Fakulta riadenia a informatiky

Informačné a riadiace systémy

Zadanie 1 – Špecifikácia D

*Jednoduchý IS pre voľby*

Valentíková Eva – PIA 06,07 Jakub Vráb 5ZI024

2012 Martin Ďurčanský 5ZI024

**Zadanie č.3 (**termín odovzdania **12-13.týždeň** semestra **- 20 bodov, oponovanie – 5 bodov)**

**(team max 4.študentov)**

Implementujte informačný systém v prostredí ORACLE, pomocou nástroja SQL\*Plus.

 **procedúry** pre vstup, modifikáciu a rušenie údajov

 **parametrizované zostavy** podľa špecifikácie zadania č.1

 vytvorte **triggre**, ktoré budú monitorovať **modifikáciu** aspoň jednej tabuľky

 každý zo študentov musí spraviť aspoň jednu procedúru, jeden trigger (integritné obmedzenie, logovanie, prideľovanie ID), jeden report – Táto požiadavka bude striktne kontrolovaná.

**Pre získanie 5 bodov za oponovanie je nutné v dokumentácii doložiť oponentský posudok.**

**Zviazaná dokumentácia v papierovej forme musí obsahovať:**

 **Úvodná strana** –fakulta, program, zadanie a špecifikácia, zloženie tímu, skupina, navštevované cvičenia (vyučujúci, deň, bloky, šk. rok)

 **Zadanie** + 5 požadovaných výstupov od konzultanta

 **Dátový diagram**(podľa zadania 1)

 **Programátorská príručka**

o popis vytvorených DB objektov – typ, názov, parametre, výstup, autor, zoznam použitých zdrojov (tabuľky...), stručný popis riešenia

o popis reportov – umiestnenie , názov súboru, parametre, autor, zoznam použitých zdrojov, stručný popis riešenia

o príprava databázy a generovanie dát – autori, problémy, použité techniky

 **Užívateľská príručka**

o prehľad dát, ktoré je možné použiť na testovanie

o priradenie jednotlivých bodov zadania a príkazov na spustenie príslušného reportu, alebo procedúry

 **Záver** – zhrnutie práce, hodnotenie prínosu jednotlivých členov tímu, popis problémov, s ktorými ste sa stretli a ich riešenia

 **Oponentský posudok**

**Výsledok zadania 3 v elektronickej forme:**

 **Názov –** specifikacia\_vyucujuci\_DENblok.zip (príklad mandarino\_V\_UTO0708.zip)

 **Obsah:**

o dokumentácia zadania 2 **(\*.doc, \*.pdf)**

o adresár so zdrojovými kódmi reportov

o adresár so zdrojovými kódmi funkcií, procedúr a triggrov

o schéma databázy – výsledok zadania1

o export databázy bez DB objektov používaných na výuku

Pre prezentáciu semestrálnej práce si študenti donesú navyše 1 ks **vytlačenej** dokumentácie.

**Špecifikácia D:** *Jednoduchý IS pre voľby*

V jednej malebnej krajine, kde sa piesok lial a voda sa sypala (a teda neplatili tam ani iné „fyzikálne zákony“) sa občania zrazu rozhodli pre parlamentne voľby. Aby bola zabezpečená ich transparentnosť, čestnosť a objektívnosť, rozhodli sa, že vytvoria informačný systém, ktorý umožní každému nádejnému voličovi dozvedieť sa potrebné informácie, ktoré mu umožnia čo najlepšie sa rozhodnúť pre danú stranu alebo kandidáta.

**Zrealizujte priebeh volieb. Budete potrebovať:**

 Keďže voľby sú tajné, rátajte s tým, že volič:

o Nemusí prísť.

o Bude hlasovať platne.

o Bude hlasovať neplatne.

 Občan môže hlasovať iba raz.

 Nie je možné zistiť kto dal akej strane hlas, je možné vedieť iba či občan volil alebo nie.

 ***BONUS***(***pre 3.semestralnu prácu****)* ***– simulácia volieb*:** automatizujte priebeh volieb a použite niečo na štýl: **dbms\_random.value** alebo podobné ...

**Pri návrhu tohto systému mali občania nasledovné požiadavky:**

 evidencia subjektov (strán) ktoré sú k dispozícii

 evidencia osôb kandidujúcich za jednotlivé strany, ich profil (CV)

 evidencia sponzorov (firiem) ktorí financujú jednotlivé strany, + výška podpory

 evidencia mítingov volebnej kampane jednotlivých strán (kedy, kde...)

 evidencia voličov (občanov)

 priebeh hlasovania (volieb)

**Z informačného systému je potrebné získavať nasledujúce zostavy:**

(Po skončení volieb, resp. počas ich priebehu musíte vedieť poskytnúť štatistiku)

 Podrobný prehľad: Volebná účasť resp. neúčasť, počet platných a neplatných hlasov.

 Percentuálne: počet platných hlasov, počet neplatných, jednotlivé počty percent pre politické strany. (percentá pre strany sa rátajú iba z platných hlasov)

 Spravte report, ktorý na základe výsledku volieb poskytne prepočet miest v parlamente. Predpokladajme že parlament ma 150 miest a miesta v ňom sa delia pomerovo vzhľadom na dosiahnuté výsledky (podmienka účasti v parlamente je min 5% získaných hlasov vo voľbách).

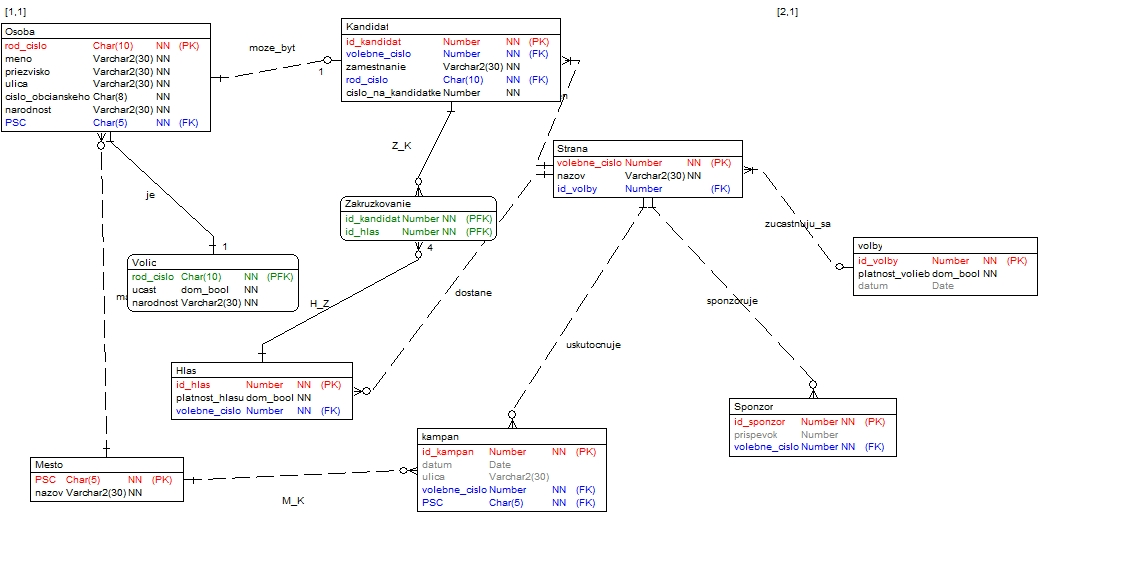
 Každá strana kt. dosiahne viac ako 3% (platných hlasov) ma nárok od štátu za každý získaný hlas 7 eur. Vytvorte report, ktorý zobrazí zoznam strán s ich percentom a výškou príspevku (ak ho získali).

 Podrobný prehľad financovania strán.

**Výpis požadovaný od konzultanta**

1. Výpis strany ktoré mali menej ako 5 %
2. Výpis stranu ktorá mala najviac hlasov
3. Výpis stranu ktorá mala najmenej hlasov
4. Výpis kandidáta ktorý mal najviac hlasov
5. Výpis kandidáta ktorý mal najmenej hlasov

**Dátový diagram**



# **Programátorská príručka**

**Procedúry**

***-- vloží nové mesto***

create or replace procedure vloz\_mesto

(

PSC in varchar2,

nazov in varchar2)

is

begin

if PSC is null or nazov is null then

raise\_application\_error(-20000,'Vsetky polozky musia byt zadane!');

else begin

insert into mesto(PSC,nazov) values (PSC,nazov);

end;

end if;

end vloz\_mesto;

/

***-- upraví mesto***

CREATE OR REPLACE PROCEDURE EDIT\_MESTO

(

epsc in varchar2,

enazov in varchar2)

as

begin

update mesto set

nazov = enazov where psc=epsc;

end;

/

***-- vymaže mesto***

create or replace procedure zmaz\_mesto

(epsc in varchar2,

enazov in varchar2)

as

begin

delete from mesto where psc = epsc and nazov = enazov;

end;

/

***--vloží novu osobu***

create or replace procedure vloz\_osoba

(

rod\_cislo in char,

meno in varchar2,

priezvisko in varchar2,

ulica in varchar2,

cislo\_obcianskeho in char,

narodnost in varchar2,

PSC in varchar2)

is

begin

if rod\_cislo is null or meno is null or priezvisko is null

or ulica is null or cislo\_obcianskeho is null or narodnost is null or PSC is null

then

raise\_application\_error(-20000,'Vsetky polozky musia byt zadane!');

else begin

insert into osoba(rod\_cislo,meno,priezvisko,ulica,cislo\_obcianskeho,narodnost,PSC)

values(rod\_cislo,meno,priezvisko,ulica,cislo\_obcianskeho,narodnost,PSC);

end;

end if;

end vloz\_osoba;

/

***--upraví osobu***

CREATE OR REPLACE PROCEDURE EDIT\_osoba

(

erod\_cislo in char,

emeno in varchar2,

epriezvisko in varchar2,

eulica in varchar2,

ecislo\_obcianskeho in char,

enarodnost in varchar2,

ePSC in varchar2)

as

begin

update osoba set

meno = emeno,

priezvisko = epriezvisko,

ulica = eulica,

cislo\_obcianskeho = ecislo\_obcianskeho,

narodnost = enarodnost,

psc = epsc

where erod\_cislo = rod\_cislo;

end;

/

***--vymaže osobu***

create or replace procedure zmaz\_osoba

(erod\_cislo in char)

as

begin

delete from osoba where erod\_cislo = rod\_cislo;

end;

/

***--vloží nový hlas***

create or replace procedure vloz\_hlas

(

platnost\_hlasu in hlas.platnost\_hlasu%type,

volebne\_cislo in number)

is

begin

insert into hlas(platnost\_hlasu,volebne\_cislo) values (platnost\_hlasu,volebne\_cislo);

end vloz\_hlas;

/

**--upraví hlas**

CREATE OR REPLACE PROCEDURE EDIT\_hlas

(

eid\_hlas in number,

eplatnost\_hlasu in hlas.platnost\_hlasu%type,

evolebne\_cislo in number)

as

begin

update hlas set

platnost\_hlasu = eplatnost\_hlasu,

volebne\_cislo = evolebne\_cislo

where id\_hlas=eid\_hlas;

end;

/

***--vymaže hlas***

create or replace procedure zmaz\_hlas

(eid\_hlas in number)

as

begin

delete from hlas where id\_hlas=eid\_hlas;

end;

/

***--vloží nového voliča***

create or replace procedure vloz\_volic

(

rod\_cislo in volic.ucast%type,

ucast in volic.ucast%type,

narodnost in volic.ucast%type)

is

begin

if rod\_cislo is null or ucast is null or narodnost is null then

raise\_application\_error(-20000,'Vsetky polozky musia byt zadane!');

else begin

insert into Volic(rod\_cislo,ucast,narodnost) values (rod\_cislo,ucast,narodnost);

end;

end if;

end vloz\_volic;

/

***--upraví voliča***

CREATE OR REPLACE PROCEDURE EDIT\_volic

(

erod\_cislo in volic.ucast%type,

eucast in volic.ucast%type,

enarodnost in volic.ucast%type)

as

begin

update volic set

ucast = eucast,

narodnost = enarodnost

where rod\_cislo=erod\_cislo;

end;

/

***--vymaže voliča***

create or replace procedure zmaz\_volic

(erod\_cislo in volic.ucast%type)

as

begin

delete from volic where rod\_cislo=erod\_cislo;

end;

/

***--vloží nového kandidáta***

create or replace procedure vloz\_kandidat

(

volebne\_cislo in kandidat.volebne\_cislo%type,

zamestnanie in kandidat.zamestnanie%type,

rod\_cislo in kandidat.rod\_cislo%type,

cislo\_na\_kandidatke in kandidat.cislo\_na\_kandidatke%type)

is

begin

insert into kandidat(volebne\_cislo,zamestnanie,rod\_cislo,cislo\_na\_kandidatke)

values(volebne\_cislo,zamestnanie,rod\_cislo,cislo\_na\_kandidatke);

end vloz\_kandidat;

/

***--vloží novú stranu***

create or replace procedure vloz\_strana

(

volebne\_cislo in number,

nazov in varchar2,

id\_volby in number)

is

begin

insert into strana(volebne\_cislo,nazov,id\_volby) values (volebne\_cislo,nazov,id\_volby);

end vloz\_strana;

/

***--vloží nového sponzora***

create or replace procedure vloz\_sponzor

(

prispevok in number,

volebne\_cislo in number)

is

begin

insert into sponzor(prispevok,volebne\_cislo) values (prispevok,volebne\_cislo);

end vloz\_sponzor;

/

***--vloží nové voľby***

create or replace procedure vloz\_volby

(

platnost\_volieb in volby.platnost\_volieb%type,

datum in date)

is

begin

insert into volby(platnost\_volieb,datum) values (platnost\_volieb,datum);

end vloz\_volby;

/

***--vloží novú kampaň***

create or replace procedure vloz\_kampan

(

datum in date,

ulica in varchar2,

volebne\_cislo in number,

psc in varchar2)

is

begin

insert into kampan(datum,ulica,volebne\_cislo,psc)

values (datum,ulica,volebne\_cislo,psc);

end vloz\_kampan;

/

**Funkcie**

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--vypocita pocet percent hlasov dosiahnutych vo volbach

create or replace function percenta (platnost\_hlasu in hlas.platnost\_hlasu%type,volebne\_cislo in hlas.volebne\_cislo%type)

return integer

is

celkovy\_pocet integer;

pocet\_strany integer;

begin

select count(\*) into celkovy\_pocet from hlas where platnost\_hlasu=percenta.platnost\_hlasu;

select count(\*) into pocet\_strany from hlas where platnost\_hlasu=percenta.platnost\_hlasu and volebne\_cislo=percenta.volebne\_cislo;

return trunc(pocet\_strany/celkovy\_pocet\*100);

end percenta;

/

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--vypocita pocet percent kresiel dosiahnutych vo volbach

create or replace function kresla (volebne\_cislo in hlas.volebne\_cislo%type)

return integer

is

begin

if percenta(1,volebne\_cislo)>4 then

return percenta(1,volebne\_cislo)\*1.5;

else return 0;

end if;

end kresla;

/

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--vypocita pocet hlasov kandidatov

create or replace function pocethlasovk (id\_kandidat in zakruzkovanie.id\_kandidat%type)

return integer

is

pocet integer;

begin

select count(\*) into pocet from zakruzkovanie where id\_kandidat=pocethlasovk.id\_kandidat;

return pocet;

end pocethlasovk;

/

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--vypocita pocet hlasov strany

create or replace function pocethlasov (platnost\_hlasu in hlas.platnost\_hlasu%type,volebne\_cislo in hlas.volebne\_cislo%type)

return integer

is

pocet integer;

begin

select count(\*) into pocet from hlas where platnost\_hlasu=pocethlasov.platnost\_hlasu and volebne\_cislo=pocethlasov.volebne\_cislo;

return pocet;

end pocethlasov;

/

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--vypocita mnozstvo penazi ziskanych po volbach

create or replace function peniaze (volebne\_cislo in hlas.volebne\_cislo%type)

return integer

is

begin

if percenta(1,volebne\_cislo)>3 then

return pocethlasov (1,volebne\_cislo)\*7;

else return 0;

end if;

end peniaze;

/

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--najde strany ktore sa nedostali do volieb

create or replace function nedostali (volebne\_cislo in hlas.volebne\_cislo%type)

return integer

is

begin

if percenta(1,volebne\_cislo)<5 then

return volebne\_cislo;

else

return 0;

end if;

end nedostali;

/

**Triggre**

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--trigger vlozi do log\_tab operaciu robenu nad mestom,meno a datum

create table log\_tab(

user\_name varchar2(20),

datum date,

operacia char(1),

PSC char(5),

nazov varchar2(30));

create or replace trigger tr\_log\_mesto\_I

before insert on mesto

for each row

begin

insert into log\_tab(user\_name,datum,operacia,psc,nazov)

SELECT user,sysdate, 'I', :new.psc,:new.nazov

FROM dual;

end;

/

alter trigger tr\_log\_mesto\_I enable;

create or replace trigger tr\_log\_mesto\_U

before update on mesto

for each row

begin

insert into log\_tab(user\_name,datum,operacia,psc,nazov)

SELECT user,sysdate, 'U', :new.psc,:new.nazov

FROM dual;

end;

/

alter trigger tr\_log\_mesto\_U enable;

create or replace trigger tr\_log\_mesto\_D

before delete on mesto

for each row

begin

insert into log\_tab(user\_name,datum,operacia,psc,nazov)

SELECT user,sysdate, 'D', :old.psc,:old.nazov

FROM dual;

end;

/

alter trigger tr\_log\_mesto\_D enable;

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--trigger vlozi do log\_tab2 operaciu robenu nad hlasom,meno a datum

create table log\_tab2(

user\_name varchar2(20),

datum date,

operacia char(1),

platnost\_hlasu char(5),

volebne\_cislo varchar2(30));

create or replace trigger tr\_log\_hlas\_I

before insert on hlas

for each row

begin

insert into log\_tab2(user\_name,datum,operacia,platnost\_hlasu,volebne\_cislo)

SELECT user,sysdate, 'I', :new.platnost\_hlasu,:new.volebne\_cislo

FROM dual;

end;

/

alter trigger tr\_log\_hlas\_I enable;

create or replace trigger tr\_log\_hlas\_U

before insert on hlas

for each row

begin

insert into log\_tab2(user\_name,datum,operacia,platnost\_hlasu,volebne\_cislo)

SELECT user,sysdate, 'U', :new.platnost\_hlasu,:new.volebne\_cislo

FROM dual;

end;

/

alter trigger tr\_log\_hlas\_U enable;

create or replace trigger tr\_log\_hlas\_D

before insert on hlas

for each row

begin

insert into log\_tab2(user\_name,datum,operacia,platnost\_hlasu,volebne\_cislo)

SELECT user,sysdate, 'D', :old.platnost\_hlasu,:old.volebne\_cislo

FROM dual;

end;

/

alter trigger tr\_log\_hlas\_D enable;

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--triggre automaticky pridavaju IDecka

CREATE SEQUENCE seq\_kandidat START WITH 3 INCREMENT BY 1;

CREATE OR REPLACE TRIGGER ins\_kandidat

BEFORE INSERT ON kandidat

FOR EACH ROW

BEGIN

select seq\_kandidat.nextval into :new.id\_kandidat from dual;

END;

/

CREATE SEQUENCE seq\_sponzor START WITH 6 INCREMENT BY 1;

CREATE OR REPLACE TRIGGER ins\_sponzor

BEFORE INSERT ON sponzor

FOR EACH ROW

BEGIN

select seq\_sponzor.nextval into :new.id\_sponzor from dual;

END;

/

CREATE SEQUENCE seq\_hlas START WITH 21 INCREMENT BY 1;

CREATE OR REPLACE TRIGGER ins\_hlas

BEFORE INSERT ON hlas

FOR EACH ROW

BEGIN

select seq\_hlas.nextval into :new.id\_hlas from dual;

END;

/

CREATE SEQUENCE seq\_volby START WITH 3 INCREMENT BY 1;

CREATE OR REPLACE TRIGGER ins\_volby

BEFORE INSERT ON volby

FOR EACH ROW

BEGIN

select seq\_volby.nextval into :new.id\_volby from dual;

END;

CREATE SEQUENCE seq\_kampan START WITH 4 INCREMENT BY 1;

CREATE OR REPLACE TRIGGER ins\_kampan

BEFORE INSERT ON kampan

FOR EACH ROW

BEGIN

select seq\_kampan.nextval into :new.id\_kampan from dual;

END;

/

**Reporty**

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--vypise prehlad volieb, strana a pocet percent

SET PAGESIZE 40

SET LINESIZE 70

COLUMN nazov HEADING "Strany"

COLUMN percenta(1,hlas.volebne\_cislo) HEADING "pocet percent"

TTITLE CENTER "Prehlad volieb"

CREATE OR REPLACE VIEW prehlad\_volieb

AS

SELECT distinct strana.nazov as nazov\_strany,percenta(1,hlas.volebne\_cislo) as pocet\_percent,kresla(hlas.volebne\_cislo) as pocet\_kresiel

FROM strana join hlas on strana.volebne\_cislo = hlas.volebne\_cislo ;

SELECT \* FROM prehlad\_volieb;

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--vypise stranu s najvacsim poctom hlasov, aj pocet hlasov

SET PAGESIZE 40

SET LINESIZE 60

COLUMN nazov HEADING "Nazov"

TTITLE CENTER "maximum hlasov"

CREATE OR REPLACE VIEW najviac\_hlasov

AS

select nazov,max(pocethlasov(1,hlas.volebne\_cislo))as pocet

from hlas join strana on strana.volebne\_cislo = hlas.volebne\_cislo

where pocethlasov (1,hlas.volebne\_cislo)>=all(select pocethlasov (1,hlas.volebne\_cislo) from hlas) group by nazov;

SELECT \* FROM najviac\_hlasov;

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--vypise stranu s najmensim poctom hlasov, aj pocet hlasov

SET PAGESIZE 40

SET LINESIZE 60

COLUMN nazov HEADING "Nazov"

TTITLE CENTER "minimum hlasov"

CREATE OR REPLACE VIEW najmenej\_hlasov

AS

select nazov,min(pocethlasov(1,hlas.volebne\_cislo))as pocet

from hlas join strana on strana.volebne\_cislo = hlas.volebne\_cislo

where pocethlasov (1,hlas.volebne\_cislo)<=all(select pocethlasov (1,hlas.volebne\_cislo) from hlas) group by nazov;

SELECT \* FROM najmenej\_hlasov;

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--vypise kandidata s najmensim poctom hlasov, aj pocet hlasov

SET PAGESIZE 50

SET LINESIZE 80

COLUMN meno HEADING "meno"

COLUMN priezvisko HEADING "priezvisko"

TTITLE CENTER "minimum hlasov kandidatovi"

CREATE OR REPLACE VIEW najm\_hlasov

AS

select meno,priezvisko,min(pocethlasovk(kandidat.id\_kandidat))as pocet

from kandidat join strana on strana.volebne\_cislo = kandidat.volebne\_cislo join osoba on kandidat.rod\_cislo = osoba.rod\_cislo

where pocethlasovk(kandidat.id\_kandidat)<=all(select pocethlasovk(kandidat.id\_kandidat) from kandidat) group by meno,priezvisko;

SELECT \* FROM najm\_hlasov;

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--vypise kandidata s najvacsim poctom hlasov, aj pocet hlasov

SET PAGESIZE 50

SET LINESIZE 80

COLUMN meno HEADING "meno"

COLUMN priezvisko HEADING "priezvisko"

TTITLE CENTER "maximum hlasov kandidatovi"

CREATE OR REPLACE VIEW najv\_hlasov

AS

select meno,priezvisko,max(pocethlasovk(kandidat.id\_kandidat))as pocet

from kandidat join strana on strana.volebne\_cislo = kandidat.volebne\_cislo join osoba on kandidat.rod\_cislo = osoba.rod\_cislo

where pocethlasovk(kandidat.id\_kandidat)>=all(select pocethlasovk(kandidat.id\_kandidat) from kandidat) group by meno,priezvisko;

SELECT \* FROM najv\_hlasov;

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--vypise stranu a mnozstvo eur zarobenych vo volbach

TTITLE CENTER "zarobene"

CREATE OR REPLACE VIEW prehl

AS

SELECT distinct strana.nazov as nazov\_strany,peniaze(hlas.volebne\_cislo) as pocet\_eur

FROM strana join hlas on strana.volebne\_cislo = hlas.volebne\_cislo ;

SELECT \* FROM prehl;

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--vypise strany ktore sa nedostali do volieb

TTITLE CENTER "nedostali sa"

CREATE OR REPLACE VIEW pr

AS

SELECT distinct strana.nazov as nazov\_strany, nedostali(hlas.volebne\_cislo) as nedostali\_sa

FROM strana join hlas on strana.volebne\_cislo = hlas.volebne\_cislo

where nedostali(hlas.volebne\_cislo)>0;

SELECT \* FROM pr;

**Prehľad dát**

Tabuľka mesto

LOAD DATA

INFILE 'mesto.unl'

INTO TABLE mesto

FIELDS TERMINATED BY '|'

(

PSC

,nazov

)

Tabuľka strana

LOAD DATA

INFILE 'strana.unl'

INTO TABLE strana

FIELDS TERMINATED BY '|'

(

volebne\_cislo

,nazov

,id\_volby

)

Tabuľka sponzor

LOAD DATA

INFILE 'sponzor.unl'

INTO TABLE sponzor

FIELDS TERMINATED BY '|'

(

id\_sponzor

,prispevok

,volebne\_cislo

)

Tabuľka osoba

LOAD DATA

INFILE 'osoba.unl'

INTO TABLE osoba

FIELDS TERMINATED BY '|'

(

rod\_cislo

,meno

,priezvisko

,ulica

,cislo\_obcianskeho

,narodnost

,PSC

)

Tabuľka volic

LOAD DATA

INFILE 'volic.unl'

INTO TABLE volic

FIELDS TERMINATED BY '|'

(

rod\_cislo

,ucast

,narodnost

)

Tabuľka kandidat

LOAD DATA

INFILE 'kandidat.unl'

INTO TABLE kandidat

FIELDS TERMINATED BY '|'

(

id\_kandidat

,volebne\_cislo

,zamestnanie

,rod\_cislo

,cislo\_na\_kandidatke

)

Tabuľka zakruzkovanie

LOAD DATA

INFILE 'zakruzkovanie.unl'

INTO TABLE zakruzkovanie

FIELDS TERMINATED BY '|'

(

id\_kandidat

,id\_hlas

)

Tabuľka kampan

LOAD DATA

INFILE 'kampan.unl'

INTO TABLE kampan

FIELDS TERMINATED BY '|'

(

id\_kampan

,datum DATE 'dd/mm/YYYY'

,ulica

,volebne\_cislo

,psc

)

Tabuľka hlas

LOAD DATA

INFILE 'hlas.unl'

INTO TABLE hlas

FIELDS TERMINATED BY '|'

(

id\_hlas

,platnost\_hlasu

,volebne\_cislo

)

Tabuľka volby

LOAD DATA

INFILE 'volby.unl'

INTO TABLE volby

FIELDS TERMINATED BY '|'

(

id\_volby

,platnost\_volieb

,datum DATE 'dd/mm/YYYY'

)

**Záver**

V tejto semestrálnej práci sme sa postupne naučili vytvárať základný databázový systém, vytvárali sme procedúry a funkcie na jeho ovládanie a v neposlednom rade sme sa naučili, ako sa takéto systémy vytvárajú, správajú a ako ich máme používať.

Našu prácu sme vyvárali počas 3 krokov a to najprv vytvorením ERA diagramu a vygenerovaním SQL skriptov. V 2 kroku sme v dvoch tímoch tento ERA diagram normalizovali do 3NF a BCNF formy a v poslednom kroku sme vytvárali procedúry, funkcie, triggre a reporty, s ktorých pomocou sa dali do našej databázy pridávať záznamy, alebo vypisovať hľadané výsledky.