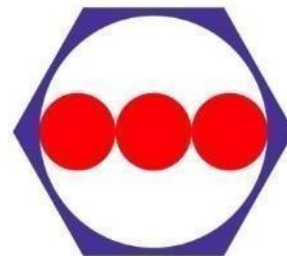




Univerzitet u Novom Sadu
FAKULTET TEHNIČKIH NAUKA

Departman za industrijsko inženjerstvo i inženjerski menadžment
Inženjerstvo informacionih sistema



INFORMACIONI SISTEM ZA PODRŠKU RADA SLUŽBE
ZA LJUDSKE RESURSE
-SISTEMI BAZA PODATAKA-

Sandra Lazarević IT76/2019

Novi Sad, 2022.

Sadržaj

UVOD	3
1.ANALIZA PROGRAMSKOG DOMENA.....	4
2.ER MODEL	5
3.ER MODEL PODŠEME	6
4.TABELARNI PRIKAZ OBELEŽJA I OGRANIČENJA.....	8
5.RELACIONI MODEL	15
6.DDL	19
7.DML	22
8.SQL	24
9.OBJEKTI.....	25
10.UPITI	30
11.FUNKCIJE	34
12.PROCEDURE	38
13.TRIGERI	43
ZAKLJUČAK	46

Spisak slika

Slika 1 ER šema cele baze podataka (na nivou naziva tipova).....	5
Slika 2 ER podšema Nagrade, sankcije I obuka	6
Slika 3 Kreiranje tebele DDL naredbom CREATE	19
Slika 4 Izmena tabele DDL naredbom ALTER	20
Slika 5 Brisanje tabele DDL naredbom DROP	20
Slika 6 Uklanjanje zapisa iz tabele DDL naredbom TRUNCATE	20
Slika 7 Primer selekcija podataka iz tabele	22
Slika 8 Primer izmene podata u tabeli	22
Slika 9 Primer insertovanja podataka u tabelu	23
Slika 10 Primer brisanja podataka iz tabele	23
Slika 11 Primer iščitavanja sankcija zaposlenih u Trebinju i Beogradu	24
Slika 12 Primer kreiranja i brisanja šeme baze podataka	25
Slika 13 Primer kreiranja i brisanja tabele baze podataka.....	25
Slika 14 Slika 13 Primer kreiranja i brisanja sekevence baze podataka	26
Slika 15 Primer kreiranja i brisanja funkcija baze podataka	26
Slika 16 Primer kreiranja i brisanja procedure baze podataka	27
Slika 17 Primer kreiranja i brisanja trigeru baze podataka.....	28
Slika 18 Primer kreiranja i uklanjanja kursora u okviru baze podataka	28
Slika 19 Rezultat prvog upita	30
Slika 20 Rezultat drugog upita	31
Slika 21 Rezultat trećeg upita	31
Slika 22 Rezultat četvrtog upita.....	32
Slika 23 Rezultat petog upit	33
Slika 24 Pozitivan rezultat prve funkcije	35
Slika 25 Negativan rezultat prve funkcije	35
Slika 26 Negativan rezultat prve funkcije	35
Slika 27 Pozitivan rezultat druge funkcije	37
Slika 28 Pozitivan rezultat druge funkcije	37
Slika 29 Pozitivan rezultat druge funkcije	37
Slika 30 Negativan rezultat druge funkcije	37
Slika 31 Pozitivan rezultat prve procedure.....	39
Slika 32 Pozitivan rezultat prve procedure.....	39
Slika 33 Pozitivan rezultat druge procedure.....	42
Slika 34 Negativan rezultat druge procedure	42
Slika 35 Negativan rezultat druge procedure	42
Slika 36 Rezultat kada su uslovi trigeru uspesno zadovoljeni	44
Slika 37 Rezultat kada su uslovi trigeru nesupesno zadovoljeni	44
Slika 38 Rezultat drugog trigeru	45
Slika 39 Rezultat drugog triger.....	45

UVOD

Projekat koji je predstavljen u ovoj dokumentaciji podrazumeva razvoj baze podataka za potrebe rada jednog dela službe za ljudske resurse, odnosno razvoj baze podataka HRIS (engl. *Human Resource Information System*) u ulozi tehnološke, digitalne informacione podrške menadžmentu ljudskih resursa.

Kako ljudi predstavljaju najznačajniji resurs svake organizacije, i proces upravljanja njima veoma je složen i višedimenzionalan. Ovaj neprekidni i dinamični proces ima zadatak da obezbedi da pravi ljudi dođu na pravo mesto u organizaciji, kao i da poboljša radni učinak zaposlenih na strateški, etički i društveno odgovoran način. Upravljanje zaposlenima, njihovim ponašanjem i potencijalima nije lako, jer je ljudsko ponašanje promenljivog karaktera, a često i nepredvidivo. Savremene organizacije sve jasnije shvataju da im valjano upravljanje ljudskim resursima može značajno doprineti sticanju održivih kompetitivnih prednosti. Svrha takvih sistema je da se pomogne kreiranje, organizovanje i raspoloživost značajnih poslovnih informacija i znanja bilo kada i bilo gde da su u organizaciji potrebne.

Ukupni ljudski resursi u savremenim organizacijama obuhvataju: znanje, stručnost, iskustvo, sposobnosti i veštine, inovativne i kreativne ideje, motivacija i uključenost u ostvarivanje organizacionih ciljeva. Osnovni princip upravljanja ljudskim resursima jeste da pravi čovek, u pravo vreme bude na pravom mestu i da svako dobije ono što mu pripada, odnosno da se svakom pojedincu omogući da napreduje u skladu sa svojim sposobnostima i kvalitetima za čije ostvarivanje upravo u velikoj meri pomažu ovakvi sistemi.

U daljem tekstu, priložen je ER model podataka, čijim prevođenjem je kreiran relacioni model podataka za ovaj sistem. Takođe, priložen je i tabelarni prikaz ograničenja pojave tipa, integriteta tipa i jedinstvenih vrednosti. Na samom kraju, na osnovu relacionog modela podataka, izvršena je implementacija šeme baze podataka, gde su kreirane tabele i ograničenja, kao i funkcije, procedure, kursori i trigeri.

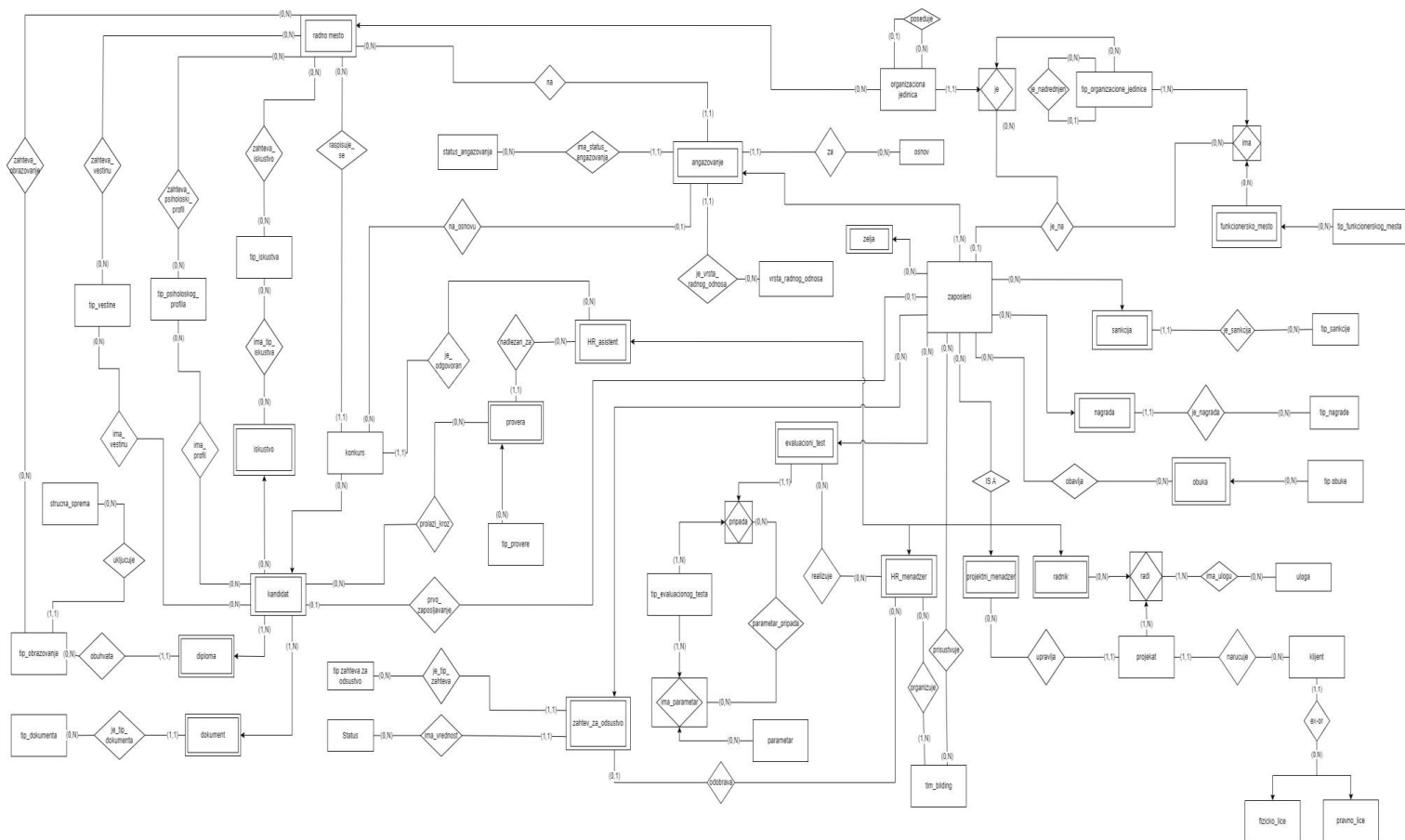
1. ANALIZA PROGRAMSKOG DOMENA

Osnovni cilj upravljanja ljudskim resursima jeste poboljšanje koordinacije i pojačavanje intenziteta snage preduzeća u pravcu ostvarivanja ciljeva organizacije. Ciljevi upravljanja ljudskim resursima moraju biti usklađeni i kompatibilni sa ciljevima celokupnog poslovanja. Neophodno je uskladiti individualne, organizacione i društvene ciljeve, a istovremeno omogućiti ostvarivanje strategije i ublažiti negativne uticaje koji dolaze iz okruženja. Zaposleni i ulaganja u njih predstavljaju investiciju u poslovanje i razvoj, a povrat na nju organizacija ostvaruje najboljom mogućom upotrebom i razvijanjem njihovih potencijala. Iz toga proističe zaključak da je osnovni poslovni cilj osigurati pravi broj zaposlenih, pravi kvalitet, u pravo vreme, na pravom mestu i na pravi način iskoristiti potencijale radi ostvarivanja organizacijskih ciljeva, povećanje organizacijske konkurentske snage i uspešnosti kao i da su krajnji ekonomski ciljevi: porast proizvodnosti i profitabilnosti, snižavanje ukupnih troškova, osiguranje konkurentske sposobnosti, porast ukupne organizacijske uspešnosti.

Nema efikasnog strategijskog upravljanja ljudskim resursima bez podrške informacionih tehnologija i informacionih sistema. Razvoj informacionih tehnologija i informacionih sistema dramatično menja upravljanje ljudskim resursima; transformiše tradicionalne pristupe u digitalizovane onlajn procese u internet okruženju i vrši podršku svim subprocessima i aktivnostima i obezbeđuje informacije HR osoblju, operativnom, taktičkom i strategijskom menadžmentu. Dakle, drugi važan kapital organizacije, posmatran u kontekstu strategijskog upravljanja i upravljanja ljudskim resursima, jeste informacioni kapital.

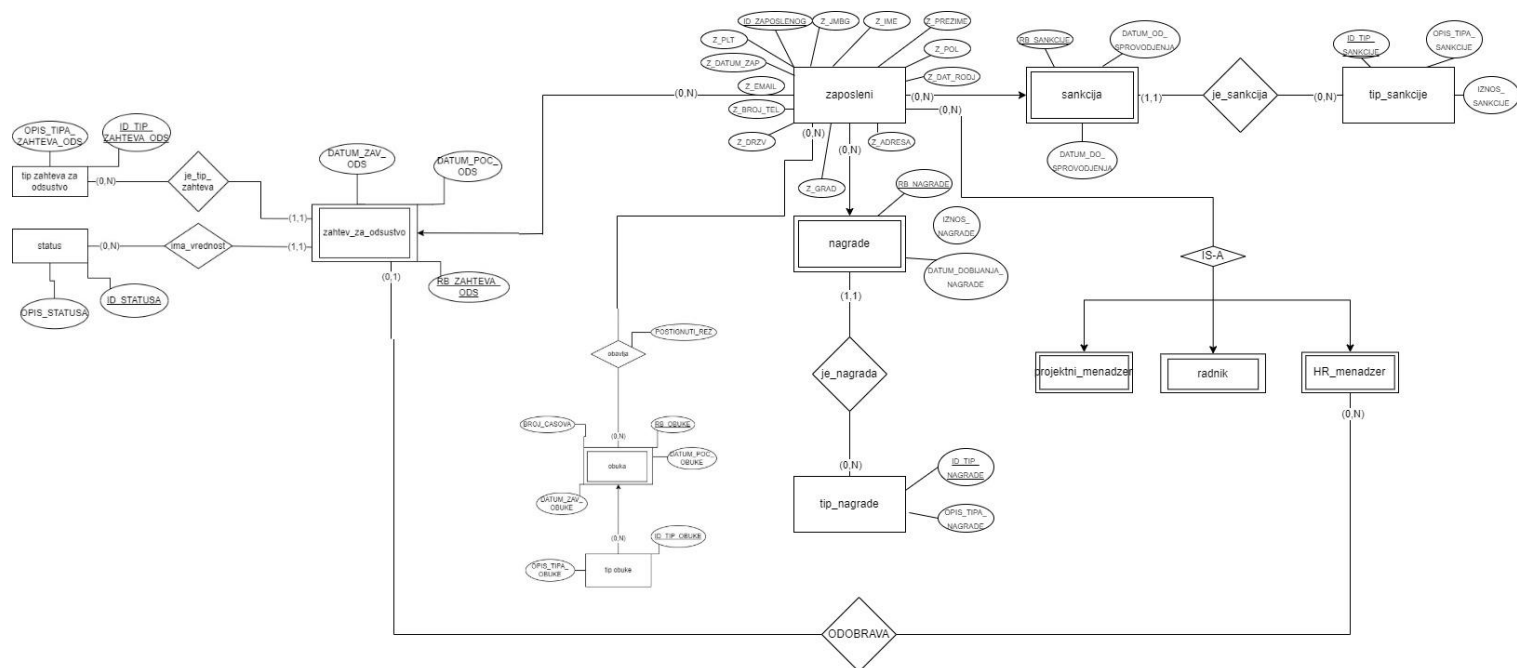
Nema nijedne IT ili vrste informacionog sistema koji ne doprinese upravljanju ljudskim resursima. Neki sistemi to čine u jednom domenu više i bolje, a u drugom manje i slabije, dok drugi sistemi daju sasvim obrnut benefit i uticaj. Najviše se raspoloživost informacionog kapitala u upravljanju ljudskim resursima meri nivom razvijenosti onlajn sistemima transakcione obrade, bilo da je reč o modulu nekog ERP sistema, ili da je reč o HRIS-u (engl. *Human Resources Information Systems*). Druga tehnologija, koja ima još veći strategijski značaj, je BI (*Business Intelligence*) tehnologija i BI sistemi podrške odlučivanja.

2.ER MODEL



Slika 1 ER šema cele baze podataka (na nivou naziva tipova)

3.ER MODEL PODŠEME



Slika 2 ER podšema Nagrade, sankcije I obuka

Sistem vodi evidenciju i o nagradama koje zaposleni dobijaju za svoj trud, rad i posvećenost poslu, kao i sankcijama za njihov nedovoljno dobro odrađen posao. Sistem nagrađivanja u preduzeću čine direktne i indirektne nadoknade. Pod direktnim nadoknadama se podrazumevaju osnovna plata i nagrade (bonusi). Osnovna plata može biti obračunata na osnovu broja radnih sati, a može biti i fiksna bez obzira na broj radnih sati. Nagrade mogu biti individualne (povišice, stimulacije), grupne i organizacione, koje se dobijaju na osnovu seniorata (pretpostavka je da razlike u visini plate zavise od iskustva ili dužine radnog staža, pa se nagrađuju stabilni,iskusni kadrovi), individualnog učinka, ostvarenog profita ili drugih mera uspeha. Pod indirektnim nadoknadama podrazumeva se sledeće:

- zdravstveno osiguranje;
- socijalno osiguranje;
- penziono osiguranje;
- godišnji odmor;

- društveni život, rekreacija i druge beneficije.

Određeni su tipovi nagrada i tipovi sankcija koji postoje u organizaciji i u okviru tih tipova može se nalaziti i više konkretnih nagrada/sankcija koje je zaposleni dobio. U okviru nagrade/sankcije beleži se tip nagrade/sankcije kao i zaposleni koji je dobio tu nagradu/sankciju. Zaposleni može da dobije više nagrada/sankcija, a ne mora nijednu.

Prisustva na radnim mestima i odsustva sa istih moraju biti evidentirana, kako zbog nagrađivanja tako i zbog sankcionisanja nepoštovanja radnog vremena. Pod odsustvovanjem se podrazumevaju i godišnji odmori, bolovanja, slobodni dani i slično. Status zahteva za odsustvo se modeluje tipom entiteta *Status*, koji je putem tipa poveznika *ima_vrednost* povezan sa tipom entiteta *zahtev_za_odsustvo*. Pod statusom se podrazumeva, na primer, informacija da li je zahtev za odsustvo odobren, na čekanju ili odbijen. Jedan zahtev za odsustvo ima tačno jedan status zahteva, dok jedan status može imati više zahteva za odsustvo. Odobravanje zahteva za odsustvo obavlja HR menadžer koji je modelovan putem tipa entiteta *HR_menadzer*. Jedan zahtev za odsustvo može biti odobren od strane najviše jednog HR menadžera, dok HR menadžer može da odobri više zahteva za odsustvo.

Da bi preduzeće bilo konkurentno, neophodno je stalno ulagati u razvoj kadrova. Dakle, neohodno je periodično sprovoditi obuke kadrova sa ciljem zadovoljenja kako potreba preduzeća, tako i samih zaposlenih, jer se na ovaj način zaposleni razvijaju i napreduju na hijerarhijskoj lestvici. Obuku modelujemo pomoću tipa entiteta *obuka*, koja mora biti određenog tipa obuke. U tip entiteta *tip_obuke* beležimo opis obuke za koju se zaposleni opredelio. *Obavlja* je tip poveznika koji sadrži 7atribute kao što su uspešnost zaposlenog u odabranoj obuci, kao i datumi početka i završetka obuke.

4.TABELARNI PRIKAZ OBELEŽJA I OGRANIČENJA

Tabela 1 Tip entiteta zaposleni

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	ID_ZAPOSLENOG	Identifikacioni broj zaposlenog	INTEGER	5	⊥	$d > 0$
2.	Z_JMBG	Jedinstveni matični broj zaposlenog	INTEGER	13	⊥	$d > 0$
3.	Z_IME	Ime zaposlenog	VARCHAR	20	⊥	Δ
4.	Z_PREZIME	Prezime zaposlenog	VARCHAR	20	⊥	Δ
5.	Z_POL	Pol zaposlenog	VARCHAR	1	T	$d = \{ 'M', 'Z' \}$
6.	Z_DAT_RODJ	Datum rođenja zaposlenog	DATE		⊥	Δ
7.	Z_ADRESA	Ulica i broj adrese zaposlenog	VARCHAR	20	⊥	Δ
8.	Z_GRAD	Grad u kojem živi zaposleni	VARCHAR	20	T	Δ
9.	Z_DRZV	Država u kojoj živi zaposleni	VARCHAR	20	T	Δ
10.	Z_BROJ_TEL	Broj telefona zaposlenog	VARCHAR	20	⊥	Δ
11.	Z_EMAIL	Email zaposlenog	VARCHAR	20	⊥	Δ
12.	Z_DATUM_ZAP	Datum zapošljavanja	DATE		⊥	Δ
13.	Z_PLT	Plata zaposlenog	REAL	20	⊥	Δ
Ključ	K={ID_ZAPOSLENOG}					

	Ne postoje dva zaposlena sa istim Z_JMBG
--	---

Tabela 2 Tip entiteta sankcija

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	RB_SANKCIJE	Identifikacioni broj sankcije	INTEGER	5	⊥	d>0
2.	DATUM_OD_SPROVODJENJA	Datum od kog se sprovodi sankcija	DATE		T	Δ
3.	DATUM_DO_SPROVODJENJA	Datum do kog se sprovodi sankcija	DATE		T	Δ
Ključ	K={ID_ZAPOSLENOG+RB_SANKCIJE}					

Tabela 3 Tip poveznika je_sankcija

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
Ključ	K={ID_ZAPOSLENOG+RB_SANKCIJE}					

Tabela 4 Tip entiteta tip_sankcije

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	ID_TIP_SANKCIJE	Identifikacioni broj tipa sankcije	INTEGER	5	⊥	d>0
2.	OPIS_TIPA_SANKCIJE	Opis tipa sankcije(ukidanje slobodnih dana, odbijanje od plate)	VARCHAR	30	⊥	Δ
3.	IZNOS_SANKCIJE	Iznos novca ilibroja dana sankcije	VARCHAR	10	⊥	d>0
Ključ	K={ID_TIP_SANKCIJE}					

Tabela 5 Tip entiteta nagrade

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	RB_NAGRADE	Identifikacioni broj nagrade	INTEGER	5	⊥	d>0
2.	IZNOS_NAGRADE	Iznos konkretne nagrade dodeljene radniku(novčani iznos,broj slobodnih dana)	VARCHAR	30	⊥	Δ
3.	DATUM_DOBIJANJA_NAGRADE	Datum kada je nagrada dodeljena	DATE		T	Δ
Ključ	K={ID_ZAPOSLENOG+RB_NAGRADE}					

Tabela 6 Tip poveznika je_nagrada

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
Ključ	K={ID_ZAPOSLENOG+RB_NAGRADE}					

Tabela 7 Tip entiteta tip_nagrada

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	ID_TIP_NAGRADE	Identifikacioni broj tipa nagrade	INTEGER	5	⊥	d>0
2.	OPIS_TIPA_NAGRADE	Opis tipa nagrade(dodatakna platu, dodeljivanje slobodnih dana, letovanje i slično)	VARCHAR	50	⊥	Δ
3.	OZNAKA	Oznaka da li je individualni, grupni ili organizacioni tip nagrade	VARCHAR	1	⊥	d = {'I','G','O'}
4.	VRSTA	Da li se radi o direktnom ili indirektnom tipu nagrade	VARCHAR	10	⊥	d = {'direktna', 'indirektna'}
Ključ	K={ID_TIP_NAGRADE}					

Tabela 8 Tip poveznika obavlja

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	POSTIGNUTI_REZ	Uspešnost savladavanja obuke iskazana uvidu ocene	INTEGER	3	T	$d \geq 0$ A $d \leq 100$
Ključ	K={ ID_ZAPOSLENOG + ID_TIP_OBUKE + RB_OBUKE }					

Tabela 9 Tip entiteta obuka

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	RB_OBUKE	Identifikacionibroj obuke	INTEGER	5	\perp	$d > 0$
2.	DATUM_POC_OBUKE	Datum kada je obuka počela	DATE		\perp	Δ
3.	DATUM_ZAV_OBUKE	Datum kada je obuka završena	DATE		\perp	Δ
4.	BROJ_CASOVA	Ukupan broj casova trajanja obuke	INTEGER	3	T	$d \geq 0$ A $d \leq 100$
Ključ	K = {ID_TIP_OBUKE +RB_OBUKE}					

Tabela 10 Tip entiteta tip_obuke

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	ID_TIP_OBUKE	Identifikacionibroj tipa obuke	INTEGER	5	\perp	$d > 0$
2.	OPIS_TIPA_OBUKE	Detaljniji opis tipa obuke	VARCHAR	300	\perp	Δ
Ključ	K={ID_TIP_OBUKE}					

Tabela 11 Tip entiteta zahtev_za_odsustvo

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	RB_ZAHTEVA_ODS	Identifikacioni broj zahteva za odsustvo	INTEGER	5	⊥	d>0
2.	DATUM_POC_ODS	Datum početka odsustva	DATE		⊥	Δ
3.	DATUM_ZAV_ODS	Datum završetka odsustva	DATE		⊥	Δ
4.	PRIORITET	Prioritet zahteva za odsustvo (1-urgentno, 2-hitno, 8-nizak..)				
Ključ	K = {ID_ZAPOSLENOG+RB_ZAHTEVA_ODS}					

Tabela 12 Tip poveznika je_tip_zahteva

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
Ključ	K = {ID_ZAPOSLENOG+RB_ZAHTEVA_ODS}					

Tabela 13 Tip entiteta tip_zahteva_za_odsustvo

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	ID_TIP_ZAHTEVA_ODS	Identifikacioni broj tipa zahteva za odsustvo	INTEGER	5	⊥	d>0
2.	OPIS_TIPA_ZAHTEVA_ODS	Opis tipa zahteva za odsustvo	VARCHAR	45	⊥	Δ
Ključ	K = {ID_TIP_ZAHTEVA_ODS}					

Tabela 14 Tip poveznika ima_vrednost

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
Ključ	K = {ID_ZAPOSLENOG+RB_ZAHTEVA_ODS}					

Tabela 15 Tip entiteta status

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	ID_STATUSA	Identifikacioni broj statusa	INTEGER	5	⊥	d>0
2.	OPIS_STATUSA	Opis statusa zahteva za odsustvo	VARCHAR	40	⊥	Δ
Ključ	K= {ID_STATUSA }					

Tabela 16 Tip entiteta projektni_menadzer

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
Ključ	K={PROJ_MENADZER}					

Tabela 17 Tip entiteta radnik

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
Ključ	K={ID_RADNIKA}					

Tabela 18 Tip entiteta HR_menadzer

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
Ključ	K={HR_MENADZER}					

5.RELACIONI MODEL

U ranim godinama baza podataka, svaka aplikacija je čuvala podatke u svojoj jedinstvenoj strukturi. Kada su programeri želeli da naprave aplikacije za korišćenje tih podataka, morali su da znaju mnogo o određenoj strukturi podataka da bi pronašli podatke koji su im potrebni. Ove strukture podataka su bile neefikasne, teške za održavanje i teško optimizovane za isporuku dobrih performansi aplikacije. Model relacione baze podataka je dizajniran da reši problem više proizvoljnih struktura podataka. Relacioni model podataka je obezbedio standardni način predstavljanja i upita podataka koji bi mogla da koristi bilo koja aplikacija.

Od samog početka, programeri su prepoznali da je glavna snaga modela relacione baze podataka u upotrebi tabela, koje su bile intuitivan, efikasan i fleksibilan način za skladištenje i pristup strukturiranim informacijama.

Relacioni model za upravljanje bazom podataka je apstraktni model koji se koristi za organizovanje i upravljanje podacima uskladištenim u bazi podataka. U ovom modelu, podaci su organizovani u dvodimenzionalnim međusobno povezanim tabelama, takođe poznatim kao **relacije** u kojima svaki red(**torka**) predstavlja entitet, a svaka kolona predstavlja svojstva(**atribute**) entiteta.

Relacioni model znači da su logičke strukture podataka — tabele podataka, prikazi i indeksi — odvojene od fizičkih struktura skladištenja. Ovo razdvajanje znači da administratori baze podataka mogu upravljati fizičkim skladištenjem podataka bez uticaja na pristup tim podacima kao logičkoj strukturi. Na primer, preimenovanje datoteke baze podataka ne preimenuje tabele koje se nalaze u njoj.

Upotreba tabela za skladištenje podataka pružila je jednostavan, efikasan i fleksibilan način za skladištenje i pristup strukturiranim informacijama. Zbog ove jednostavnosti, ovaj model podataka omogućava lako sortiranje podataka i pristup podacima. Zbog toga se široko koristi širom sveta za skladištenje i obradu podataka.

Baza podataka implementirana i organizovana u smislu relacionog modela poznata je kao sistem za upravljanje **relacionim bazama podataka** (RDBMS). Dakle, relacioni model opisuje kako se podaci čuvaju u relacionim bazama podataka.

Relacioni model podataka u ovom projektu je sledeći:

zaposleni ({ID_ZAPOSLENOG, Z_JMBG, Z_IME, Z_PREZIME, Z_POL, Z_DAT_RODJ, Z_ADRESA, Z_GRAD, Z_DRZV, Z_BROJ_TEL, Z_EMAIL, Z_DATUM_ZAP, Z_PLT}, {ID_ZAPOSLENOG})

sankcija ({RB_SANKCIJE, DATUM_OD_SPROVODJENJA, DATUM_DO_SPROVODJENJA, ID_ZAPOSLENOG, ID_TIP_SANKCIJE}, {ID_ZAPOSLENOG + RB_SANKCIJE})

sankcija [ID_ZAPOSLENOG] \subseteq *zaposleni* [ID_ZAPOSLENOG]

tip_sanckije ({ID_TIP_SANKCIJE, OPIS_TIPA_SANKCIJE, IZNOS_SANKCIJE}, {ID_TIP_SANKCIJE})

sankcija [ID_TIP_SANKCIJE] \subseteq *tip_sanckije* [ID_TIP_SANKCIJE]

Null (*sankcija*, ID_TIP_SANKCIJE) = \perp

nagrada ({RB_NAGRADE, IZNOS_NAGRADE, DATUM_DOBIJANJA_NAGRADE, ID_ZAPOSLNOG, ID_TIP_NAGRADE}, {ID_ZAPOSLENOG + RB_NAGRADE})

nagrada [ID_ZAPOSLENOG] \subseteq *zaposleni* [ID_ZAPOSLENOG]

tip_nagrade ({ID_TIP_NAGRADE, OPIS_TIPA_NAGRADE, OZNAKA, VRSTA}, {ID_TIP_NAGRADE})

nagrada [ID_TIP_NAGRADE] \subseteq *tip_nagrade* [ID_TIP_NAGRADE]

Null(*nagrada*, ID_TIP_NAGRADE) = \perp

tip_obuke ({ID_TIP_OBUKE, OPIS_TIPA_OBUKE}, {ID_TIP_OBUKE})

obuka ({RB_OBUKE, DATUM_POC_OBUKE, DATUM_ZAV_OBUKE, BROJ_CASOVA, ID_TIP_OBUKE}, {ID_TIP_OBUKE + RB_OBUKE})

$obuka[ID_TIP_OBUKE] \subseteq tip_obuke[ID_TIP_OBUKE]$

$obavlja (\{POSTIGNUTI_REZ, ID_ZAPOSLENOG, ID_TIP_OBUKE, RB_OBUKE\},$
 $\{ID_ZAPOSLENOG + ID_TIP_OBUKE + RB_OBUKE\})$

$obavlja [ID_ZAPOSLENOG] \subseteq zaposleni [ID_ZAPOSLENOG]$

$obavlja [ID_TIP_OBUKE + RB_OBUKE] \subseteq obuka [ID_TIP_OBUKE +$
 $RB_OBUKE]$

$zahtev_za_odsustvo(\{RB_ZAHTEVA_ODS, DATUM_POC_ODS,$
 $DATUM_ZAV_ODS, ID_ZAPOSLENOG, ID_STATUS,$
 $ID_TIP_ZAHTEVA_ODS, HR_MENADZER\},$
 $\{ID_ZAPOSLENOG + RB_ZAHTEVA_ODS\})$

$zahtev_za_odsustvo [ID_ZAPOSLENOG] \subseteq zaposleni [ID_ZAPOSLENOG]$

$zahtev_za_odsustvo [HR_MENADZER] \subseteq HR_menadzer [HR_MENADZER]$

$tip_zahteva_za_odsustvo(\{ID_TIP_ZAHTEVA_ODS,$
 $OPIS_TIPA_ZAHTEVA_ODS\},$
 $\{ID_TIP_ZAHTEVA_ODS\})$

$zahtev_za_odsustvo [ID_TIP_ZAHTEVA_ODS] \subseteq tip_zahteva_za_odsustvo$
 $[ID_TIP_ZAHTEVA_ODS]$

$Null (zahtev_za_odsustvo, ID_TIP_ZAHTEVA_ODS) = \perp$

$status(\{ID_STATUSA, OPIS_STATUSA\},$
 $\{ID_STATUSA\})$

$zahtev_za_odsustvo [ID_STATUS] \subseteq status [ID_STATUS]$

$Null (zahtev_za_odsustvo, ID_STATUSA) = \perp$

$hr_menadzer (\{HR_MENADZER\}, \{HR_MENADZER\})$

$dom (HR_MENADZER) \subseteq dom (ID_ZAPOSLENOG)$

$HR_menadzer [HR_MENADZER] \subseteq zaposleni [ID_ZAPOSLENOG]$

projektni_menadzer ($\{PROJ_MENADZER\}, \{PROJ_MENADZER\}$)

$dom (PROJ_MENADZER) \subseteq dom (ID_ZAPOSLENOG)$

projektni_menadzer [$PROJ_MENADZER$] \subseteq *zaposleni* [$ID_ZAPOSLENOG$]

radnik ($\{ID_RADNIKA\}, \{ID_RADNIKA\}$)

$dom (ID_RADNIKA) \subseteq dom (ID_ZAPOSLENOG)$

radnik [$ID_RADNIKA$] \subseteq *zaposleni* [$ID_ZAPOSLENOG$]

6.DDL

Jezik definicije podataka ili definicija podataka (DDL) je sintaksa koja omogućava kreiranje i modifikovanje objekata baze podataka kao što su korisnici baze podataka, šeme, tabele, pogledi, kolone, funkcije i kreiranje uskladištenih procedura. DDL iskazi su slični računarskom programskom jeziku za definisanje struktura podataka, posebno šema baze podataka. Uobičajeni primeri DDL naredbi uključuju CREATE, ALTER i DROP.

DDL ima za cilj da uspostavi i modifikuje strukturu objekata uskladištenih u bazi podataka. Ove definicije i modifikacije kontrolišu opise šeme baze podataka. One se razlikuju od komandi jezika za manipulaciju podacima (DML) koje se koriste u svrhe modifikacije podataka jer se DDL komande koriste za promenu strukture baze podataka. Primeri ovoga su kreiranje novih tabela ili objekata zajedno sa svim njihovim atributima kao što su tip podataka, ime tabele itd.

Create

Ova komanda pravi novu tabelu i ima unapred definisanu sintaksu.

Sintaksa naredbe create je:

**CREATE TABLE [ime tabele]
([definicije kolone]) [parametri tabele]);**

Na primer:

```
create table Nagrade_sankcije_obuka.Tip_nagrada
(id_tip_nagrada numeric(5) not null,
opis_tipa_nagrada varchar(300) not null,
vrsta varchar not null check(vrsta in('i','o','g')),
oznaka varchar(10) not null check(oznaka='direktna' or oznaka='indirektna'),
constraint PK_Tip_nagrada primary key(id_tip_nagrada)
)
```

Slika 3 Kreiranje tebele DDL naredbom CREATE

Alter

Alter komanda menja postojeću tabelu baze podataka. Ova komanda može dodati dodatne kolone, izbaciti postojeće kolone i čak promeniti tip podataka kolona uključenih u tabelu baze podataka.

Sintaksa alter komande je:

ALTER tipa objekta parametri imena objekta;

Na primer:

```
alter table Nagrade_sankcije_obuka.Status  
add constraint DFT_Status_opis_statusa default('Nepoznat')
```

Slika 4 Izmena tabele DDL naredbom ALTER

Drop

Komanda drop se koristi za brisanje objekata kao što su tabela, indeks ili prikaz. Naredba DROP se ne može vratiti, tako da kada se objekat uništi, nema načina da se oporavi.

Sintaksa naredbe drop je:

DROP tipa objekta ime objekta;

Na primer:

```
if object_id('Nagrade_sankcije_obuka.Zaposleni') is not null  
drop table Nagrade_sankcije_obuka.Zaposleni
```

Slika 5 Brisanje tabele DDL naredbom DROP

U ovom primeru brišemo tabelu Zaposleni.

Truncate

Slično DROP, naredba TRUNCATE se koristi za brzo uklanjanje svih zapisa iz tabele. Međutim, za razliku od DROP-a koji potpuno uništava tabelu, TRUNCATE čuva svoju punu strukturu da bi se kasnije ponovo koristio.

Sintaksa skraćenog iskaza je:

TRUNCATE TABLE ime_tabele;

Na primer:

```
truncate table Nagrade_sankcije_obuka.Sankcija;
```

Slika 6 Uklanjanje zapisa iz tabele DDL naredbom TRUNCATE

Druge često korišćene komande uključuju **RENAME** i **COMMENT**. Prva se koristi sa naredbom **ALTER TABLE** za promenu imena objekta (tabela, kolona, itd.). **COMMENT** se koristi za dodavanje komentara u jednom redu, više redova i komentara.

DDL skript projekta se nalazi u prilogu.

7.DML

Jezik za manipulaciju podacima (DML) računarski jezika koji uključuje komande koje dozvoljavaju korisnicima da manipulišu podacima u bazi podataka. Ova manipulacija uključuje umetanje podataka u tabele baze podataka, preuzimanje postojećih podataka, brisanje podataka iz postojećih tabela i modifikovanje postojećih podataka. DML je uglavnom ugrađen u SQL baze podataka.

DML poboljšava efikasnu interakciju korisnika sa sistemom. Funkcionalna sposobnost DML-a je organizovana u komandama za manipulaciju kao što su SELECT, UPDATE, INSERT INTO i DELETE, FROM.

Select

Ova komanda se koristi za preuzimanje redova iz tabele. Sintaksa je SELECT [naziv(a) kolone] od [ime tabele] gde je [uslovi].

SELECT je najčešće korišćena DML komanda u SQL-u.

Primer:

```
select rb_nagrade,iznos_nagrade,datum_dobijanja_nagrade,id_zaposlenog,id_tip_nagrade  
from Nagrade_sankcije_obuka.Nagrada
```

Slika 7 Primer selekcija podataka iz tabele

Update

Ova komanda menja podatke jednog ili više zapisa. Sintaksa komande za ažuriranje je

UPDATE [ime tabele] SET [ime kolone = vrednost] gde je [uslov].

Primer:

```
update Nagrade_sankcije_obuka.Zaposleni  
set z_pol='m'  
where id_zaposlenog in(21)
```

Slika 8 Primer izmene podata u tabeli

Insert

Ova komanda dodaje jedan ili više zapisa u tabelu baze podataka. Sintaksa komande za umetanje je

INSERT INTO [naziv tabele] [kolona(e)] VREDNOSTI [vrednost(e)].

Primer:

```
insert into Nagrade_sankcije_obuka.Tip_sankcije(id_tip_sankcije, opis_tipa_sankcije, iznos_sankcije)
values(1, 'Ukidanje slobodnih dana', '7 dana');
insert into Nagrade_sankcije_obuka.Tip_sankcije(id_tip_sankcije, opis_tipa_sankcije, iznos_sankcije)
values(2, 'Odbijanje od plate', '5000');
insert into Nagrade_sankcije_obuka.Tip_sankcije(id_tip_sankcije, opis_tipa_sankcije, iznos_sankcije)
values(3, 'Prekovremeni rad', '2 sata');
insert into Nagrade_sankcije_obuka.Tip_sankcije(id_tip_sankcije, opis_tipa_sankcije)
values(4, 'Ukor(opomena)');
```

Slika 9 Primer insertovanja podataka u tabelu

Delete

Ova komanda uklanja jedan ili više zapisa iz tabele u skladu sa određenim uslovima. Sintaksa komande za brisanje je

DELETE FROM [ime tabele] gde je [uslov]

Primer:

```
delete Nagrade_sankcije_obuka.Zaposleni
where id_zaposlenog=1
```

Slika 10 Primer brisanja podataka iz tabele

DML skript projekta se nalazi u prilogu.

8.SQL

SQL je skraćenica za strukturirani jezik upita. SQL se koristi za komunikaciju sa bazom podataka. Prema ANSI (Američki nacionalni institut za standarde), to je standardni jezik za sisteme upravljanja relacionim bazama podataka. SQL izrazi se koriste za obavljanje zadataka kao što su ažuriranje podataka u bazi podataka ili preuzimanje podataka iz baze podataka. Neki uobičajeni sistemi za upravljanje relacionim bazama podataka koji koriste SQL su: Oracle, Sibase, Microsoft SQL Server, Access, Ingres, itd.

Iako postoje neki nedostaci SQL-a, kao što su nezgrapni interfejs i neefikasnost troškova, prednosti imaju tendenciju da nadmašuju njegove nedostatke. SQL je izuzetno dostupan na različitim platformama.

- **SQL je prenosiv**, njegova prenosivost čini ga pogodnom opcijom za korisnike, jer mogu da ga prenesu sa jednog uređaja na drugi bez problema.

- **Brzo obrađuje upite**, bez obzira koliko veliki podaci mogu biti, SQL ih može brzo i efikasno preuzeti.

- **Ne zahteva veštinu kodiranja**, zahteva samo korišćenje jednostavnih ključnih reči kao što su *select*, *insert* i *update*.

- **Koristi standardizovani jezik.**

- **Pruža više prikaza podataka.**

Primer SQL upita:

```
select avg(prosek) as 'Prosecan iznos sankcija'
from (select id_zaposlenog, sum(try_cast(iznos_sankcije as int)) as prosek
from Nagrade_sankcije_obuka.Sankcija san join
Nagrade_sankcije_obuka.Tip_sankcije tsan on san.id_tip_sankcije=tsan.id_tip_sankcije
group by id_zaposlenog) as NovaTab
where id_zaposlenog
in(select id_zaposlenog from Nagrade_sankcije_obuka.Zaposleni where z_grad='Trebinje' or z_grad='Beograd')
```

Slika 11 Primer iščitavanja sankcija zaposlenih u Trebinju i Beogradu

Skript sa SQL upitima za ceo projekat se nalazi u prilogu.

9.OBJEKTI

Objekat baze podataka u relacionoj bazi podataka je struktura podataka koja se koristi za skladištenje ili referenciranje podataka. Najčešći predmet sa kojim korisnici komuniciraju su tabele. Ostali objekti su indeksi, uskladištene procedure, sekvence, pogledi i još mnogo toga.

U okviru projekta korišćeni su objekti, poput šema, tabela sekvenci, funkcija, procedura, trigera i kursora. U daljem tekstu biće predstavljano šta je svaki od njih i primeri njihovog kreiranja i upotrebe.

Šema baze podataka definiše kako su podaci organizovani u okviru relacione baze podataka, odnosno smatra se „nacrtom“ baze podataka koji opisuje kako se podaci mogu odnositi na druge tabele ili druge modele podataka. Šema zapravo ne sadrži podatke već logička ograničenja kao što su imena tabela, polja, tipovi podataka i odnosi između ovih entiteta.

Kreiranja i brisanja šeme baze podataka

```
if schema_id('Nagrade_sankcije_obuka') is not null
drop schema Nagrade_sankcije_obuka
go
create schema Nagrade_sankcije_obuka
go
```

Slika 12 Primer kreiranja i brisanja šeme baze podataka

Ovaj proces dizajna šeme baze podataka je takođe poznat kao modeliranje podataka

Tabele su osnovne komponente relacione baze podataka, koje predstavljaju više redova strukturiranih elemenata podataka (koji se nazivaju kolone). Svaka pojedinačna stavka podataka unesena u relaciju bazu podataka je predstavljena vrednošću u koloni.

Kreiranje i brisanje tabele baze podataka

```
create table Nagrade_sankcije_obuka.Obuka
(rb_obuke numeric(5) not null default(next value for Nagrade_sankcije_obuka.ObukaId),
datum_poc_obuke date not null,
datum_zav_obuke date not null,
broj_casova numeric(3) check(len(broj_casova)>0 and len(broj_casova)<=100),
id_tip_obuke numeric(5) not null,
constraint PK_Obuka primary key(id_tip_obuke,rb_obuke),
constraint FK_Obuka_Tip_obuke foreign key(id_tip_obuke) references Nagrade_sankcije_obuka.Tip_obuke(id_tip_obuke),
constraint CHK_Obuka_datum_zav_obuke check(datum_zav_obuke>datum_poc_obuke)
)
```

Slika 13 Primer kreiranja i brisanja tabele baze podataka

Sekvenca je objekat baze podataka koji omogućava korisnicima da generišu jedinstvene celobrojne vrednosti. Sekvenca se povećava svaki put kada se generiše broj sekvence. Povećanje se dešava čak i ako se transakcija vrati, što može dovesti do praznina između brojeva. Slično, praznine mogu nastati kada dva korisnika istovremeno povećavaju istu sekvencu.

Kreiranje i brisanje sekvence baze podataka

```
create sequence Ngrade_sankcije_obuka.ObukaId as numeric
start with 1
minvalue 1
increment by 1
no cycle
```

Slika 14 Slika 13 Primer kreiranja i brisanja sekvence baze podataka

Korisnički definisana funkcija je objekat baze podataka koji se koristi za čuvanje dugog ili komplikovanog upita tako da se može više puta pokretati pomoću jednostavnih komandi. Funkcija može da vrati jednu vrednost pozivaocu ili može da vrati skup rezultata. Korišćenjem parametara, funkcije se takođe mogu učiniti dinamičkim.

Kreiranje i brisanje funkcija baze podataka

```
if object_id('Ngrade_sankcije_obuka.BrojDanaOdsustva') is not null
drop function Ngrade_sankcije_obuka.BrojDanaOdsustva
go
create function Ngrade_sankcije_obuka.BrojDanaOdsustva
(
    @rbZahteva as numeric(5)
)
returns varchar(200)
as
begin
    declare @trajanje as int
    declare @poruka as varchar(100)
    declare @datPoc as date
    declare @datZav as date
    declare @status as varchar(10)

    if (@rbZahteva <= 0)
    begin
        set @poruka = 'Ne postoji zahtev za odsustvo sa unetim rednim brojem: ' + convert(varchar, @rbZahteva)
        return @poruka
    end

    else
    begin
        set @datPoc = (select datum_poc_ods from Ngrade_sankcije_obuka.Zahtev_za_odsustvo where rb_zahteva_ods = @rbZahteva)
        set @datZav = (select datum_zav_ods from Ngrade_sankcije_obuka.Zahtev_za_odsustvo where rb_zahteva_ods = @rbZahteva)
        set @status = (select opis_statusa from Ngrade_sankcije_obuka.Zahtev_za_odsustvo zah
            join Ngrade_sankcije_obuka.Status stat on zah.id_status = stat.id_status where rb_zahteva_ods = @rbZahteva)

        if (@status != 'Odobren')
        begin
            set @poruka = 'Zahtev za odsustvo nije odobren.'
            return @poruka
        end

        else
        begin
            set @trajanje = DATEDIFF(day, @datPoc, @datZav)
            set @poruka = 'Zahtev za odsustvo je odobren. Odsustvo ce trajati ' + convert(varchar, @trajanje) + ' dan/a.'
            return @poruka
        end
    end
end
return @poruka
end
go
```

Slika 15 Primer kreiranja i brisanja funkcija baze podataka

Skladištena procedura je blok SQL naredbi u sistemu za upravljanje relacionim bazama podataka i obično je piše programer, administrator baze podataka, analitičar podataka, a čuva se i ponovo koristi za više programa. Takođe može imati različite tipove u zavisnosti od različitih RDBMS-a. Međutim, dve najvažnije uskladištene procedure koje se nalaze u bilo kom RDBMS-u su: -korisnički definisana uskladištena procedura

-sistemska uskladištena procedura

Sačuvana procedura omogućava da se upit napiše jednom, a upit ima ime koje se može sačuvati i izvršiti više puta ako je potrebno.

Kreiranje i brisanje korisnički definisane procedure baze podataka

```

~/
if object_id('Nagrada_sankcije_obuka.ImejlZaObuku') is not null
drop procedure Nagrade_sankcije_obuka.ImejlZaObuku
go
:create procedure Nagrade_sankcije_obuka.ImejlZaObuku
@idZap as numeric(5)
as
:begin
declare @brZap as numeric(5)=(select count(id_zaposlenog) from Nagrade_sankcije_obuka.Zaposleni where id_zaposlenog=@idZap)
declare @ime as varchar(20)=(select z_ime from Nagrade_sankcije_obuka.Zaposleni where id_zaposlenog=@idZap)
declare @prez as varchar(20)=(select z_prezime from Nagrade_sankcije_obuka.Zaposleni where id_zaposlenog=@idZap)
declare @grad as varchar(20)=(select z_grad from Nagrade_sankcije_obuka.Zaposleni where id_zaposlenog=@idZap)
declare @mej1 as varchar(100)
declare @ekstenzija as varchar(40)
:if @brZap=0
3 begin
print 'Zaposleni sa id-jem: '+convert(varchar,@idZap)+' ne postoji u bazi.'
end
-
else
3 begin
if(@grad='Beograd')
set @ekstenzija='@bgobuka.com'
else if(@grad='Novi Sad')
set @ekstenzija='@nsobuka.com'
else if(@grad='Trebinje')
set @ekstenzija='@tbobuka.com'
else if(@grad='Prijevor')
set @ekstenzija='@pdobuka.com'
else if(@grad='Kragujevac')
set @ekstenzija='@kgobuka.com'
else
3 begin
set @ekstenzija='@'+lower(replace(@grad, ' ', ''))+'.obuka.com'
end
-
set @mej1=lower(@prez)+'_'+lower(@ime)+'_'+convert(varchar,@idZap)+convert(varchar,@ekstenzija)

3 update Nagrade_sankcije_obuka.Zaposleni
set z_email=@mej1
where id_zaposlenog=@idZap

3 print 'Imejl zaposlenog '+concat(@ime, ' ',@prez)+' sa id-jem '+convert(varchar,@idZap)+' iz '+@grad+
' je postavljen na: '+ @mej1
-
end
end
go

```

Slika 16 Primer kreiranja i brisanja procedure baze podataka

Trigeri ili okidači, su slični uskladištenim procedurama. Okidač može uključiti SQL i PL/SQL naredbe koje se izvršavaju kao jedinica i može pozvati uskladištene procedure. Međutim, procedure i okidači se razlikuju po načinu na koji se pozivaju. Dok proceduru eksplicitno izvršava korisnik, jedan ili više okidača pokreće (izvršava) kada se izda naredba INSERT, UPDATE ili DELETE, bez obzira na to koji korisnik je povezan ili koja je aplikacija se koristi.

Kreiranje i brisanje trigera baze podataka

```
on Nagrade_sankcije_obuka.Oblavlja
after insert
as
begin
if @@ROWCOUNT=0
return;
declare @postignutiRez as numeric(3)=(select postignuti_rez from inserted)
declare @idZap as numeric
declare @imeZ as varchar(20)
declare @prezZ as varchar(20)
declare @plata as numeric(30)
declare @postignuti_rez as numeric(20)
declare @rbObuke as numeric(5)=(select rb_obuke from inserted)
if @postignutiRez > =90
begin
declare curosor_plata cursor for
select zap.id_zaposlenog,z_ime,z_prezime,z_plt,postignuti_rez
from Nagrade_sankcije_obuka.Zaposleni zap join Nagrade_sankcije_obuka.Oblavlja
obv on zap.id_zaposlenog=obv.id_zaposlenog
where rb_obuke=@rbObuke

open curosor_plata

fetch next from curosor_plata into @idZap,@imeZ,@prezZ,@plata,@postignuti_rez

while @@FETCH_STATUS=0
begin
if (@idZap=(select projektni_menadzer from
Nagrade_sankcije_obuka.Projektni_menadzer where projektni_menadzer=@idZap))
begin
update Nagrade_sankcije_obuka.Zaposleni
set z_plt=z_plt+7000
where id_zaposlenog=@idZap
end
else if (@idZap=(select hr_menadzer from
Nagrade_sankcije_obuka.Hr_menadzer where hr_menadzer=@idZap))
begin
update Nagrade_sankcije_obuka.Zaposleni
set z_plt=z_plt+5000
where id_zaposlenog=@idZap
end

declare @novaPlata as numeric(10)=(select z_plt from Nagrade_sankcije_obuka.Zaposleni
where id_zaposlenog=@idZap)
print (@imeZ+' '+@prezZ+', zarada pre izmene:'+try_cast(@plata as varchar)
+', zarada posle izmene:'+try_cast(@novaPlata as varchar))

fetch next from curosor_plata into @idZap,@imeZ,@prezZ,@plata,@postignuti_rez

end
end
```

Slika 17 Primer kreiranja i brisanja trigera baze podataka

Kursor baze podataka može se smatrati pokazivačem na određeni red u okviru rezultata upita. Pokazivač se može pomerati iz jednog reda u drugi, a z kod nekih je moguće pomeranje i na prethodni red. Kursor prati poziciju u skupu rezultata i omogućava izvršavanje više operacija red po red protiv skupa rezultata, sa ili bez vraćanja na originalnu tabelu.

Kreiranje i uklanjanje kursora u okviru baze podataka

```
declare zahtev_kursor cursor for
select rb_zahteva_ods,zap.z_ime,zap.z_prezime,zap.id_zaposlenog,opis_tipa_zahteva_ods,Nagrade_sankcije_obuka.prioritet_funkcija(rb_zahteva_ods
from Nagrade_sankcije_obuka.Zaposleni zap join Nagrade_sankcije_obuka.Zahtev_zah_odsustvo zah on zap.id_zaposlenog=zah.id_zaposlenog
join Nagrade_sankcije_obuka.Tip_zahteva_zah_odsustvo tn on zah.id_tip_zahteva_ods=tn.id_tip_zahteva_ods
where hr_menadzer=@idMen and id_status=(select id_status from Nagrade_sankcije_obuka.Status where opis_statusa='Odobren')
open zahtev_kursor
fetch next from zahtev_kursor into @rbZah,@imeZap,@prezZap,@idZap,@zahtev,@prioritet,@datumP
while @@FETCH_STATUS=0
begin

print 'Zaposlenom '+@imeZap+' '+@prezZap+' sa id-jem '+convert(varchar,@idZap)+
' odobren je zahtev pod rednim brojem '+convert(varchar,@rbZah)+' '+@zahtev+' prioriteta '+
@prioritet+' sa pocetkom od: '+convert(varchar,@datumP)

fetch next from zahtev_kursor into @rbZah,@imeZap,@prezZap,@idZap,@zahtev,@prioritet,@datumP

end
close zahtev_kursor
deallocate zahtev_kursor
```

Slika 18 Primer kreiranja i uklanjanja kursora u okviru baze podataka

Uz dokumentaciju priloženi su i

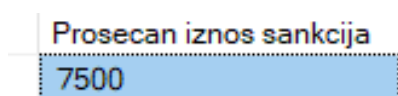
- script za kreiranje i brisanje šema i tabela projekta,
- script za insert podataka projekta,
- script sa SQL upitima projekta,
- script za kreiranje funkcija projekta,
- script za kreiranje procedura projekta,
- script za kreiranje trigera projekt

10.UPITI

Prvi upit

Upit prikazuje prosečan iznos sankcija za zaposlene koji žive u Trebinju i Beogradu.

```
select avg(prosek) as 'Prosecan iznos sankcija'
from (select id_zaposlenog, sum(try_cast(iznos_sankcije as int)) as prosek
from Nagrade_sankcije_obuka.Sankcija san join
Nagrade_sankcije_obuka.Tip_sankcije tsan on
san.id_tip_sankcije=tsan.id_tip_sankcije
group by id_zaposlenog) as NovaTab
where id_zaposlenog
in(select id_zaposlenog from Nagrade_sankcije_obuka.Zaposleni where
z_grad='Trebinje' or z_grad='Beograd')
```



Prosecan iznos sankcija
7500

Slika 19 Rezultat prvog upita

Drugi upit

Upit izlistava prosečan broj bodova postignutih rezultata po seminaru kao i celokupan rezultat jednog seminara, koji zaposleni pohađa, prebrojava koliko zaposlenih je bilo na seminarima koji su trajali od 22-10-2021 do 15-11-2021. Ispis je sortiran po prosečnoj oceni od najbolje do najlošije postignutih ukupnih prosečnih rezultata.

```
select obv.rb_obuke as 'Redni broj obuke',BrojZap as 'Broj zaposlenih na
seminaru',UkupniRez/BrojZap as 'Prosek ostvarenog rezultata',UkupniRez as
'Ukupan rezultat celog tima'
from Nagrade_sankcije_obuka.Obavlja obv join
((select ob.rb_obuke,count(id_zaposlenog) BrojZap,sum(postignuti_rez)
UkupniRez
from Nagrade_sankcije_obuka.Obuka ob join
Nagrade_sankcije_obuka.Obavlja obv
on ob.rb_obuke=obv.rb_obuke where datum_poc_obuke='2021-10-22' group by
ob.rb_obuke)union
(select ob.rb_obuke,count(id_zaposlenog) BrojZap, sum(postignuti_rez)
UkupniRez
from Nagrade_sankcije_obuka.Obuka ob join
Nagrade_sankcije_obuka.Obavlja obv
on ob.rb_obuke=obv.rb_obuke where datum_zav_obuke='2021-11-15'
group by ob.rb_obuke)) novaT on novaT.rb_obuke=obv.rb_obuke
group by obv.rb_obuke,BrojZap,UkupniRez
order by UkupniRez/BrojZap desc
```

	Redni broj obuke	Broj zaposlenih na seminaru	Prosek ostvarenog rezultata	Ukupan rezultat celog tima
1	1	10	95.100000	951
2	13	2	91.000000	182
3	2	10	89.100000	891

Slika 20 Rezultat drugog upita

Treći upit

Izlistavaju se podaci o imenu, prezimenu, id-u zaposlenih iz Novog Sada koji imaju platu veću od prosečne plate svih radnika koji su podneli zahtev za odsustvo, prikazuje se i prioriteti zahteva opisno kao i njihov status.

Opis prioriteta zahteva može da ima vrednost:

- 'Hitno' – za zahtev čiji je prioritet 1,
- 'Urgentno' – za zahtev čiji je prioritet između 2. i 4,
- 'Redovno' – za zahtev čiji je prioritet između 5 i 7,
- 'Nizak prioritet' – za zahtev čiji je prioritet između 8 i 10,
- 'Nepoznato' – za zahtev čiji je prioritet nepoznat.

```
select z_ime as Ime, z_prezime as Prezime, zap.id_zaposlenog as "ID
zaposlenog", opis_tipa_zahteva_ods as
"Tip zahteva za odsustvo", opis_statusa as Status,
((select 'Urgentno' where prioritet=1) union (select 'Hitno' where
prioritet between 2 and 4)
union (select 'Redovno' where prioritet between 5 and 7) union (select
'Nizak prioritet' where prioritet between 8 and 10) union (select
'Nepoznato' where prioritet is null)) as Prioritet
from Nagrade_sankcije_obuka.Zaposleni zap join
Nagrade_sankcije_obuka.Zahtev_za_odsustvo zah
on zap.id_zaposlenog=zah.id_zaposlenog join
Nagrade_sankcije_obuka.Tip_zahteva_za_odsustvo tzah
on zah.id_tip_zahteva_ods=tzah.id_tip_zahteva_ods join
Nagrade_sankcije_obuka.Status stat on zah.id_status=stat.id_status
where z_grad='Novi Sad' and z_plt>(select avg(z_plt) from
Nagrade_sankcije_obuka.Zaposleni)
group by
z_ime,z_prezime,zap.id_zaposlenog,opis_tipa_zahteva_ods,prioritet,opis_sta
tusa,z_plt
```

	Ime	Prezime	ID zaposlenog	Tip zahteva za odsustvo	Status	Prioritet
1	Miroslav	Mirkovic	22	Zahtev za bolovanje	Odobren	Redovno

Slika 21 Rezultat trećeg upita

Četvrti upit

Izlistavaju se zaposleni kojima ime počinje karakterima 'A', 'S' ili 'M', sa mestom stanovanja u Novom Sadu ili Beogradu. Podaci koji su prikazani su: ime, kontakt telefon, mesto i adresa zaposlenog. Takođe, u okviru kolone Nagrade je prikazano da li je zaposleni dobio nagradu u poslednjih 6 godina, i ako jeste prikazaće se tip i vrsta nagrade i datum kada je dobijena nagrada, a ukoliko nije ispisati poruku "Nema nagrade". Vrsta nagrade se ispisuje kao:

- 'INDIVIUDALNA' – za nagrade čija je vrsta označena sa i,
- 'GRUPNA' – za nagrade čija je vrsta označena sa g,
- 'ORGANIZACIONA' – za nagrade čija je vrsta označena sa o.

Izlistavaju se samo oni zaposleni koji zarađuju manje od nekog drugog zaposlenog koji je hr menadžer.

```
select z_ime as Ime, z_broj_tel as 'Kontakt telefon', z_grad as Grad,
z_adresa as Adresa,
if(opis_tipa_nagrade is not null,
replace(LEFT(opis_tipa_nagrade, CHARINDEX('-', opis_tipa_nagrade)), '-', '
'), 'Nema nagrade')
as Nagrade , datum_dobijanja_nagrade as 'Datum dobijanja nagrade',
case
when vrsta='i' then 'INDIVIDUALNA'
when vrsta='o' then 'ORGANIZACIONA'
when vrsta='g' then 'GRUPNA'
end as Vrsta
from Nagrade_sankcije_obuka.Zaposleni zap join
Nagrade_sankcije_obuka.Nagrada nag on zap.id_zaposlenog=nag.id_zaposlenog
join Nagrade_sankcije_obuka.Tip_nagrade tn on
nag.id_tip_nagrade=tn.id_tip_nagrade
where z_grad in('Novi Sad','Beograd') and (z_ime like 'A%' or z_ime
like 'S%' or z_ime like 'M%')
and (year(datum_dobijanja_nagrade) between year(getDate())-6 and
year(getDate())) and (z_plt<any(select z_plt
from Nagrade_sankcije_obuka.Zaposleni zap
join Nagrade_sankcije_obuka.Hr_menadzer hrm on
zap.id_zaposlenog=hrm.hr_menadzer))
```

	Ime	Kontakt telefon	Grad	Adresa	Nagrade	Datum dobijanja nagrade	Vrsta
1	Sandra	066/5193312	Novi Sad	Dr Sime Milosevica	Bonusi	2022-01-05	INDIVIDUALNA
2	Marija	065/5244236	Novi Sad	Novosadskog sajma	Bonusi	2022-01-05	INDIVIDUALNA
3	Anja	066/4183341	Novi Sad	Bulevar Cara Lazara	Vauceri	2022-01-05	INDIVIDUALNA

Slika 22 Rezultat četvrtog upita

Peti upit

Izlistavaju se podaci o zaposlenima kao id zaposlenog, ime, godina zaposlenja, radni staž za najbolje plaćene radnike među hr menadžerima i projektnim menadžerima I njihove plate, koji su osvojili neku od indirektnih nagrada, kao i podatke o tim nagradama, koji su sortirani srazmerno rastućuj plati.

```
select zap.id_zaposlenog as 'Id zaposlenog', z_ime as Ime,
year(zap.z_datum_zap) as 'Godina zaposlenja',
concat(DATEDIFF(year, z_datum_zap, getDate()), 'g.') as 'Radni staz',
replace(LEFT(opis_tipa_nagrade, CHARINDEX('-', opis_tipa_nagrade)), '-', '
') as Nagrada, oznaka as 'Oznaka nagrade', maxPlt as 'Najveca plata'
from Nagrade_sankcije_obuka.Zaposleni zap join
Nagrada_sankcije_obuka.Nagrada nag on zap.id_zaposlenog=nag.id_zaposlenog
join Nagrade_sankcije_obuka.Tip_nagrade tn on
nag.id_tip_nagrade=tn.id_tip_nagrade join
((select nag.id_tip_nagrade, max(zap.z_plt) as maxplt
from Nagrade_sankcije_obuka.Nagrada nag join
Nagrada_sankcije_obuka.Hr_menadzer hrm on
nag.id_zaposlenog=hrm.hr_menadzer join Nagrade_sankcije_obuka.Zaposleni
zap on zap.id_zaposlenog=hrm.hr_menadzer
join Nagrade_sankcije_obuka.Tip_nagrade tn on
nag.id_tip_nagrade=tn.id_tip_nagrade
where oznaka='indirektna'
group by nag.id_tip_nagrade)
union
(select nag.id_tip_nagrade, max(zap.z_plt) as maxPlt
from Nagrade_sankcije_obuka.Nagrada nag join
Nagrada_sankcije_obuka.Projektni_menadzer prm on
nag.id_zaposlenog=prm.projektni_menadzer join
Nagrada_sankcije_obuka.Zaposleni zap on
zap.id_zaposlenog=prm.projektni_menadzer
join Nagrade_sankcije_obuka.Tip_nagrade tn on
nag.id_tip_nagrade=tn.id_tip_nagrade
where oznaka='indirektna'
group by nag.id_tip_nagrade)) maxPlata on
nag.id_tip_nagrade=maxplata.id_tip_nagrade
where zap.z_plt=maxPlt
order by z_plt
```

	Id zaposlenog	Ime	Godina zaposlenja	Radni staz	Nagrada	Oznaka nagrade	Najveca plata
1	12	Milan	2019	3g.	Vauceri	indirektna	85000
2	22	Miroslav	2004	18g.	Zdravstveno osiguranje	indirektna	400000
3	22	Miroslav	2004	18g.	Penziono osiguranje	indirektna	400000
4	22	Miroslav	2004	18g.	Putovanje	indirektna	400000

Slika 23 Rezultat petog upit

11.FUNKCIJE

Prva funkcija-BrojDanaOdsustva

Na osnovu prosledjenog rednog broja zahteva, korisniku se vraća odgovarajuća poruka o zaposlenom koji je tražio odsustvo u trajanju zahtevanog odsustva izraženom u broju dana.

Ako je korisnik uneo nepostojeći redni broj zahteva biće obavešten o tome, u suprotnom korisnik se obaveštava o broju dana trajanja odsustva samo za zahteve koji su odobreni.

```
if object_id('Nagrada_sankcije_obuka.BrojDanaOdsustva') is not null
drop function Nagrade_sankcije_obuka.BrojDanaOdsustva
go
create function Nagrade_sankcije_obuka.BrojDanaOdsustva
(
    @rbZahteva as numeric(5)
)
returns varchar(500)
as
begin
declare @trajanje as int
declare @poruka as varchar(500)
declare @datPoc as date
declare @datZav as date
declare @status as varchar(10)
declare @ime as varchar(20)
declare @prezime as varchar(20)
declare @idZap as numeric(5)
declare @provera as numeric(5)=(select rb_zahteva_ods from
Nagrade_sankcije_obuka.Zahtev_za_odsustvo where rb_zahteva_ods=@rbZahteva)

if(@provera is null)
begin
    set @poruka='Ne postoji zahtev za odsustvo sa unetim rednim
brojem: '+convert(varchar,@rbZahteva)
    return @poruka
end

else
begin
    set @datPoc=(select datum_poc_ods from
Nagrade_sankcije_obuka.Zahtev_za_odsustvo where rb_zahteva_ods=@rbZahteva)
    set @datZav=(select datum_zav_ods from
Nagrade_sankcije_obuka.Zahtev_za_odsustvo where rb_zahteva_ods=@rbZahteva)
    set @status=(select opis_statusa from
Nagrade_sankcije_obuka.Zahtev_za_odsustvo zah
```

```

join Nagrade_sankcije_obuka.Status stat on zah.id_status=stat.id_status
where rb_zahteva_ods=@rbZahteva)
set @ime=(select z_ime from Nagrade_sankcije_obuka.Zahtev_za_odsustvo
zao join Nagrade_sankcije_obuka.Zaposleni zap on
zao.id_zaposlenog=zap.id_zaposlenog where rb_zahteva_ods=@rbZahteva)
set @prezime=(select z_prezime from
Nagrade_sankcije_obuka.Zahtev_za_odsustvo zao join
Nagrade_sankcije_obuka.Zaposleni zap on
zao.id_zaposlenog=zap.id_zaposlenog where rb_zahteva_ods=@rbZahteva)
set @idZap=(select id_zaposlenog from
Nagrade_sankcije_obuka.Zahtev_za_odsustvo where
rb_zahteva_ods=@rbZahteva)
end

if(@status != 'Odobren')
begin
set @poruka='Zahtev za odsustvo sa rednim brojem
'+convert(varchar,@rbZahteva)+' jos uvek nije odobren zaposlenom '+
concat(@ime, ' ',@prezime)+' sa id-jem '+ convert(varchar,@idZap)
return @poruka
end

else
begin
set @trajanje=DATEDIFF(day,@datPoc,@datZav)
set @poruka='Zahtev za odsustvo sa rednim brojem
'+convert(varchar,@rbZahteva)+' je odobren zaposlenom '+
concat(@ime, ' ',@prezime)+' sa id-jem '+ convert(varchar,@idZap)
+'. Odsustvo ce trajati '+ convert(varchar,@trajanje)+' dan/a.'
return @poruka
end

return @poruka
end
go

```

1 Zahtev za odsustvo sa rednim brojem 5 je odobren zaposlenom Marija Petrovic sa id-jem 2. Odsustvo ce trajati 7 dan/a.

Slika 24 Pozitivan rezultat prve funkcije

Zahtev za odsustvo sa rednim brojem 17 jos uvek nije odobren zaposlenom Natasa Jovanovic sa id-jem 5

Slika 25 Negativan rezultat prve funkcije

1 Ne postoji zahtev za odsustvo sa unetim rednim brojem

Slika 26 Negativan rezultat prve funkcije

Druga funkcija-Senioritet

Na osnovu unetog id_zaposlenog, vraća se senioriteta zaposlenog na osnovu njegove plate i broja obuka koje je zaposleni prošao.

Zaposleni je junior ako ima platu između 50 000 i 70 000 i broj završenih obuka manji i jednak od 3,

zaposleni je medior ako ima platu između 80 000 i 250 000 i broj završenih obuka između 5 i 7,

Zaposleni je senior ako ima platu između 255 000 i 400 000 i broj završenih obuka veći i jednak od 8.

```
if object_id('Nagrade_sankcije_obuka.Senioritet') is not null
drop function Nagrade_sankcije_obuka.Senioritet
go
create function Nagrade_sankcije_obuka.Senioritet
(
    @idZap as numeric(5)
)
returns varchar(100)
as
begin
declare @ispis as varchar(100)
declare @plata as numeric(10)
declare @brojObuka as numeric(5)
declare @ime as varchar(20)
declare @prezime as varchar(20)
declare @provera as numeric(5)=(select id_zaposlenog from
Nagrade_sankcije_obuka.Zaposleni where id_zaposlenog=@idZap)
if(@provera is null)
begin
set @ispis='Ne postoji zaposleni sa unetim id='+convert(varchar,@idZap)
return @ispis
end

else
begin
set @plata=(select z_plt from Nagrade_sankcije_obuka.Zaposleni where
id_zaposlenog=@idZap)
set @brojObuka=(select count(id_zaposlenog) from Nagrade_sankcije_obuka.Obavlja
where id_zaposlenog=@idZap)
set @ime=(select z_ime from Nagrade_sankcije_obuka.Zaposleni where
id_zaposlenog=@idZap)
set @prezime=(select z_prezime from Nagrade_sankcije_obuka.Zaposleni where
id_zaposlenog=@idZap)
if(@plata between 50000 and 70000 and @brojObuka<=3)
begin
set @ispis='Zaposleni '+concat(@ime, ' ',@prezime)+' je junior.'
return @ispis
end

else if (@plata between 80000 and 250000 and @brojObuka between 5 and 7)
begin
set @ispis='Zaposleni '+concat(@ime, ' ',@prezime)+' je medior.'
return @ispis
end
end
```

```

else if (@plata between 255000 and 400000 and @brojObuka>=8)
begin
set @ispis='Zaposleni '+concat(@ime, ' ',@prezime)+' je senior.'
return @ispis
end

end
return @ispis
end
go

```

1	Zaposleni Natasa Jovanovic je junior.
---	---------------------------------------

Slika 27 Pozitivan rezultat druge funkcije

1	Zaposleni Marko Rakic je medior.
---	----------------------------------

Slika 28 Pozitivan rezultat druge funkcije

1	Zaposleni Dusan Perisin je senior.
---	------------------------------------

Slika 29 Pozitivan rezultat druge funkcije

1	Ne postoji zaposleni sa unetim id=158
---	---------------------------------------

Slika 30 Negativan rezultat druge funkcije

12.PROCEDURE

Prva procedura- MejlZaObuku

Procedura koja na osnovu prosleđenog id-ja zaposlenog kreira imejl koji se može koristiti za prijavu na sajt za prijavljivanje na neku obuku. Imejl se zasniva od id-ja, imena, prezimena, kao i lokacije gde je klijent prijavljen, te se na osnovu gradova mogu ubuduće grupisati mesta za održavanje obuka(do 10 mesta). Za 5 gradova koji su aktuelni u bazi, unapred su kreirane skraćenice, dok se za ostale uzima puno ime grada (ukoliko se unese novi zaposleni koji je iz Niša, njegov imejl bi imao ekstenziju @nisobuka.com). Unapred kreirane skraćenice po gradovima su:

Beograd -bg, Novi Sad - ns, Trebinje - tb, Prijedor - pd, Kragujevac – kg.

```
if object_id('Nagrade_sankcije_obuka.MejlZaObuku') is not null
drop procedure Nagrade_sankcije_obuka.MejlZaObuku
go
create procedure Nagrade_sankcije_obuka.MejlZaObuku
@idZap as numeric(5)
as
begin
declare @provera as numeric(5)=(select id_zaposlenog from
Nagrade_sankcije_obuka.Zaposleni where id_zaposlenog=@idZap)
declare @ime as varchar(20)=(select z_ime from Nagrade_sankcije_obuka.Zaposleni where
id_zaposlenog=@idZap)
declare @prez as varchar(20)=(select z_prezime from Nagrade_sankcije_obuka.Zaposleni
where id_zaposlenog=@idZap)
declare @grad as varchar(20)=(select z_grad from Nagrade_sankcije_obuka.Zaposleni
where id_zaposlenog=@idZap)
declare @mejl as varchar(100)
declare @ekstenzija as varchar(40)
if @provera is null
begin
print 'Zaposleni sa id-jem: '+convert(varchar,@idZap)+' ne postoji u bazi.'
end
else
begin
if(@grad='Beograd')
set @ekstenzija='@bgobuka.com'
else if(@grad='Novi Sad')
set @ekstenzija='@nsobuka.com'
else if(@grad='Trebinje')
set @ekstenzija='@tbobuka.com'
else if(@grad='Prijedor')
set @ekstenzija='@pdobuka.com'
else if(@grad='Kragujevac')
set @ekstenzija='@kgobuka.com'
else
begin
set @ekstenzija='@'+lower(replace(@grad, ' ', ''))+ 'obuka.com'
end
end
```

```

set
@mejl=lower(@prez)+'_'+lower(@ime)+'_'+convert(varchar,@idZap)+convert(varchar,@eksten
zija)

update Nagrade_sankcije_obuka.Zaposleni
set z_email=@mejl
where id_zaposlenog=@idZap

print 'Imejl zaposlenog sa id-jem '+convert(varchar,@idZap)+' je postavljen na: '+
@mejl
end
end
go

```

```

(1 row affected)
Imejl zaposlenog sa id-jem 1 je postavljen na: lazarevic_sandra_1@nsobuka.com

Completion time: 2022-06-03T19:49:37.9405779+02:00

```

Slika 31 Pozitivan rezultat prve procedure

```

(1 row affected)
Imejl zaposlenog sa id-jem 24 je postavljen na: derikonjic_luka_24@nisobuka.com

Completion time: 2022-06-03T19:50:23.0359733+02:00

```

Slika 32 Pozitivan rezultat prve procedure

Druga procedura- OdobreniZahteviZaOdst

Procedura koja na osnovu prosleđenog id-ja hr menadžera generiše spisak svih zahteva za odsustvo koji je hr menadžer odobrio, uz izlistavanje osnovnih informacija o zahtevu poput opisa i prioriteta zahteva. Takođe, u okviru ispisa se i prilaže datum početka odsustva kao i ime i prezime zaposlenog kom je odobreno odsustvo. Ispisuje se i ukupan broj zahteva koji je hr menadžer odobrio. Ako hr menadžer nije odobrio još uvek ni jedan zahtev, ispisuje se poruka, kao i ako nema podataka o zahtevu.

(U okviru ove procedure koršćen je **kursor** zahtev_kursor koji prolazi kroz torke zahteva za odsustvo koje je hr menadžer odobrio, za potrebe kursora kerirana je i funkcija PrioritetFunkcija koja izlistava opisno prioritet zahteva u zavisnosti od numeričke vrednosti koja je dodeljena zahtevu.)

```
if object_id('Nagrada_sankcije_obuka.OdobreniZahteviZaOdst') is not null
drop procedure Nagrade_sankcije_obuka.OdobreniZahteviZaOdst
go
create procedure Nagrade_sankcije_obuka.OdobreniZahteviZaOdst
@idMen as numeric(5)
as
begin
    declare @provera as numeric(5)=(select hr_menadzer from
Nagrade_sankcije_obuka.Hr_menadzer where hr_menadzer=@idMen)
    declare @imeM as varchar(20)=(select z_ime from Nagrade_sankcije_obuka.Hr_menadzer
hrm join Nagrade_sankcije_obuka.Zaposleni zap on hrm.hr_menadzer=zap.id_zaposlenog
where hr_menadzer=@idMen)
    declare @brojZ as numeric(5)=(select count(rb_zahteva_ods) from
Nagrade_sankcije_obuka.Zahtev_zah_odsustvo zah
where id_status=(select id_status from Nagrade_sankcije_obuka.Status where
opis_statusa='Odobren')
and hr_menadzer=@idMen)

    if @provera is null
    begin
        print 'Hr menadzer sa id-jem '+convert(varchar,@idMen) +' ne postoji u
bazi.'
    end
    else
    begin
        if @brojZ>0
        begin
            print 'Hr menadzer sa id-jem '+convert(varchar,@idMen) +' je odobrio sledece
zahteve za odsustvo:'
            declare @rbZah as numeric(5)
            declare @imeZap as varchar(20)
            declare @prezZap as varchar(20)
            declare @idZap as numeric(5)
            declare @zahtev as varchar(50)
            declare @prioritet as varchar(20)
            declare @datumP as date
            declare zahtev_kursor cursor for
            select
rb_zahteva_ods,zap.z_ime,zap.z_prezime,zap.id_zaposlenog,opis_tipa_zahteva_ods,Nagrade
_sankcije_obuka.PrioritetFunkcija(rb_zahteva_ods),datum_poc_ods
```

```

        from Nagrade_sankcije_obuka.Zaposleni zap join
Nagrade_sankcije_obuka.Zahtev_za_odsustvo zah on zap.id_zaposlenog=zah.id_zaposlenog
        join Nagrade_sankcije_obuka.Tip_zahteva_za_odsustvo tn on
zah.id_tip_zahteva_ods=tn.id_tip_zahteva_ods
        where hr_menadzer=@idMen and id_status=(select id_status from
Nagrade_sankcije_obuka.Status where opis_statusa='Odobren')
        open zahtev_kursor
        fetch next from zahtev_kursor into
@rbZah,@imeZap,@prezZap,@idZap,@zahtev,@prioritet,@datumP
        while @@FETCH_STATUS=0
        begin

            print 'Zaposlenom '+@imeZap+' '+@prezZap+' sa id-jem
'+convert(varchar,@idZap)+
            ' odobren je zahtev pod rednim brojem '+convert(varchar,@rbZah)+' '
+@zahtev+' prioriteta '+
            @prioritet+' sa pocetkom od: '+convert(varchar,@datumP)

            fetch next from zahtev_kursor into
@rbZah,@imeZap,@prezZap,@idZap,@zahtev,@prioritet,@datumP

            end
            close zahtev_kursor
            deallocate zahtev_kursor
            print 'Hr menadzer sa id-jem '+convert(varchar,@idMen) +' je odobrio ukupno:
'+convert(varchar,@brojZ)+' zahteva'
            end
        else
        begin
            print 'Hr menadzer sa id-jem '+convert(varchar,@idMen) +' jos uvek nije
odobrio ni jedan zahtev.'
        end
    end
end
go

```

Funkcija PrioritetFunkcija

```

if object_id('Nagrade_sankcije_obuka.PrioritetFunkcija')is not null
drop function Nagrade_sankcije_obuka.PrioritetFunkcija
go
create function Nagrade_sankcije_obuka.PrioritetFunkcija
(@rbzahtev as numeric(5))
returns varchar(20)
as
begin

declare @pr as varchar(20)
set @pr=(select
case
when prioritet=1 then 'Urgentno'
when prioritet between 2 and 4 then 'Hitno'
when prioritet between 5 and 7 then 'Redovno'
when prioritet between 8 and 10 then 'Nizak prioritet'
when prioritet is null then 'Nepoznato'
end as Prioritet
from Nagrade_sankcije_obuka.Zahtev_za_odsustvo
where rb_zahteva_ods=@rbzahtev)
return @pr
end go

```

Hr menadzer sa id-jem 12 je odobrio sledece zahteve za odsustvo:
Zaposlenom Marija Petrovic sa id-jem 2 odobren je zahtev pod rednim brojem 1 Zahtev za porodiljskim odsustvom prioriteta Hitno sa pocetkom od: 2019-06-01
Zaposlenom Marija Petrovic sa id-jem 2 odobren je zahtev pod rednim brojem 3 Zahtev za odsustvo zbog sklapanja braka prioriteta Hitno sa pocetkom od: 2018-04-08
Zaposlenom Marija Petrovic sa id-jem 2 odobren je zahtev pod rednim brojem 5 Zahtev za odsustvo sa rada radi nege deteta prioriteta Urgentno sa pocetkom od: 2021-02-10
Zaposlenom Lazar Filipovic sa id-jem 13 odobren je zahtev pod rednim brojem 13 Zahtev za godisnji odmor prioriteta Nepoznato sa pocetkom od: 2020-11-29
Zaposlenom Lenka Lolic sa id-jem 15 odobren je zahtev pod rednim brojem 7 Zahtev za odsustvo sa rada radi nege deteta prioriteta Hitno sa pocetkom od: 2020-11-29
Zaposlenom Miroslav Mirkovic sa id-jem 22 odobren je zahtev pod rednim brojem 15 Zahtev za bolovanje prioriteta Redovno sa pocetkom od: 2020-05-18
Hr menadzer sa id-jem 12 je odobrio ukupno: 6 zahteva

Completion time: 2022-06-03T20:00:31.8068306+02:00

Slika 33 Pozitivan rezultat druge procedure

! Messages

Hr menadzer sa id-jem 13 jos uvek nije odobrio ni jedan zahtev.

Completion time: 2022-06-03T20:01:27.7831240+02:00

Slika 34 Negativan rezultat druge procedure

Hr menadzer sa id-jem 125 ne postoji u bazi.

Completion time: 2022-06-03T20:02:21.4280903+02:00

Slika 35 Negativan rezultat druge procedure

13. TRIGERI

Prvi trigger- PlataTriger

Aktivira se pri pokušaju unosa u tabelu Obavlja.

Okidanje ovog trigeru bi trebalo da rezultuje izmenom u tabelu Zaposleni.

Glavna svrha ovog trigeru je da se zaposlenima, koji su učestvovali u nekoj obuci i postigli rezultate veće od 90, dodeli povećanje na platu koja za zaposlene koji su projektni menadžeri iznosi 7000 NJ (novčanih jedinica), za hr menadžere iznosi 5000NJ, a za obične radnike 3000NJ. Tako se dalje u kodu deklarise kursor koji će proći kroz listu zaposlenih i uz određene uslove dodeliti svakom zaposlenom koji je učestvovao pomenutu sumu novca. Ispisuje se ime i prezime zaposlenog, kao i zarada pre i posle uplate novca. Ukoliko postignuti rezultati nisu veći od 90 podize se greška.

Za potrebe izvršenja ovog trigeru kreiran je kursor curosor_plata za potreben preuzimanja plata zaposlenih koji su na nekoj obuci osvojili više od 90 bodova.

```
if object_id('Nagrade_sankcije_obuka.PlataTriger') is not null
drop trigger Nagrade_sankcije_obuka.PlataTriger
go
create trigger Nagrade_sankcije_obuka.PlataTriger
on Nagrade_sankcije_obuka.Obavlja
instead of insert
as
begin
if @@ROWCOUNT=0
return;
declare @postignutiRez as numeric(3)=(select postignuti_rez from inserted)
declare @idZap as numeric=(select id_zaposlenog from inserted)
declare @idTip0 as numeric(5)=(select id_tip_obuke from inserted)
declare @rbObuke as numeric(5)=(select rb_obuke from inserted)
declare @imeZ as varchar(20)
declare @prezZ as varchar(20)
declare @plata as numeric(30)
if @postignutiRez > =90
begin
    declare curosor_plata cursor for
    select zap.id_zaposlenog,z_ime,z_prezime,z_plt,postignuti_rez
    from Nagrade_sankcije_obuka.Zaposleni zap join Nagrade_sankcije_obuka.Obavlja
    obv on zap.id_zaposlenog=obv.id_zaposlenog
    where rb_obuke=1 and zap.id_zaposlenog=@idZap

    open curosor_plata

    fetch next from curosor_plata into @idZap,@imeZ,@prezZ,@plata,@postignutiRez

    while @@FETCH_STATUS=0
    begin
        if (@idZap=(select projektni_menadzer from
        Nagrade_sankcije_obuka.Projektni_menadzer where projektni_menadzer=@idZap))
        begin
            update Nagrade_sankcije_obuka.Zaposleni
            set z_plt=z_plt+7000
            where id_zaposlenog=@idZap
```

```

end
else if (@idZap=(select hr_menadzer from
Nagrada_sankcije_obuka.Hr_menadzer where hr_menadzer=@idZap))
begin
update Nagrade_sankcije_obuka.Zaposleni
set z_plt=z_plt+5000
where id_zaposlenog=@idZap

insert
Nagrada_sankcije_obuka.Obavlja(postignuti_rez,id_zaposlenog,id_tip_obuke,rb_obuke)
values(@postignutiRez,@idZap,@idTipO,@rbObuke);
end
else if (@idZap !=(select radnik from
Nagrada_sankcije_obuka.Radnik where radnik=@idZap))
begin
update Nagrade_sankcije_obuka.Zaposleni
set z_plt=z_plt+3000
where id_zaposlenog=@idZap
insert
Nagrada_sankcije_obuka.Obavlja(postignuti_rez,id_zaposlenog,id_tip_obuke,rb_obuke)
values(@postignutiRez, @idZap,@idTipO,@rbObuke);
end

declare @novaPltr as numeric(10) =(select z_plt from
Nagrada_sankcije_obuka.Zaposleni
where id_zaposlenog=@idZap)
print (@imeZ+' '+@prezZ+', zarada pre izmene:'+try_cast(@plata as varchar)
+', zarada posle izmene:'+try_cast(@novaPltr as varchar))

fetch next from curosor_plata into
@idZap,@imeZ,@prezZ,@plata,@postignutiRez

end
close curosor_plata
deallocate curosor_plata
end
else
print 'Postignuti rezultati su manji od 90, nema promene plate.'
insert
Nagrada_sankcije_obuka.Obavlja(postignuti_rez,id_zaposlenog,id_tip_obuke,rb_obuke)
values(@postignutiRez,@idZap,@idTipO,@rbObuke);
end

(1 row affected)
Cveta Adamovic, zarada pre izmene:285000, zarada posle izmene:292000

(1 row affected)

(1 row affected)

Completion time: 2022-06-03T20:49:36.5225995+02:00

go

```

Slika 36 Rezultat kada su uslovi trigera uspesno zadovoljeni

Messages

```

Postignuti rezultati su manji od 90, nema promene plate.

(1 row affected)

(1 row affected)

Completion time: 2022-06-03T20:48:31.0238095+02:00

```

Slika 37 Rezultat kada su uslovi trigera nesuesno zadovoljeni

Drugi trigger- ZahtevTriger

Triger se pokreće prilikom unosa u teblu Zahtev za odsustvo i treba da signalizira grešku (i time sprečiti insert) ukoliko je ukupan broj evidentiranih zahteva za odsustvo zaposlenog za kojeg se pokušava uneti novi zahtev veći od 5.

```
if object_id('Nagrade_sankcije_obuka.ZahtevTriger') is not null
drop trigger Nagrade_sankcije_obuka.ZahtevTriger
go
create trigger Nagrade_sankcije_obuka.ZahtevTriger
on Nagrade_sankcije_obuka.Zahtev_za_odsustvo
instead of insert
as
begin
if @@ROWCOUNT=0
return;
declare @idZap as numeric(5)=(select id_zaposlenog from inserted)
declare @datumP as date=(select datum_poc_ods from inserted)
declare @datumZ as date=(select datum_zav_ods from inserted)
declare @prioritet as numeric(2)=(select prioritet from inserted)
declare @status as numeric(5)=(select id_status from inserted)
declare @tipZ as numeric(5)=(select id_tip_zahteva_ods from inserted)
declare @hrM as numeric(5)=(select hr_menadzer from inserted)

declare @brZahteva as numeric(5)=(select count(rb_zahteva_ods) from
Nagrade_sankcije_obuka.Zahtev_za_odsustvo
where id_zaposlenog=@idZap)
if @brZahteva >5
begin
declare @msg as varchar(100)
set @msg=formatmessage('Zaposleni sa id-jem '+convert(varchar,@idZap)
+' ima vise od 5 zahteva za odsustvo, ne moze mu se dodeliti novo.')
raiserror(@msg,16,0)
end
else
begin
print 'Zaposleni sa id-jem '+convert(varchar,@idZap)+' ima manje od 5 zahteva za
odsustvo moze mu se uneti novi.'
insert
Nagrade_sankcije_obuka.Zahtev_za_odsustvo(datum_poc_ods,datum_zav_ods,prioritet,id_zap
oslenog,id_status,id_tip_zahteva_ods,hr_menadzer)
values(@datumP,@datumZ,@prioritet,@idZap,@status,@tipZ,@hrM)
end
end
go
```

Zaposleni 1 ima manje od 5 zahteva za odsustvo moze mu se uneti novi.

(1 row affected)

(1 row affected)

Completion time: 2022-06-03T21:52:21.1322515+02:00

Slika 38 Rezultat drugog triger

Messages

Msg 50000, Level 16, State 0, Procedure ZahtevTriger, Line 24 [Batch Start Line 151]
Zaposleni sa id-jem 1 ima vise od 5 zahteva za odsustvo, ne moze mu se dodeliti novo.

(1 row affected)

Completion time: 2022-06-03T21:54:38.2682391+02:00

Slika 39 Rezultat drugog triger

ZAKLJUČAK

Za relazaciju projekta iz predmeta Sistemi baza podataka korišćena je šema projekta sa predmeta Projektovanje baza podataka. Krajnjim rezultat modelovanja šeme na tom projektu, mislili smo da smo dobili zadovoljavajući model na kojem ćemo kasnije moći dalje razvijamo sistem.

Međutim, radivši na ovom projektu, jasno se može zaključiti da tadašnja znanja nisu bila dovoljna da se u kompletu razume sistem kako bi se efikasno izmodelovali svi delovi. Radivši na jednoj od podšema, bilo je prilike da se više posveti konkretnim delovima i tipovima entiteta/poveznika, te je neretko bilo potrebno izmeniti određene sitnice, koje bi kasnije svakako pravile problem pri konzistentnosti baze podataka. Takođe, poslovna logika i elementi koje nije moguće implementirati bez prethodno smišljene logike, bili su još jedan izazov koji je rezultirao uspešnoj realizaciji zadataka, kao i kvalitetnim idejama koje će svima kasnije koristiti.