moramo unositi svaki put,
npr. naziv popusta i njegovu vrijednost nego se referenciramo na tablicu koja to već sadrži.

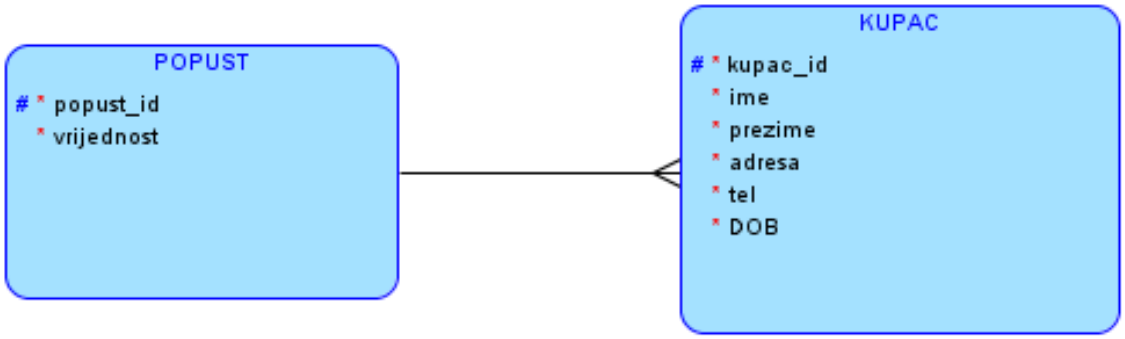
Analogno su stvorene i tablice VLAK, POSTAJA, STAJALIŠTE i KARTA.

2.2 Relacije

Kako je prethodno spomenuto, tablica KUPAC je povezana s tablicom POPUST,
gdje svaki kupac mora imati popust, a gdje popust mora imati barem jednog kupca.

CREATE TABLE *popust* (
popust_id VARCHAR2(20) NOT NULL,
vrijednost INTEGER NOT NULL
);

ALTER TABLE *popust* ADD CONSTRAINT *popust_pk* PRIMARY KEY (*popust_id*);



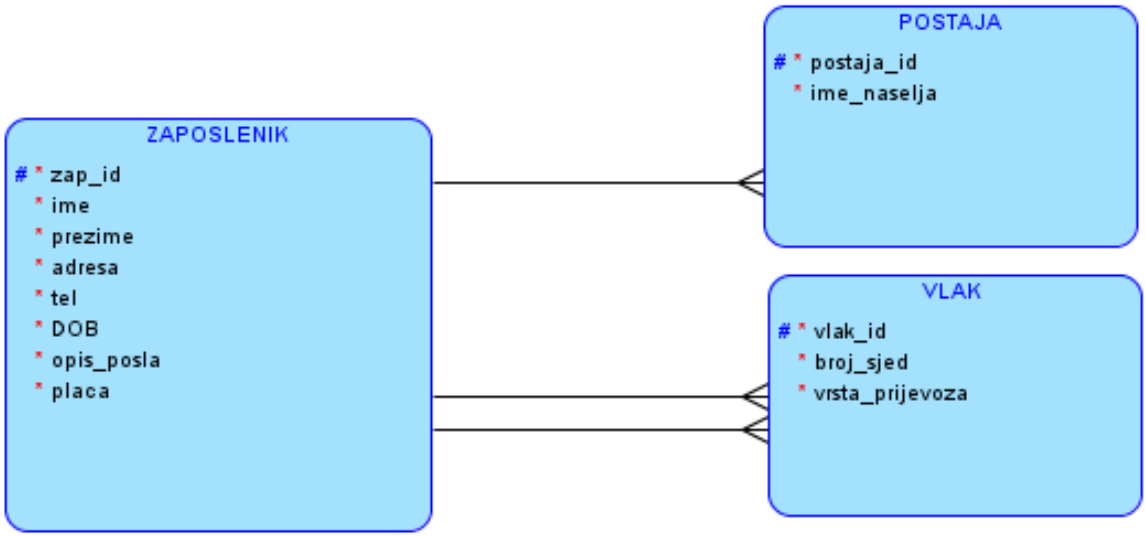
Da bismo došli do najvažnije tablice KARTA, moramo uvesti tablicu ZAPOSLENIK te radna mjesta zaposleniku,
tj. tabice VLAK i STAJALIŠTA.

ALTER TABLE *postaja*
ADD CONSTRAINT *postaja_zaposlenik_fk* FOREIGN KEY (*zaposlenik_zap_id*)
REFERENCES zaposlenik (*zap_id*);

ALTER TABLE *vlak*
ADD CONSTRAINT *vlak_zaposlenik_fk* FOREIGN KEY (*zaposlenik_zap_id*)
REFERENCES zaposlenik (*zap_id*);

ALTER TABLE *vlak*
ADD CONSTRAINT *vlak_zaposlenik_fkv*2 FOREIGN KEY (*zaposlenik_zap_id*1)
REFERENCES zaposlenik (*zap_id*);

Sasvim prirodno je reći da nam zaposlenik kojemu je opis posla kondukter ili strojovođa, da on radi u barem jednom vlaku.
Također nam je prirodno reći da nam i u svakom vlaku radi jedan kondukter ili strojovođa. Isto se odnosi i na zaposlenika koji je blagajnik i radi na postaji.

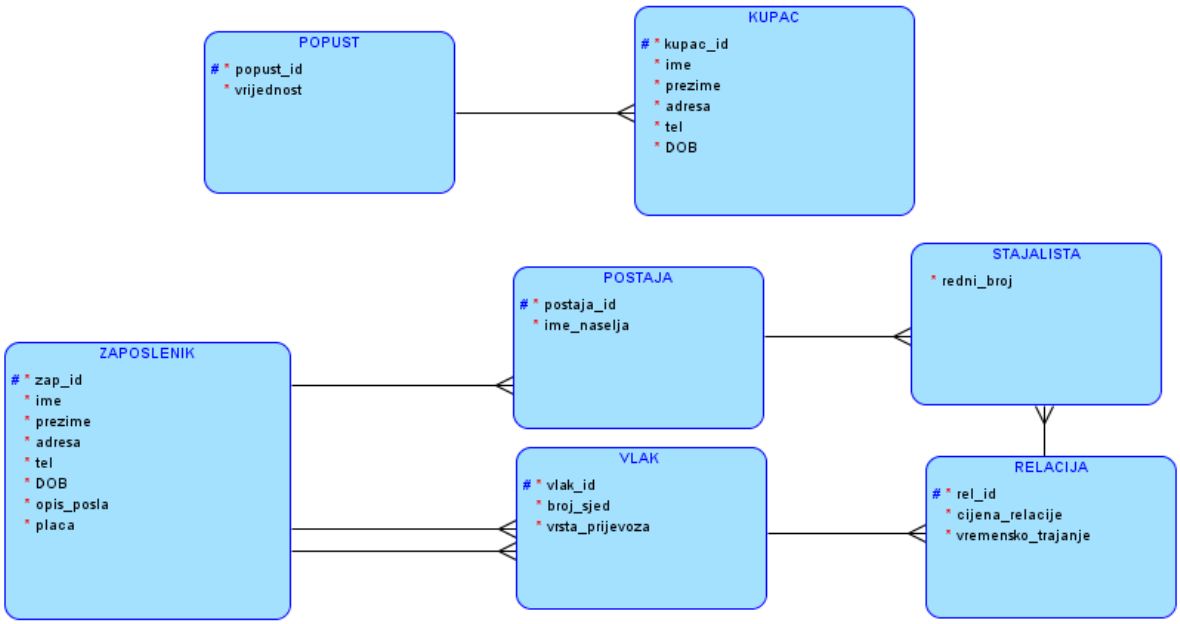


Još nam preostaje pokazati da nam vlak vozi barem jednu relaciju, a tako i da svaka relacija mora imati vlak koji ju vozi.
Pri ćemu nam svaka relacija ima više stajališta, a stajalište svoj redni broj i relaciju kojom se vozi.
Također je svakom stajalištu pridružena postaja, dok je postaji pridruženo barem jedno stajalište.

ALTER TABLE *relacija*
ADD CONSTRAINT *relacija_vlak_fk* FOREIGN KEY (*vlak_vlak_id*)
REFERENCES vlak (*vlak_id*);

ALTER TABLE *stajalista*
ADD CONSTRAINT *stajalista_relacija_fk* FOREIGN KEY (*relacija_rel_id*)
REFERENCES relacija (*rel_id*);

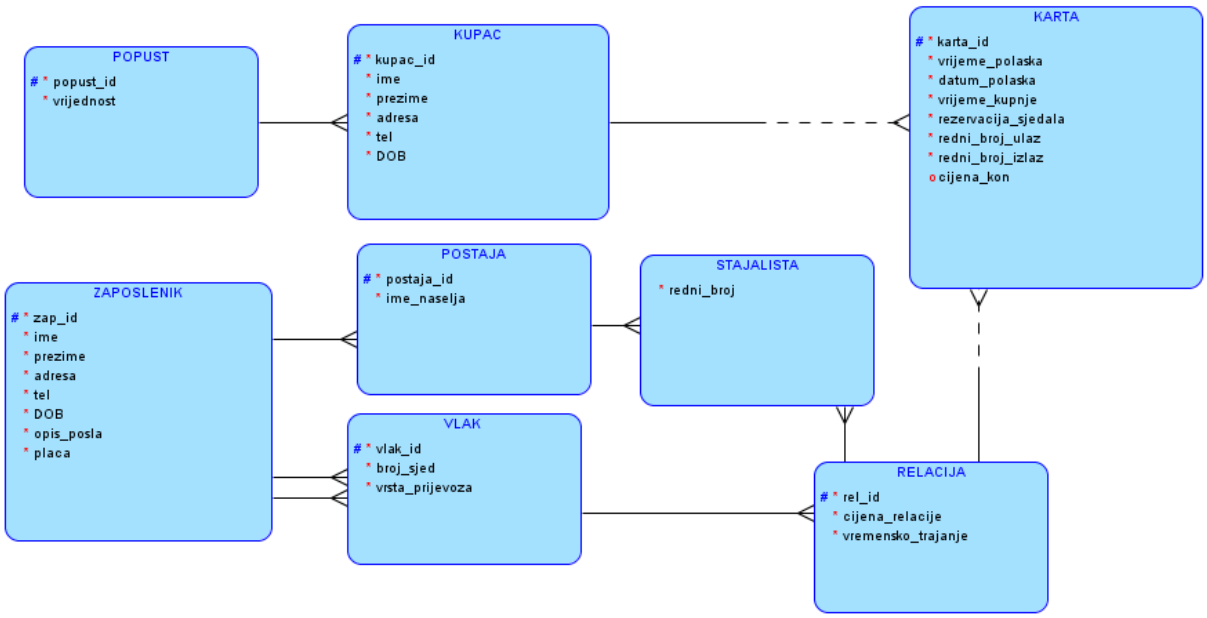
ALTER TABLE *stajalista*
ADD CONSTRAINT *stajalista_postaja_fk* FOREIGN KEY (*postaja_postaja_id*)
REFERENCES postaja (*postaja_id*);

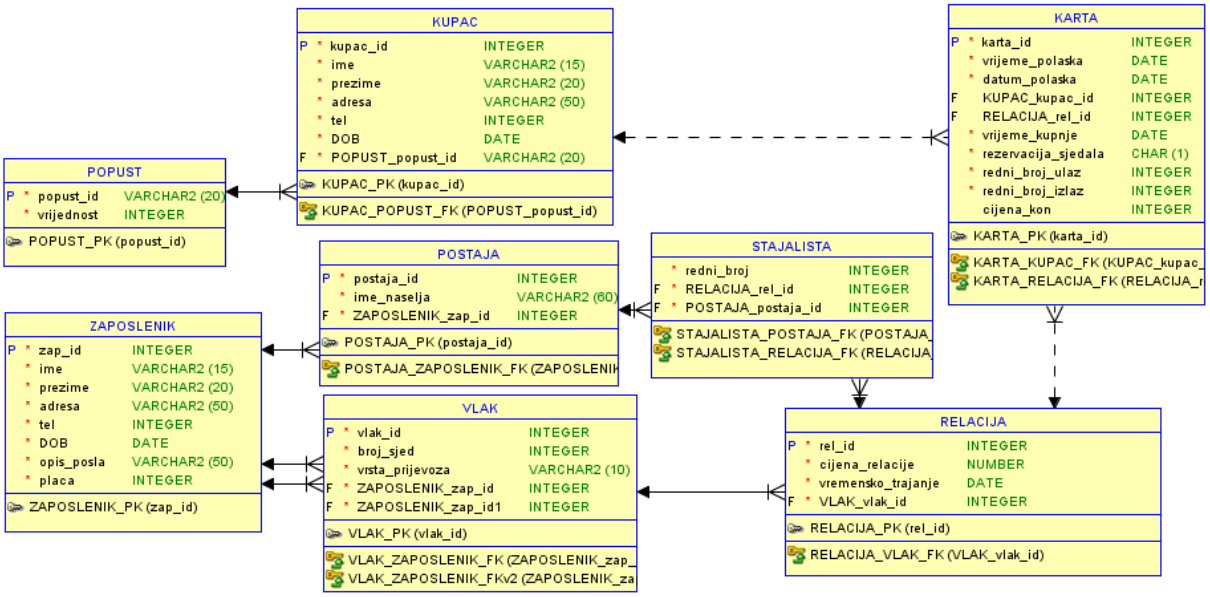


Sada nam konačno preostaje povezati tablicu KARTA. Znamo da relacija se može izdati na jednoj ili više karta, ali karta mora imati relaciju kojom se vozi.
Također tu istu kartu mora kupiti kupac, pri ćemu nam kupac može imati jednu ili više karata.
I time smo uveli sve relacije.

ALTER TABLE *karta*
ADD CONSTRAINT *karta_relacija_fk* FOREIGN KEY (*relacija_rel_id*)
REFERENCES relacija (*rel_id*);

ALTER TABLE *karta*
ADD CONSTRAINT *karta_kupac_fk* FOREIGN KEY (*kupac_kupac_id*)
REFERENCES kupac (*kupac_id*);




KUPAC
P * kupac_id INTEGER
* ime VARCHAR2 (15)
* prezime VARCHAR2 (20)
* adresa VARCHAR2 (50)
* tel INTEGER
* DOB DATE
F * POPUST_popust_id VARCHAR2 (20)
KUPAC_PK (kupac_id)
KUPAC_POPUST_FK (POPUST_popust_id)
KARTA
P * karta_id INTEGER
* vrijeme_polaska DATE
* datum_polaska DATE
F KUPAC_kupac_id INTEGER
F RELACIJA_rel_id INTEGER
* vrijeme_kupnje DATE
* rezervacija_sjedala CHAR (1)
* redni_broj_ulaz INTEGER
* redni_broj_izlaz INTEGER
cijena_kon INTEGER
KARTA_PK (karta_id)
KARTA_KUPAC_FK (KUPAC_kupac_
KARTA_RELACIJA_FK (RELACIJA_r
POPUST
P * popust_id VARCHAR2 (20)
* vrijednost INTEGER
POPUST_PK (popust_id)
POSTAJA
P * postaja_id INTEGER
* ime_naselja VARCHAR2 (60)
F * ZAPOSLENIK_zap_id INTEGER
POSTAJA_PK (postaja_id)
POSTAJA_ZAPOSLENIK_FK (ZAPOSLENIK
STAJALISTA
* redni_broj INTEGER
F * RELACIJA_rel_id INTEGER
F * POSTAJA_postaja_id INTEGER
STAJALISTA_POSTAJA_FK (POSTAJA_
STAJALISTA_RELACIJA_FK (RELACIJA
ZAPOSLENIK
P * zap_id INTEGER
* ime VARCHAR2 (15)
* prezime VARCHAR2 (20)
* adresa VARCHAR2 (50)
* tel INTEGER
* DOB DATE
* opis_posla VARCHAR2 (50)
* placa INTEGER
ZAPOSLENIK_PK (zap_id)
VLAK
P * vlak_id INTEGER
* broj_sjed INTEGER
* vrsta_prijevoza VARCHAR2 (10)
F * ZAPOSLENIK_zap_id INTEGER
F * ZAPOSLENIK_zap_id1 INTEGER
VLAK_PK (vlak_id)
VLAK_ZAPOSLENIK_FK (ZAPOSLENIK_zap_
VLAK_ZAPOSLENIK_FKv2 (ZAPOSLENIK_za
RELACIJA
P * rel_id INTEGER
* cijena_relacije NUMBER
* vremensko_trajanje DATE
F * VLAK_vlak_id INTEGER
RELACIJA_PK (rel_id)
RELACIJA_VLAK_FK (VLAK_vlak_id)

2.3 Procedure

Prva procedura koju ćemo prikazati jeste procedura koja ažurira vrijednosti popusta u tablici POPUST u ovisnosti od proslijeđenog parametra povećanja i same vrste popusta.

CREATE PROCEDURE *azurirajpopust*
(*v_popust_id* IN VARCHAR, *povecanje* IN INTEGER) AS *n_vrijednost* NUMBER;
BEGIN
SELECT COUNT(*)
INTO *n_vrijednost* FROM *popust* WHERE *popust_id* = *v_popust_id*;
IF *n_vrijednost* ¿= 1 THEN
UPDATE *popust*
SET *vrijednost=vrijednost+povecanje*
WHERE *popust_id=v_popust_id*;
COMMIT;
END IF;
EXCEPTION
WHEN OTHERS THEN
ROLLBACK;
END *azurirajpopust*;

CALL *azurirajpopust*('*ucenik*', 0.1);

SELECT *vrijednost* FROM *popust* WHERE *popust_id='ucenik'*;

Zatim imamo proceduru *cijenakarte* koja nam kako ime kaže računa konačnu cijenu karte koju unosimo u atribut *cijena_kon* u tablici KARTA nakon razgledanih ulaznih i izlaznih stanica te nakon primjene samog popusta na cijenu cijele relacije.

CREATE PROCEDURE *cijenakarte*
(*k_karta_id* IN INTEGER) AS
postoji INTEGER;
kup_kupac_id INTEGER;
pop_popust_id VARCHAR(20);
vrijed INTEGER;
cijena_rel INTEGER;
relacij_rel_id INTEGER;
redni_broj_ul INTEGER;
$redni_broj\_iz$ INTEGER;
BEGIN
SELECT COUNT(*) INTO *postoji* FROM *karta* WHERE *karta_id=k_karta_id*;
SELECT *kupac_kupac_id* INTO *kup_kupac_id* FROM *karta* WHERE *karta_id=k_karta_id*;
SELECT *popust_popust_id* INTO *pop_popust_id* FROM *kupac* WHERE *kupac_id=kup_kupac_id*;
SELECT *vrijednost* INTO *vrijed* FROM *popust* WHERE *popust_id=pop_popust_id*;
SELECT *relacija_rel_id* INTO *relacij_rel_id* FROM *karta* WHERE *karta_id=k_karta_id*;
SELECT *cijena_relacije* INTO *cijena_rel* FROM *relacija* WHERE *rel_id=relacij_rel_id*;
SELECT *redni_broj_ulaz* INTO *redni_broj_ul* FROM *karta* WHERE *karta_id* =k_karta_id;
*SELECT*redni_broj_izlaz*INTO*redni_broj_iz*FROM*karta*WHERE*karta_id=*k_karta_id*;
IF *postoji* = 1 THEN
UPDATE *karta*
SET *cijena_kon* = (*cijena_rel* - (*cijena_rel*/((*redni_broj_iz*+1)-*redni_broj_ul*)))*(1-*vrijed*)
WHERE *karta_id=k_karta_id*;
COMMIT;
END IF;
EXCEPTION

WHEN OTHERS THEN
ROLLBACK;
END *cijenakarte*;

CALL *cijenakarte*(10);

SELECT *cijena_kon* FROM *karta* WHERE *karta_id*=10;

Još jedna procedura koja nam je potrebna je u slućaju da imamo osobu koja želi napraviti svoju člansku karticu,
tj. postati kupac pri ćemu moramo unjeti sve potrebne podatke.

CREATE PROCEDURE *novikupac*(*k_kupac_id* IN NUMBER, *k_ime* IN VARCHAR, *k_prezime* IN VARCHAR, *k_adresa* IN
VARCHAR, *k_tel* IN INTEGER, *k_dob* IN DATE, *popust_popust_id* IN VARCHAR) AS
rel_rel_id NUMBER;
BEGIN
INSERT INTO *kupac* VALUES (*k_kupac_id*, *k_ime*, *k_prezime*, *k_adresa*, *k_tel*, *k_dob*, *popust_popust_id*);
END *novikupac*;

CALL *novikupac*(777,'Dario','Zavisic','Absimica11',93283923,'10/10/20','ucenik');

SELECT * FROM *kupac* WHERE *kupac_id*=777;

2.4 Okidači

Prije nego što krenemo unositi podatke u bazu, uvodimo okidače.

Okidač je dodan na tablicu postaja:

CREATE TRIGGER *promjena_posla* AFTER UPDATE OF *zaposlenik_zap_id* ON *postaja*
FOR EACH ROW WHEN
(NEW.*zaposlenik_zap_id* ¿ 556)
BEGIN
DBMS_OUTPUT.PUT_LINE('UPOZORENJE: Moras promjeniti i ulogu!');
END *promjena_posla*;

UPDATE *postaja*
SET *zaposlenik_zap_id* = 600
WHERE *postaja_id*=11;
SELECT * FROM *postaja* WHERE *postaja_id*=11;

Dodavanjem ovog okidača upozorujemo da se mora ažurirati i uloga zaposlenika koji je prešao na posao blagajnika,
i tako smanjujemo mogučnost da nam kondukter ili strojovođa rade na postaji bez da nisu blagajnici.

Imamo i okidač na razini tablice:

DROP TRIGGER *cijenakonacna*; CREATE TRIGGER *cijenakonacna* AFTER INSERT ON *karta* BEGIN UPDATE *karta* SET *cijena_kon* = 0; END *cijenakonacna*;

2.5 Unosi

Za unose u bazu počnemo prvo od tablica POPUST i ZAPOSLENIK.
U ovome seminaru nećemo prikazati velik broj unosa kao što je u SQL skripti, nego dovoljno da se pokaže način unošenja u tablicu.

INSERT INTO popust (popust_id, vrijednost) VALUES ('student',0.5);
INSERT INTO popust (popust_id, vrijednost) VALUES ('ucenik',0.3);
INSERT INTO popust (popust_id, vrijednost) VALUES ('djeca',0.5);
INSERT INTO popust (popust_id, vrijednost) VALUES ('umirovljenik',0.5);
INSERT INTO popust (popust_id, vrijednost) VALUES ('invalid',0.75);
INSERT INTO popust (popust_id, vrijednost) VALUES ('mladi',0.3);
INSERT INTO popust (popust_id, vrijednost) VALUES ('nema',0);

INSERT INTO zaposlenik (zap_id, ime, prezime, adresa, tel, dob, opis_posla, placa) VALUES
(1,'Sarah','Petty','Petra Kruzica 24',6929874171,TO_DATE('08/11/1996', 'DD/MM/YYYY'),'blagajnik',5626);
INSERT INTO zaposlenik (zap_id, ime, prezime, adresa, tel, dob, opis_posla, placa) VALUES
(2,'Aryanna','Vazquez','Bubici 88',3695037471,TO_DATE('02/04/1998', 'DD/MM/YYYY'),'blagajnik',4776);
INSERT INTO zaposlenik (zap_id, ime, prezime, adresa, tel, dob, opis_posla, placa) VALUES
(3,'Layla','Leach','Sesartici 52',1250296593,TO_DATE('04/04/1999', 'DD/MM/YYYY'),'blagajnik',8623);
INSERT INTO zaposlenik (zap_id, ime, prezime, adresa, tel, dob, opis_posla, placa) VALUES
(4,'Nathalie','Gates','Milisici 8',7179143183,TO_DATE('02/11/1988', 'DD/MM/YYYY'),'blagajnik',5104);
INSERT INTO zaposlenik (zap_id, ime, prezime, adresa, tel, dob, opis_posla, placa) VALUES
(5,'Ansley','Lamb','Ulica Blage Zadre 67',4612498723,TO_DATE('15/11/1982', 'DD/MM/YYYY'),'blagajnik',6127);

Sada možemo pokazati kako se popuni tablica KUPAC, VLAK i POSTAJA:

INSERT INTO kupac (kupac_id, ime, prezime, adresa, tel, DOB, popust_popust_id) VALUES
(1,'Ilija','Matic','Marka Marulica 1',033673704,TO_DATE('21/10/1994', 'DD/MM/YYYY'),'nema');
INSERT INTO kupac (kupac_id, ime, prezime, adresa, tel, DOB, popust_popust_id) VALUES
(2,'David','Dragic','M. Vidosevica 2',031554742,TO_DATE('15/04/1983', 'DD/MM/YYYY'),'djeca');
INSERT INTO kupac (kupac_id, ime, prezime, adresa, tel, DOB, popust_popust_id) VALUES
(3,'Dario','Vlasic','Marka Uvodica 3',021343499,TO_DATE('27/06/1990', 'DD/MM/YYYY'),'mladi');
INSERT INTO kupac (kupac_id, ime, prezime, adresa, tel, DOB, popust_popust_id) VALUES
(4,'Stipe','Dragic','Marusinac 4',035410858,TO_DATE('15/12/1989', 'DD/MM/YYYY'),'djeca');
INSERT INTO kupac (kupac_id, ime, prezime, adresa, tel, DOB, popust_popust_id) VALUES
(5,'Ivor','Kasun','Matoseva 5',052210662,TO_DATE('02/07/1995', 'DD/MM/YYYY'),'umirovljenik');

INSERT INTO vlak (vlak_id, broj_sjed, vrsta_prijevoza, zaposlenik_zap_id, zaposlenik_zap_id1) VALUES (1,10,'radnicki',557,558);
INSERT INTO vlak (vlak_id, broj_sjed, vrsta_prijevoza, zaposlenik_zap_id, zaposlenik_zap_id1) VALUES
(2,79,'brzi',559,560);
INSERT INTO vlak (vlak_id, broj_sjed, vrsta_prijevoza, zaposlenik_zap_id, zaposlenik_zap_id1) VALUES
(3,85,'brzi',561,562);
INSERT INTO vlak (vlak_id, broj_sjed, vrsta_prijevoza, zaposlenik_zap_id, zaposlenik_zap_id1) VALUES
(4,45,'brzi',563,564);
INSERT INTO vlak (vlak_id, broj_sjed, vrsta_prijevoza, zaposlenik_zap_id, zaposlenik_zap_id1) VALUES
(5,5,'teretni',565,566);

INSERT INTO postaja (postaja_id, ime_naselja, zaposlenik_zap_id) VALUES (1,'Bedenica',1); INSERT INTO postaja (postaja_id, ime_naselja, zaposlenik_zap_id) VALUES (2,'Bistra',2); INSERT INTO postaja (postaja_id, ime_naselja, zaposlenik_zap_id) VALUES (3,'Brckovljani',3); INSERT INTO postaja (postaja_id, ime_naselja, zaposlenik_zap_id) VALUES (4,'Brdovec',4); INSERT INTO postaja (postaja_id, ime_naselja, zaposlenik_zap_id) VALUES (5,'Dubrava',5);

Kako bismo došli do karte sada moramo popuniti tablicu RELACIJA pa zatim STAJALISTA, čime

smo se približili srži ove baze:

INSERT INTO relacija (rel_id, cijena_relacije, vremensko_trajanje, vlak_vlak_id) VALUES (1,181,'06:00:00',36);
INSERT INTO relacija (rel_id, cijena_relacije, vremensko_trajanje, vlak_vlak_id) VALUES (2,84,'01:30:00',47);
INSERT INTO relacija (rel_id, cijena_relacije, vremensko_trajanje, vlak_vlak_id) VALUES (3,184,'03:30:00',47);
INSERT INTO relacija (rel_id, cijena_relacije, vremensko_trajanje, vlak_vlak_id) VALUES (4,120,'06:30:00',25);
INSERT INTO relacija (rel_id, cijena_relacije, vremensko_trajanje, vlak_vlak_id) VALUES (5,164,'12:00:00',45);

INSERT INTO stajalista (redni_broj, relacija_rel_id, postaja_postaja_id) VALUES (1,1,478);
INSERT INTO stajalista (redni_broj, relacija_rel_id, postaja_postaja_id) VALUES (2,1,511);
INSERT INTO stajalista (redni_broj, relacija_rel_id, postaja_postaja_id) VALUES (3,1,193);
INSERT INTO stajalista (redni_broj, relacija_rel_id, postaja_postaja_id) VALUES (4,1,29);
INSERT INTO stajalista (redni_broj, relacija_rel_id, postaja_postaja_id) VALUES (5,1,142);

I sada konačno nakon što smo ubacili u svaku od tablica potrebnih za kartu, popunjavamo tablicu KARTA na sljedeći način:

INSERT INTO karta (karta_id, vrijeme_polaska, datum_polaska, kupac_kupac_id, relacija_rel_id, vrijeme_kupnje, rezervacija_sjedala, redni_broj_ulaz, redni_broj_izlaz) VALUES
(1,'05:00:00',TO_DATE('7/16/2020', 'MM/DD/YYYY'),1,154,'03:00:00',1,1,5);
INSERT INTO karta (karta_id, vrijeme_polaska, datum_polaska, kupac_kupac_id, relacija_rel_id, vrijeme_kupnje, rezervacija_sjedala, redni_broj_ulaz, redni_broj_izlaz) VALUES
(2,'05:45:00',TO_DATE('7/28/2020', 'MM/DD/YYYY'),2,61,'04:45:00',0,2,9);
INSERT INTO karta (karta_id, vrijeme_polaska, datum_polaska, kupac_kupac_id, relacija_rel_id, vrijeme_kupnje, rezervacija_sjedala, redni_broj_ulaz, redni_broj_izlaz) VALUES
(3,'06:45:00',TO_DATE('8/5/2020', 'MM/DD/YYYY'),3,136,'05:30:00',1,1,6);
INSERT INTO karta (karta_id, vrijeme_polaska, datum_polaska, kupac_kupac_id, relacija_rel_id, vrijeme_kupnje, rezervacija_sjedala, redni_broj_ulaz, redni_broj_izlaz) VALUES
(4,'01:30:00',TO_DATE('8/10/2020', 'MM/DD/YYYY'),4,128,'07:00:00',0,1,7);
INSERT INTO karta (karta_id, vrijeme_polaska, datum_polaska, kupac_kupac_id, relacija_rel_id, vrijeme_kupnje, rezervacija_sjedala, redni_broj_ulaz, redni_broj_izlaz) VALUES
(5,'04:45:00',TO_DATE('8/28/2020', 'MM/DD/YYYY'),5,67,'09:00:00',1,1,2);

Time smo popunili sve tablice koje imamo u ovoj bazi podataka.

2.6 Upiti

Jedan od najjednostavnijih upita možemo prikazati na primjeru, ako nas zanima redosljed stajališta za određenu relaciju.

SELECT * FROM *stajalista* WHERE *relacija_rel_id* = 10;

Izlaz:

	REDNI_BROJ	RELACIJA_REL_ID	POSTAJA_POSTAJA_ID
1	1	10	414
2	2	10	196
3	3	10	238
4	4	10	117
5	5	10	419
6	6	10	214

Zatim, ako bi nas zanimala prosječna plaća zaposlenika, nju možemo dobiti na sljedeći način:

SELECT MAX(AVG(*placa*))
FROM *zaposlenik*
WHERE *zap_id* IN
(
SELECT *zap_id*
FROM *zaposlenik*
WHERE *placa*¿1
)
GROUP BY *zap_id*;

	MAX(AVG(PLACA))
1	8974

Također ako nas zanima na kojem vlaku radi koji strojovodja, možemo pozvati upit:

SELECT *z.ime*, *z.prezime*, *z.opis_posla*, *v.vlak_id*
FROM *zaposlenik z* INNER JOIN *vlak v*
ON *z.zap_id*=*v.zaposlenik_zap_id*
ORDER BY *z.ime*;

	IME	PREZIME	OPIS_POSLA	VLAK_ID
1	Adonis	Beltran	strojovodja	59
2	Alden	Lucero	strojovodja	24
3	Alexia	Sanders	strojovodja	85
4	Alexia	Sanders	strojovodja	100
5	Alonzo	Blevins	strojovodja	9
6	Amirah	Foley	strojovodja	61
7	Antonio	Santiago	strojovodja	75
8	Antonio	Santiago	strojovodja	90
9	Ariel	Booker	strojovodja	93
10	Ariel	Booker	strojovodja	78
11	Ariella	Phelps	strojovodja	47
12	Blaine	Parsons	strojovodja	32
13	Brady	Randall	strojovodja	18
14	Branden	Cortez	strojovodja	70
15	Brandon	Pollard	strojovodja	48
16	Briley	Bell	strojovodja	21
17	Bryanna	Hughes	strojovodja	41
18	Carson	Hurst	strojovodja	38

Ako želimo vidjeti koliko karata je do sada izdano možemo pozvati sljedeći upit:

SELECT *k.ime*, *k.prezime*, *k.popust_popust_id*, *r.kupac_kupac_id*, *r.vrijeme_polaska*, *r.datum_polaska*, *r.relacija_rel_id*, *r.redni_broj_ulaz*, *r.redni_broj_izlaz*
FROM *kupac k* INNER JOIN *karta r*
ON *k.kupac_id=r.kupac_kupac_id*
ORDER BY *r.datum_polaska*;

	IME	PREZIME	POPUST_POPUST_ID	KUPAC_KUPAC_ID	VRIJEME_POLASKA	DATUM_POLASKA	RELACIJA_REL_ID	REDNI_BROJ_ULAZ	REDNI_BROJ_IZLAZ
1	Ilija	Matic	nema	1	05:00:00	16.07.20	154	1	5
2	David	Dragic	djeca	2	05:45:00	28.07.20	61	2	9
3	Dario	Vlasic	mladi	3	06:45:00	05.08.20	136	1	6
4	Stipe	Dragic	djeca	4	01:30:00	10.08.20	128	1	7
5	Ivor	Kasun	umirovljenik	5	04:45:00	28.08.20	67	1	2
6	Danijel	Pavletic	umirovljenik	6	05:45:00	02.11.20	37	1	2
7	Damjan	Dragovic	nema	7	02:30:00	06.11.20	195	2	3
8	Nino	Lovren	nema	8	14:00:00	01.01.21	126	1	3
9	Bruno	Tomic	student	9	03:45:00	05.01.21	185	2	3
10	Leo	Tomcic	nema	10	04:00:00	14.01.21	87	2	3
11	Lukas	Peric	ucenik	11	06:30:00	26.01.21	18	1	2
12	Toni	Jurisa	ucenik	12	14:00:00	10.02.21	1	2	9
13	Danijel	Ivanovic	invalid	13	02:45:00	05.04.21	193	2	6
14	Domagoj	Pavletic	ucenik	14	08:30:00	13.04.21	67	1	8
15	Lukas	Vinkovic	djeca	15	03:45:00	21.04.21	81	2	3
16	Franjo	Markovic	student	16	07:45:00	23.06.21	11	2	6
17	Rafael	Matic	student	17	04:00:00	13.07.21	130	2	3
18	Ivan	Kosar	student	18	10:00:00	03.08.21	91	2	6

Ako želimo pokazati koji sve vlakovi voze koje relacije te također pripadajuće podatke o toj relaciji i tome vlaku:

SELECT *r.rel_id* AS "Redni broj vlaka", *r.cijena_relacije* AS "Cijena putovanja", *r.vremensko_trajanje* AS "Trajanje putovanja", *v.broj_sjed* AS "Broj sjedala u vlaku", *v.vrsta_prijevoza* AS "Vrsta vlaka"
FROM *relacija r* INNER JOIN *vlak v*
ON *v.vlak_id=r.vlak_vlak_id*
ORDER BY *v.vrsta_prijevoza*;

	Redni broj vlaka	Cijena putovanja	Trajanje putovanja	Broj sjedala u vlaku	Vrsta vlaka
1	134	65	04:45:00	79	brzi
2	197	151	11:00:00	55	brzi
3	96	120	08:30:00	45	brzi
4	112	60	03:45:00	45	brzi
5	117	174	08:30:00	45	brzi
6	67	111	08:30:00	119	brzi
7	80	139	02:00:00	83	brzi
8	151	23	10:00:00	83	brzi
9	121	124	04:00:00	20	brzi
10	87	126	06:45:00	52	brzi
11	162	161	05:45:00	52	brzi
12	31	121	03:00:00	34	brzi
13	142	19	02:45:00	34	brzi
14	127	69	12:00:00	25	brzi
15	130	34	02:45:00	25	brzi
16	164	31	14:00:00	25	brzi
17	190	171	03:30:00	101	brzi

2.7 Indeksi

U ovoj bazi bi smo mogli iskoristiti indekse da poboljšamo brzinu upita nad zaposlenicima, ako bismo imali veliki broj zaposlenika s velikim brojem različitih prezimena.

SELECT *zap_id*, *ime*, *prezime*
FROM *zaposlenik*
WHERE *prezime* = 'Dennis';

CREATE INDEX *index_zaposlenik_prezime* ON *zaposlenik*(*prezime*);

Također isto nam može trebati kod kupaca, pa imamo:

SELECT *kupac_id*, *ime*, *prezime*
FROM *kupac*
WHERE *prezime* = 'Vukovic';

CREATE INDEX *index_kupac_prezime* ON *kupac*(*prezime*);