

# ZADANIE Z PREDMETU DATABÁZOVÉ SYSTÉMY

NÁZOV DOMÉNY: CESTOVNÁ KANCELÁRIA

**MENO A PRIEZVISKO: SOFIIA SAMIGULINA**

**AK. ROK: 2024/2025**

## ENTITNO-RELAČNÝ DIAGRAM



## OPIS ENTITNÝCH MNOŽÍN

Databázový systém z domény cestovná kancelária a zahŕňa v sebe 5 tabuliek.

Tabuľka Zákazník má 10 atribútov: meno, priezvisko, telefónne číslo, email, dátum narodenia, dátum prvého nákupu, kde sa dozvedel o nas, číslo dokladu, koľko zaplatil za mesiac, poistenie. Primárny kľúč je rodné číslo.

Tabuľka Oddelenie má 7 atribútov: meno riaditeľa, priezvisko riaditeľa, telefónne číslo riaditeľa, email riaditeľa, ID pracoviska, koľko zamestnancov, názov oddelenia. Primárny kľúč je ID pracoviska.

Tabuľka Zamestnanec má 12 atribútov: meno, priezvisko, ID zamestnanca, telefónne číslo, email, pozícia, ID pracoviska, mesačný plat, kedy nastúpil do práce, má deti?, vzdelanie, koľko predal za mesiac. Primárny kľúč je ID zamestnanca. Cudzí kľúč je ID pracoviska a má referenciu v tabuľke oddelenie.

Tabuľka Túra má 10 atribútov: číslo objednávky, dátum odchoda, typ dopravy, kde je start, číslo dokladu zákazníka, ID zamestnanca, kam?, typ cestovania, typ ubytovania, koľko dní. Primárny kľúč je číslo objednávky. Cudzí kľúče sú rodné číslo zákazníka (tabuľka zákazník) a ID zamestnanca (tabuľka zamestnanec).

Tabuľka Platba má 8 atribútov: číslo platby, dátum platby, typ platby, sum (koľko bola platba), číslo karty alebo IBAN (pre vrátenie), číslo darčeka preukazu, bonusy a akcie použité, číslo objednávky. Primárny kľúč je číslo platby. Cudzí kľúč je číslo objednávky, ktorý má referenciu v tabuľke Túra.

## OPIS VZŤAHOV MEDZI ENTITNÝMI MNOŽINAMI

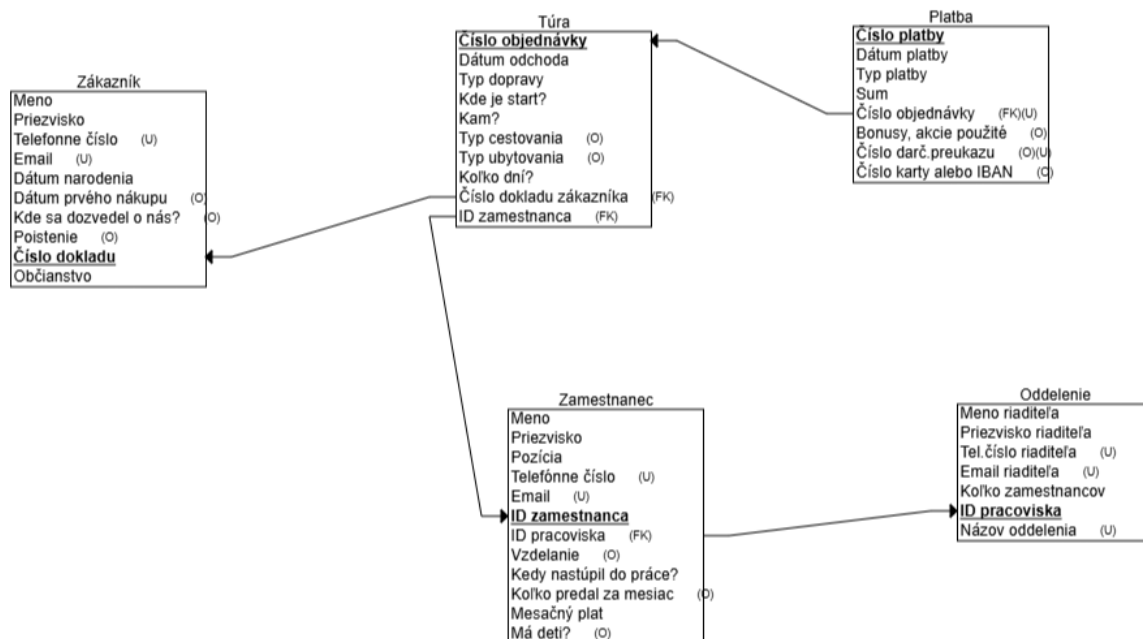
Tabuľky Zákazník a Túra majú vzťah 1:N (One-to-Many), lebo jeden Zákazník môže vykonať niekoľko nákupov, čiže kúpiť niekoľko túr (čo je predaj pre firmu).

Tabuľky Túra a Zamestnanec majú vzťah 1:N (One-to-Many), lebo jeden Zamestnanec môže predáť niekoľko túr.

Tabuľky Platba a Túra majú vzťah M:N (Many-to-Many), lebo s pomocou jednej platby alebo darčkového preukazu sa dá zaplatiť za viacero túr. Takisto jeden nákup túry môže mať viacero druhov platby: napríklad darčkový preukaz a platba kartou alebo platba kartou a nejaký použitý bonus či akcia.

Tabuľky Oddelenie a Zamestnanec majú vzťah 1:N (One-to-Many), lebo jeden zamestnanec pracuje v jednom oddelení, ale Oddelenie má viacero zamestnancov.

## RELAČNÝ MODEL



## PRVÁ ODOVZDÁVKA - SQL SKRIPT NA VYTVORENIE DATABÁZY

```
DROP TABLE IF EXISTS zakaznik CASCADE;
```

```
CREATE TABLE zakaznik (  
    meno varchar(50) not null,  
    priezvisko varchar(100) not null,  
    telefon int not null unique,  
    email varchar(50) not null unique,  
    datum_narodenia date not null,  
    datum_prveho_nakupu date,  
    poistenie varchar(30),  
    kde_sa_dozvedel_onas varchar(150),  
    cislo_dokladu varchar(20),  
    obcianstvo varchar(150) not null,  
    primary key (cislo_dokladu)  
);
```

```
DROP TABLE IF EXISTS oddelenie CASCADE;
```

```
CREATE TABLE oddelenie (  
    meno_riaditela varchar(50) not null,  
    priezvisko_riaditela varchar(100) not null,  
    telefon_riaditela int not null unique,  
    email_riaditela varchar(50) not null unique,  
    kolko_zamestnancov int not null,  
    id_pracoviska int,  
    nazov_oddelenia varchar(500) not null unique,  
    primary key (id_pracoviska)  
);
```

```
DROP TABLE IF EXISTS zamestnanec CASCADE;
```

```
CREATE TABLE zamestnanec (  
    meno varchar(50) not null,  
    priezvisko varchar(100) not null,  
    pozicia varchar(100) not null,  
    telefon int not null unique,  
    email varchar(50) not null unique,  
    ID_zamestnanca int,  
    ID_pracoviska int not null,  
    vzdelanie varchar(500),  
    kedy_nastupil_doprace date not null,  
    mesacny_plat float not null,  
    ma_deti char(5),  
    primary key (ID_zamestnanca),  
    foreign key (ID_pracoviska) references oddelenie(ID_pracoviska)  
);
```

```
DROP TABLE IF EXISTS tura CASCADE;
```

```
CREATE TABLE tura (  
    cislo_objednavky INT,  
    datum_odchoda date not null,  
    typ_dopravy varchar(100) not null,  
    kde_je_start varchar(100) not null,  
    kam varchar(100) not null,  
    typ_cestovania varchar(100),  
    typ_ubytovania varchar(100),  
    kolko_dni INT not null,  
    cislo_dokladu_zakaznika varchar(20) not null,  
    ID_zamestnanca int not null,  
    primary key (cislo_objednavky),  
    foreign key (cislo_dokladu_zakaznika) references zakaznik(cislo_dokladu),  
    foreign key (ID_zamestnanca) references zamestnanec(ID_zamestnanca)  
);
```

```
DROP TABLE IF EXISTS platba CASCADE;
```

```
CREATE TABLE platba (  
    cislo_platby int,  
    datum_platby date not null,  
    typ_platby varchar(50) not null,  
    sum float not null,  
    cislo_objednavky int not null unique,  
    bonusy_akcie varchar(100),  
    cislo_darcekového_preukazu int unique,  
    cislo_karty_IBAN varchar(100),  
    primary key (cislo_platby),  
    foreign key (cislo_objednavky) references tura(cislo_objednavky)  
);
```

## DRUHÁ ODOVZDÁVKA

### SKRIPT NA NAPLNENIE DATABÁZY (KAŽDÁ TABUĽKA ASPOŇ 5 ZÁZNAMOV)

INSERT INTO zakaznik

(meno, priezvisko, telefon, email, datum\_narodenia, datum\_prveho\_nakupu, poistenie,  
kde\_sa\_dozvedel\_onas, cislo\_dokladu, obcianstvo)

VALUES

('Alexandra', 'Polakova', 0987236451, 'sandra.polakova@yahoo.com', '1976-10-10', '2024-12-05', 'VZP',  
NULL, '761010/8517', 'Slovensko'),  
('Sonya', 'Migasova', 0959163544, 'sonya200397@gmail.com', '1997-03-20', '2024-12-10', NULL,  
'Instagram', '975320/8756', 'Rusko'),  
('Branislav', 'Kucera', 0968962123, 'kucera88.branislav@gmail.com', '1988-05-10', '2024-12-15', NULL,  
'kamarati', 'P23556911', 'Slovensko'),  
('Ivana', 'Horvathova', 0948659727, 'ivankaa1998@gmail.com', '1998-02-20', '2024-12-20', 'UNION',  
'kamarati', '985220/6056', 'Slovensko'),  
('Mirka', 'Novakova', 0998242289, 'novakova\_m@outlook.com', '1989-11-29', '2025-01-05', 'VZP',  
'Instagram', '896129/2439', 'Slovensko'),  
('Branislav', 'Tomcik', 0967891233, 'branislav10021978@outlook.com', '1978-02-10', '2025-01-07',  
NULL, 'Facebook', 'P45612978', 'Slovensko'),  
('Alina', 'Vasilenko', 0967995131, 'vasilenko\_a@gmail.com', '1994-05-15', '2025-01-10', NULL,  
'Facebook', '945515/8779', 'Ukrajina'),  
('Samuel', 'Novacikov', 0977555111, 'novacikov\_1968@yahoo.com', '1968-04-13', '2025-01-17',  
'UNION', NULL, '680413/3787', 'Slovensko'),  
('Juraj', 'Molnar', 0931998462, 'molnar\_juraj@outlook.com', '1974-12-03', '2025-01-23', 'Dovera',  
'Facebook', '741203/1506', 'Slovensko'),  
('Julia', 'Ivanova', 0936778123, 'ivanolia@gmail.com', '1988-01-31', '2025-02-01', NULL, 'Instagram',  
'885131/1743', 'Ukrajina');

INSERT INTO oddelenie

(meno\_riaditela, priezvisko\_riaditela, telefon\_riaditela, email\_riaditela, kolko\_zamestnancov,  
nazov\_oddelenia)

VALUES

('Samuel', 'Migas', 0956725463, 'migas\_it@outlook.com', 3, 'Technicke oddelenie'),  
('Antonia', 'Vargova', 0900555666, 'vargova\_predaj@outlook.com', 7, 'Predajne oddelenie'),  
('Greta', 'Blanarova', 0983421655, 'blanarova\_pr@outlook.com', 3, 'Reklamne oddelenie'),  
('Blazej', 'Vinogradov', 0924569781, 'vinogradov\_support@outlook.com', 2, 'Zakaznicke oddelenie'),  
('Irina', 'Tereshchuk', 0956698123, 'tereshchuk\_fin@outlook.com', 2, 'Financne oddelenie');

INSERT INTO zamestnanec

(meno, priezvisko, pozicia, telefon, email, ID\_zamestnanca, ID\_pracoviska, vzdelanie,  
kedy\_nastupil\_doprace, mesacny\_plat, ma\_deti)

VALUES

('Katka', 'Novakova', 'predajca zajazdov', 0904123890, 'katka.novakova@outlook.com', 1011, 101,  
NULL, '2024-11-25', 1350, 'nie'),  
('Pavol', 'Petrov', 'specialista mestskej reklamy', 0919456789, 'pavol.petrov@outlook.com', 1021, 102,  
NULL, '2024-11-25', 1350, 'ano'),  
('Mirka', 'Vargova', 'predajca zajazdov', 0905678901, 'mirka.vargova@outlook.com', 1012, 101, NULL,  
'2024-12-05', 1350, 'ano'),  
('Norbert', 'Varga', 'referent pre reklamacie', 0912456789, 'norbert.varga@outlook.com', 1015, 101,

NULL, '2024-12-15', 1050, NULL),  
('Diana', 'Kovalenko', 'specialista call centru', 0907345678, 'diana.kovalenko@outlook.com', 1031, 103, NULL, '2024-12-15', 1000, 'nie'),  
('Alexandra', 'Polarikova', 'referent pre reklamacie', 0915678901, 'alexandra.polarikova@outlook.com', 1016, 101, 'Obchodna sprava', '2024-12-17', 1050, 'nie'),  
('Samuel', 'Novakov', 'uctovnik', 0919654321, 'samuel.novakov@outlook.com', 1041, 104, NULL, '2024-12-18', 1050, 'ano'),  
('Norbert', 'Kovac', 'predajca zajazdov', 0910234567, 'norbert.kovac@outlook.com', 1013, 101, 'Obchodna sprava', '2025-01-03', 1450, 'ano'),  
('Robert', 'Polak', 'predajca zajazdov', 0911987654, 'robert.polak@outlook.com', 1014, 101, 'Obchodna sprava', '2025-01-03', 1400, 'nie'),  
('Samuel', 'Bechera', 'frontend developer', 0902987654, 'samuel.bechera@outlook.com', 1001, 100, 'Informatika', '2025-01-10', 2500, NULL),  
('Michal', 'Vadimov', 'online supporter', 0948234567, 'michal.vadimov@outlook.com', 1032, 103, 'Veterinaria', '2025-01-16', 1200, NULL),  
('Richard', 'Polarik', 'specialista telefonnej reklamy', 0940678901, 'richard.polarik@outlook.com', 1022, 102, NULL, '2025-01-17', 1350, 'nie'),  
('Dusan', 'Ambros', 'QR inzinier', 0903456789, 'dusan.ambros@outlook.com', 1002, 100, NULL, '2024-01-19', 1850, 'ano'),  
('Anya', 'Kirsanova', 'social media manazer', 0918987654, 'anya.kirsanova@outlook.com', 1023, 101, 'Reklama a marketing', '2025-01-21', 1300, NULL),  
('Kristina', 'Odincova', 'analytik predaja', 0917234567, 'kristina.odincova@outlook.com', 1017, 101, 'Analytik dat', '2025-01-25', 1500, 'ano'),  
('Ivana', 'Kovacova', 'uctovnik', 0902789012, 'ivana.kovacova@outlook.com', 1042, 104, 'Financna sprava', '2025-02-05', 1050, NULL),  
('Sofiiia', 'Samigulina', 'databazovy spravca', 0901234567, 'sofiia.samigulina@outlook.com', 1003, 100, 'Informatika', '2025-02-05', 1600, NULL);

INSERT INTO tura

(datum\_odchoda, typ\_dopravy, kde\_je\_start, kam, typ\_cestovania, typ\_ubytovania, kolko\_dni, cislo\_dokladu\_zakaznika, ID\_zamestnanca)  
VALUES

('2025-02-04', 'lietadlo', 'Vieden', 'Spanielsko', 'turizmus', 'hostel', 5, '761010/8517', 1011),  
('2025-03-08', 'lietadlo', 'Budapest', 'Francuzko', 'turizmus', 'hotel', 3, '975320/8756', 1012),  
('2025-03-10', 'lietadlo', 'Vieden', 'Amerika', NULL, 'hotel', 3, 'P23556911', 1013),  
('2025-01-15', 'lietadlo', 'Budapest', 'Slovinsko', 'navsteva', NULL, 5, '985220/6056', 1014),  
('2025-02-12', 'vlak', 'Kosice', 'Vieden', 'business', 'hotel', 4, '896129/2439', 1011),  
('2025-04-07', 'lietadlo', 'Budapest', 'Rusko', 'turizmus', 'byt', 7, 'P45612978', 1012),  
('2025-03-23', 'vlak', 'Kosice', 'Svajciarsko', 'put', NULL, 3, '945515/8779', 1013),  
('2025-03-24', 'autobus', 'Kosice', 'Varsava', 'turizmus', 'byt', 6, '680413/3787', 1014),  
('2025-04-01', 'lietadlo', 'Budapest', 'Taliansko', 'turizmus', 'hotel', 10, '741203/1506', 1011),  
('2025-05-05', 'lietadlo', 'Vieden', 'Spanielsko', 'turizmus', 'byt', 11, '885131/1743', 1012);

INSERT INTO platba

(cislo\_platby, datum\_platby, typ\_platby, sum, bonusy\_akcie, cislo\_darcekového\_preukazu, cislo\_karty\_IBAN)  
VALUES

(19872, '2024-12-05', 'bankovska karta, darcekovy preukaz', 155.28, NULL, 202400, 'SK78 1122 3344 5566 7788 9900'),  
(47932, '2024-12-10', 'hotovost', 348, 'vianocne -15%', NULL, NULL),

```
(6689, '2024-12-15', 'hotovost, darcekovy preukaz', 1335.50, 'vianocne -15%', 202401, NULL),
(4586, '2024-12-20', 'bankovska karta', 479.95, 'vianocne -15%', NULL, 'SK64 0054 0041 7432 1275
4217'),
(65487, '2025-01-05', 'darcekovy preukaz', 150, NULL, 202402, NULL),
(4239, '2025-01-07', 'bankovska karta', 1500.75, '-30% na 1.cestu mimo Shengenu', NULL, 'SK45 9876
5432 1098 7654 3210'),
(5697, '2025-01-10', 'bankovska karta', 750.50, NULL, NULL, 'SK12 1234 5678 9012 3456 7890'),
(3246, '2025-01-17', 'bankovska karta', 342.55, NULL, NULL, 'SK68 4210 8700 1714 3751 7491'),
(1256, '2025-01-23', 'bankovska karta', 850.75, NULL, NULL, 'SK68 7500 0000 0012 3456 7892'),
(21685, '2025-02-01', 'hotovost', 500, '-10% na narodeniny', NULL, NULL);
```

## 2 POHLADY S JEDNODUCHÝM NETRIVIÁLNYM SELECTOM NAD JEDNOU TABUĽKOU

1. Vypíš mená, priezviská a emaily všetkých zakázníkov, ktoré ešte nemajú poistenie a treba im ho ponúknuť.

```
CREATE VIEW bez_poistenia AS
SELECT meno, priezvisko, email
FROM zakaznik
WHERE poistenie IS NULL;
```

2. Vypíš koľko vo firme pracuje Samuel Novakov a aký má plat. Takisto vypíš ho meno.

```
CREATE VIEW samuel_novakov AS
SELECT CONCAT(meno, ' ', priezvisko) AS "Meno", (CURRENT_DATE-
zamestnanec.kedy_nastupil_doprace) AS "Dni pracovania", zamestnanec.mesacny_plat AS "Plat"
FROM zamestnanec
WHERE meno = 'Samuel' AND priezvisko = 'Novakov';
```

## 3 POHLADY SO SPÁJANÍM TABULIEK

(1X SPOJENIE ASPOŇ 2 TABULIEK, 1X SPOJENIE ASPOŇ 3 TABULIEK, 1X OUTER JOIN)

1. Vypíš meno, priezvisko, email pracovníka, ktorý predal túru do Ameriky a koľko tá túra stála.

```
CREATE VIEW kto_predal_cestu_doAmeriky AS
SELECT meno AS "Meno pracovníka", priezvisko AS "Priezvisko pracovníka", email AS "email
pracovníka", sum AS "Sum predaja"
FROM zamestnanec
JOIN tura t on zamestnanec.ID_zamestnanca = t.ID_zamestnanca
JOIN platba p on t.cislo_objednavky = p.cislo_objednavky
WHERE kam = 'Amerika';
```

2. Vypíš kam chodili ľudia, ktoré platili pomocou darčkového preukazu. Takisto vypíš ich číslo dokladu a koľko stála túra.

```
CREATE VIEW darcekovy_preukaz AS
SELECT cislo_dokladu_zakaznika AS "Doklad", kam AS "Kam", sum AS "Price"
FROM tura
JOIN platba p on tura.cislo_objednavky = p.cislo_objednavky
WHERE p.cislo_darcekového_preukazu IS NOT NULL;
```

3. Vypíš mená, priezviská pracovníkov a ich predaji: ID a cestu.



```
CREATE VIEW pracovníci_predaj AS
  SELECT meno,priezvisko, cislo_objednavky, kam
  FROM zamestnanec
  LEFT JOIN tura t on zamestnanec.ID_zamestnanca = t.ID_zamestnanca;
```

## 2 POHLADY S POUŽITÍM AGREGAČNÝCH FUNKCIÍ A/ALEBO ZOSKUPENIA

1. Vypíš koľko bolo zarobené konkrétnych rokoch.

```
CREATE VIEW sum_roky AS
  SELECT EXTRACT (YEAR from datum_platby) AS "Year", SUM(platba.sum)
  FROM platba
  GROUP BY "Year";
```

2. Vypíš zárobok po mesiacam a ukáž ich v zostupnom poradí.

```
CREATE VIEW mesiac_platby AS
  SELECT EXTRACT(MONTH FROM datum_platby) AS "Mesiac", SUM(sum) AS "Sum"
  FROM platba
  GROUP BY EXTRACT(MONTH FROM datum_platby)
  ORDER BY "Sum" DESC;
```

## TRETIA ODOVZDÁVKA

### 1 POHĽAD S POUŽITÍM MNOŽINOVÝCH OPERÁCIÍ

Vypíš všetkých zamestnancov firmy, vrátane riaditeľov a obecných zamestnancov. Stĺpce nech zahŕňajú meno a priezvisko v jednom stĺpci pomenovanom Meno, pozíciu ako Pozícia, email ako Email a telefon ako Telefon. Výsledky vypíš vzostupne podľa pozície.

```
CREATE VIEW allEmployees AS
  SELECT CONCAT(meno_riaditeľa, ' ', priezvisko_riaditeľa) AS "Meno", 'riaditeľ' AS "Pozícia",
    email_riaditeľa AS "Email",
    telefon_riaditeľa AS "Telefon"
  FROM oddelenie
  UNION
  SELECT CONCAT(meno, ' ', priezvisko) AS "Meno", pozícia AS "Pozícia", email AS "Email",
    telefon AS "Telefon"
  FROM zamestnanec
  ORDER BY "Pozícia";
```

### 2 POHĽADY S POUŽITÍM VNORENÝCH PODDOPYTŮV

1. Vypíš mená, priezviská, pozíciu a plat zamestnancov, ktorí majú plat vyšší ako price najdrahšej túry z predaných.

```
CREATE VIEW najbohatejsieZamestnanci AS
  SELECT meno, priezvisko, pozícia, mesacny_plat
  FROM zamestnanec
  WHERE mesacny_plat >
    (SELECT MAX(sum)
     FROM platba);
```

2. Vypíš mena a priezviska zákazníkov, kam, dátum nákupu túry, ktorí urobili nákup predtým, ako do zamestnania bol prijatý Robert Polak.

```
CREATE VIEW predajpredPolakom AS
  SELECT meno, priezvisko, kam, datum_platby
  FROM zakaznik
  INNER JOIN tura t ON zakaznik.cislo_dokladu = t.cislo_dokladu_zakaznika
  INNER JOIN platba p ON t.cislo_objednavky = p.cislo_objednavky
  WHERE datum_platby <
    (SELECT kedy_nastupil_doprace
     FROM zamestnanec
     WHERE zamestnanec.meno = 'Robert' AND zamestnanec.priezvisko = 'Polak'
    );
```

### SKRIPT NA VYTVORENIE TRIGGERU/TRIGGEROV, KTORÉ BUDÚ IMPLEMENTOVAŤ AUTOINKREMENTÁCIU UMELÝCH KLÚČOV

```
DROP SEQUENCE IF EXISTS prac_id_od100;
CREATE SEQUENCE prac_id_od100
INCREMENT 1
START 100;
```

```
DROP SEQUENCE IF EXISTS cislo_obj_tura;
CREATE SEQUENCE cislo_obj_tura
INCREMENT 1
START 100;
```

```
DROP SEQUENCE IF EXISTS cislo_obj_platba;
CREATE SEQUENCE cislo_obj_platba
INCREMENT 1
START 100;
```

```
CREATE OR REPLACE FUNCTION set_pracov_incr()
RETURNS TRIGGER
AS $$
BEGIN
    IF NEW.id_pracoviska IS NULL then
        NEW.id_pracoviska := nextval('prac_id_od100');
    end if;
    RETURN NEW;
END;
$$LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION set_cislo_obj_tura()
RETURNS TRIGGER
AS $$
BEGIN
    IF NEW.cislo_objednavky IS NULL then
        NEW.cislo_objednavky := nextval('cislo_obj_tura');
    end if;
    RETURN NEW;
END;
$$LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION set_cislo_obj_platba()
RETURNS TRIGGER
AS $$
BEGIN
    IF NEW.cislo_objednavky IS NULL then
        NEW.cislo_objednavky := nextval('cislo_obj_platba');
    end if;
    RETURN NEW;
END;
$$LANGUAGE plpgsql;
```

```
CREATE TRIGGER trigger_set_pracov_id
BEFORE INSERT ON oddelenie
FOR EACH ROW
EXECUTE FUNCTION set_pracov_incr();
```

```
CREATE TRIGGER trigger_set_obj_tura
```

```
BEFORE INSERT ON tura
FOR EACH ROW
EXECUTE FUNCTION set_cislo_obj_tura();
```

```
CREATE TRIGGER trigger_set_obj_platba
BEFORE INSERT ON platba
FOR EACH ROW
EXECUTE FUNCTION set_cislo_obj_platba();
```

#### SKRIPT NA ASPOŇ DVA ZMYSLUPLNÉ TRIGGRE (OKREM TRIGGEROV NA AUTOINKREMENTÁCIU)

1. Trigger zmení email aj v tabuľke Zakazník, ak on zmení email v pohľade „Bez poistenia“.

```
CREATE OR REPLACE FUNCTION changeEmailsBezPoistenia()
RETURNS TRIGGER
AS $$
BEGIN
    IF NEW.email <> OLD.email THEN
        UPDATE zakaznik
        SET email = NEW.email
        WHERE email = OLD.email;
    end if;
    RETURN NEW;
END;
$$LANGUAGE plpgsql;
```

```
CREATE TRIGGER trigger_updateEmails
INSTEAD OF UPDATE ON bez_poistenia
FOR EACH ROW
EXECUTE FUNCTION changeEmailsBezPoistenia();
```

2. Ak zmení sa číslo dokladu Zakázníka v tabuľke Zakázník, zmení sa aj v tabuľke Túra

```
CREATE OR REPLACE FUNCTION changeDokuments()
RETURNS TRIGGER
AS $$
BEGIN
    IF NEW.cislo_dokladu <> OLD.cislo_dokladu THEN
        UPDATE tura
        SET cislo_dokladu_zakaznika = NEW.cislo_dokladu
        WHERE cislo_dokladu_zakaznika = OLD.cislo_dokladu;
    end if;
    RETURN NEW;
END;
$$LANGUAGE plpgsql;
```

```
CREATE TRIGGER trigger_updateDocs
AFTER UPDATE OF cislo_dokladu ON zakaznik
FOR EACH ROW
EXECUTE FUNCTION changeDokuments();
```

## SKRIPT NA ASPOŇ JEDNU STOROVANÚ PROCEDÚRU

Zmaž človeka podľa podľa čísla dokladu.

```
CREATE OR REPLACE PROCEDURE tura_id (doklad text)
```

```
AS
```

```
$$
```

```
DECLARE
```

```
objednavka int;
```

```
BEGIN
```

```
    select cislo_objednavky into objednavka  
    from tura
```

```
    where cislo_dokladu_zakaznika = doklad;
```

```
    delete from platba
```

```
    where cislo_objednavky = objednavka;
```

```
    delete from tura
```

```
    where cislo_dokladu_zakaznika = doklad;
```

```
    delete from zakaznik
```

```
    where cislo_dokladu = doklad;
```

```
end;
```

```
$$ language plpgsql;
```

```
call tura_id('985220/6056');
```

## SKRIPT NA ASPOŇ JEDNU FUNKCIU

Vypíš meno a priezvisko riaditeľa, ktorý má najviac zamestnancov. Takisto vypíš počet ho zamestnancov. Nech to bude v tvare vety „.....má ..... zamestnancov“.

```
CREATE OR REPLACE FUNCTION riaditel_ma_zamestnancov ()
```

```
RETURNS text AS
```

```
$$
```

```
DECLARE
```

```
meno varchar(100);
```

```
pocet int;
```

```
BEGIN
```

```
    select max(kolko_zamestnancov) into pocet from oddelenie;
```

```
    select concat(meno_riaditeľa, ' ', priezvisko_riaditeľa) into meno from oddelenie
```

```
        where kolko_zamestnancov = pocet;
```

```
    return meno || ' má ' || pocet || ' zamestnancov';
```

```
end;
```

```
$$ language plpgsql;
```

```
select riaditel_ma_zamestnancov();
```