

# Dokumentace k vývoji Databáze autoservisu

Dokumentace zkouškového příkladu pro předmět 4IT218 Databáze

JAN Vítů

*2017/2018 LS*

**Uživatelské jméno: vitj07**

Body z testu SQL: 13

Body z testu DM: 19

## Obsah

1	Popis zvolené výseče světa – zadání .....	3
2	Konceptuální schéma reality .....	4
2.1	Popis atributů .....	4
3	Konceptuální datový model .....	6
4	Dokumentace databáze.....	7
4.1	Fyzický datový model .....	7
4.2	Definice relačních tabulek a souvisejících objektů .....	8
4.3	Integritní omezení .....	22
4.4	Definice přístupových práv.....	27
5	Obsah databáze .....	28
5.1	SQL příkazy pro naplnění databáze daty .....	28
5.2	Opis vložených dat.....	30

## 1 Popis zvolené výšeče světa – zadání

Tato databáze uchovává data sloužící potřebám majitele autoservisu. Autoservis byl založen 1.3.2013.

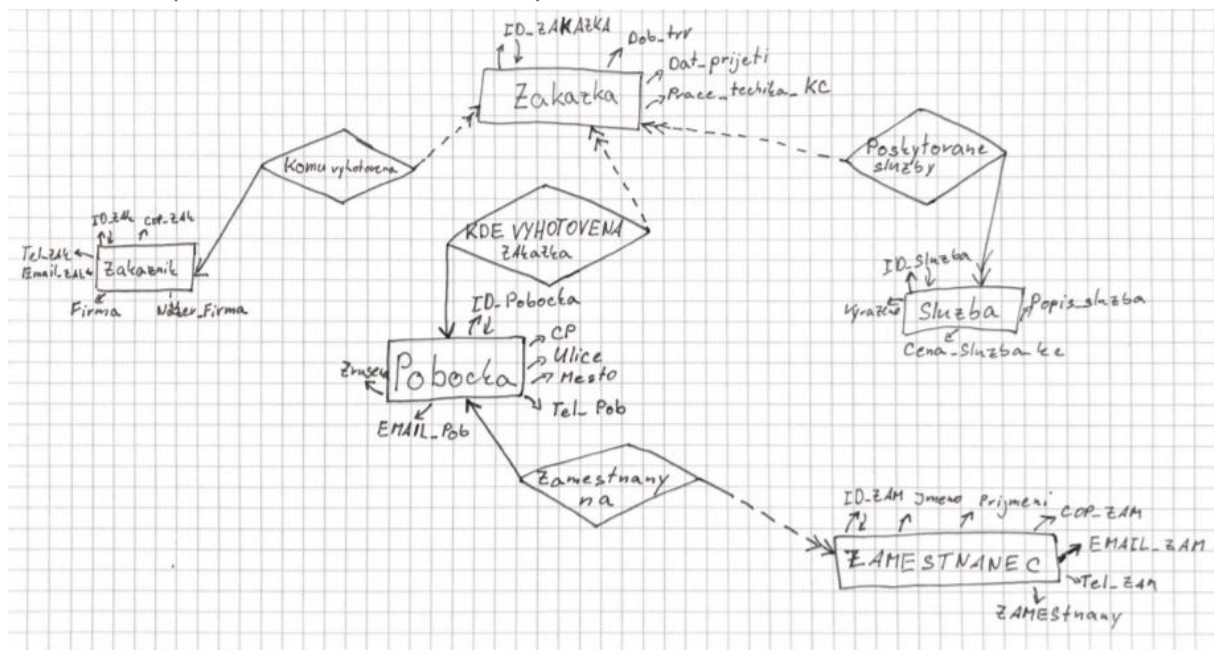
V databázi je možné evidovat zákazníky. Pro každého zákazníka se eviduje číslo zákazníka, číslo občanského průkazu zákazníka, telefonní číslo zákazníka, email zákazníka, stav zákazníka (zda zastupuje firmu či nikoli) a název firmy (povinné pole pouze pokud zákazník zastupuje firmu). Pokud je zákazník zástupce firmy má v poli firma hodnotu A, pokud firmu nezastupuje má v poli firma hodnotu N. Pokud zákazník využil jakoukoli službu (byla vyhotovena zakázka) jeho číslo je zapsáno do zakázky.

Pro zakázky je evidováno číslo zakázky, pobočka (na které byla zakázka vyhotovena), číslo zákazníka, číslo služby, dobu trvání v hodinách datum přijetí a práci technika v korunách. Nelze poskytovat vyřazenou službu nebo vyhotovit zakázku na zavřené pobočce. V jedné zakázce je možné mít minimálně jednu z poskytovaných služeb. K jednotlivým službám se eviduje číslo služby, cena služby v korunách, popis služby a stav poskytování služby (A/N). Poskytované služby nemusí být přiřazeny k zakázce, ale každá zakázka musí mít přiřazenou službu (pokud službu nikdo doposud nevyužil, prozatím není přiřazena k zakázce).

Dále jsou evidovány pobočky, které obsahují číslo pobočky, číslo popisné, ulici, město, telefonní číslo pobočky, email pobočky a stav zrušení pobočky (A/N). Na pobočkách mohou být zaměstnanci. Pro zaměstnance se eviduje číslo zaměstnance číslo pobočky (na které je zaměstnanec zaměstnán), jméno zaměstnance, příjmení zaměstnance, číslo občanského průkazu zaměstnance, email zaměstnance, telefonní číslo zaměstnance a stav zaměstnanosti (A/N). Zaměstnanec nemůže být zaměstnán na pobočce které je zrušena. Pro propuštěné zaměstnance je vytvořena nultá pobočka (fiktivní pobočka), ve které jsou evidováni.

Entity: Pobočka, poskytované služby, služby, zakázka, zákazník, zaměstnanec.

## 2 Konceptuální schéma reality



### 2.1 Popis atributů

#### a) ZAKAZKA

ID\_ZAKAZKA: primární identifikátor  
 DOB\_TRV: doba trvání práce technika  
 DAT\_PRIJETI: datum přijetí zakázky  
 PRÁCE\_TECHNIKA\_KC: finanční ohodnocení práce technika

#### b) ZAKAZNIK

ID\_ZAK: primární identifikátor  
 COP\_ZAK: číslo občanského průkazu zákazníka  
 TEL\_ZAK: telefonní číslo zákazníka  
 EMAIL\_ZAK: email zákazníka  
 FIRMA: identifikace za osoba je jednatelem firmy či nikoli (A/N)  
 NAZEV\_FIRMA: název firmy (při stavu atributu firma A je povinný)

#### c) POBOCKA

ID\_POBOCKA: primární identifikátor  
 CP: číslo popisné pobočky  
 ULICE: ulice pobočky  
 MESTO: město pobočky  
 TEL\_POB: telefonní číslo pobočky  
 EMAIL\_POB: email pobočky  
 ZRUSENA: jednoznačná identifikace, zda je pobočka zrušena či nikoli (A/N)

#### d) ZAMESTNANEC

ID\_ZAM: primární identifikátor  
 JMENO: křestní jméno zaměstnance  
 PRIJMENI: příjmení zaměstnance

COP\_ZAM: číslo občanského průkazu zaměstnance

EMAIL\_ZAM: email zaměstnance (nepovinný údaj)

TEL\_ZAM: telefonní číslo zaměstnance

ZAMESTNANY: jednoznačná identifikace, zda je zaměstnanec ve firmě zaměstnán či nikoli (A/N)

**e) SLUZBA**

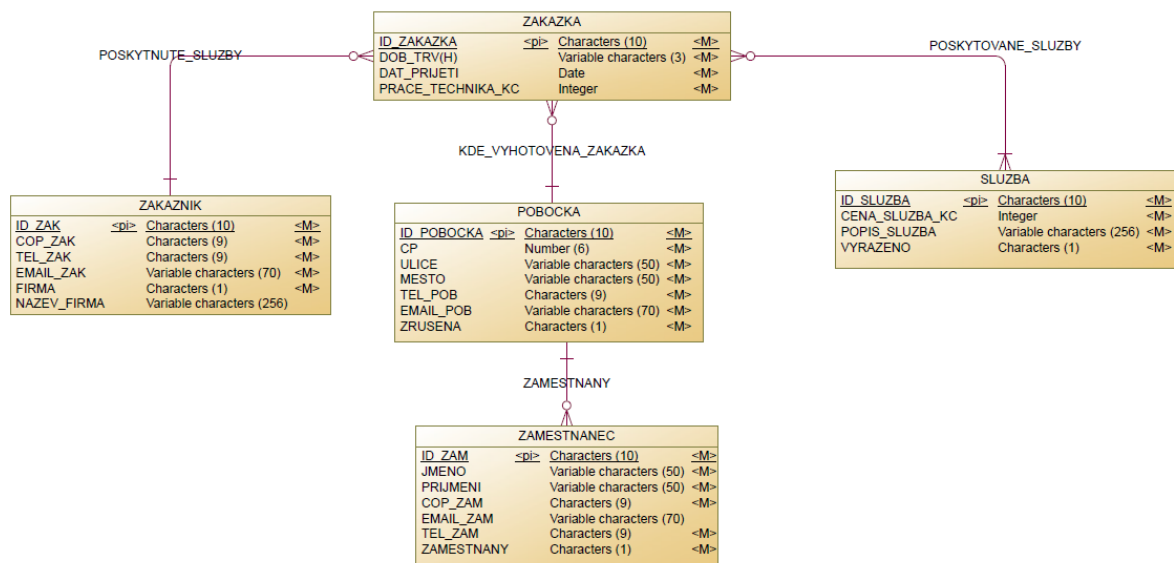
ID\_SLUZBA: primární identifikátor

POPIS\_SLUZBA: popis služby

CENA\_SLUZBA\_KC: cena služby v korunách

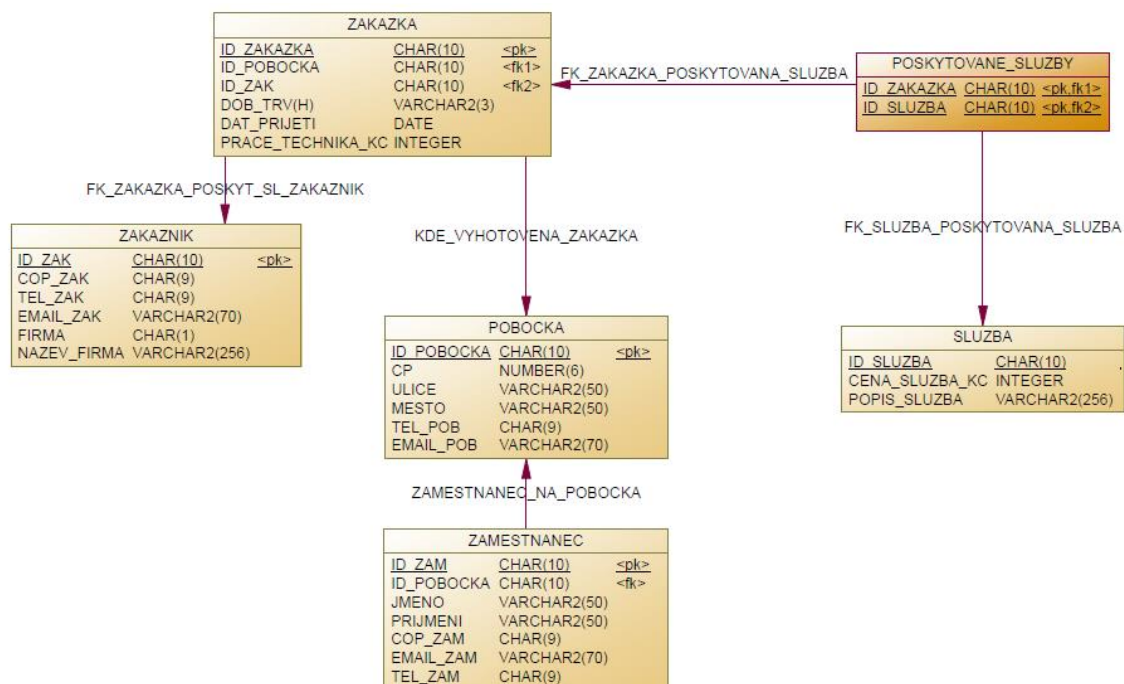
VYRAZENO: jednoznačná identifikace, zda je služba vyřazena či nikoli (A/N)

### 3 Konceptuální datový model



## 4 Dokumentace databáze

### 4.1 Fyzický datový model



Tabulka POSKYTOVANE\_SLUZBY slouží ke spojení tabulky ZAKAZKA a SLUZBA. Tato tabulka je využita pro přiřazení služby k zakázce.

## 4.2 Definice relačních tabulek a souvisejících objektů

```
/*=====*/  
/* DBMS name:   ORACLE Version 10g           */  
/* Created on:  05.05.2018 21:55:19           */  
/*=====*/
```

-- Type package declaration

create or replace package PDTypes

as

    TYPE ref\_cursor IS REF CURSOR;

end;

-- Integrity package declaration

create or replace package IntegrityPackage AS

    procedure InitNestLevel;

    function GetNestLevel return number;

    procedure NextNestLevel;

    procedure PreviousNestLevel;

end IntegrityPackage;

/

-- Integrity package definition

create or replace package body IntegrityPackage AS

    NestLevel number;

-- Procedure to initialize the trigger nest level

    procedure InitNestLevel is

    begin

        NestLevel := 0;

    end;



-- Function to return the trigger nest level

function GetNestLevel return number is

begin

if NestLevel is null then

    NestLevel := 0;

end if;

return(NestLevel);

end;

-- Procedure to increase the trigger nest level

procedure NextNestLevel is

begin

if NestLevel is null then

    NestLevel := 0;

end if;

NestLevel := NestLevel + 1;

end;

-- Procedure to decrease the trigger nest level

procedure PreviousNestLevel is

begin

NestLevel := NestLevel - 1;

end;

end IntegrityPackage;

/

drop trigger TRIG3

/

drop trigger TRIG5

/

drop trigger TRIG7

/

drop trigger TRIG1

/

drop trigger TRIG2

/

drop trigger TRIG6

/

alter table POSKYTOVANE\_SLUZBY

drop constraint FK\_ZAKAZKA\_POSKYTOVANA\_SLUZBA

/

alter table POSKYTOVANE\_SLUZBY

drop constraint FK\_SLUZBA\_POSKYTOVANA\_SLUZBA

/

alter table ZAKAZKA

drop constraint KDE\_VYHOTOVENA\_ZAKAZKA

/

alter table ZAKAZKA

drop constraint FK\_ZAKAZKA\_POSKYT\_SL\_ZAKAZNIK

/

alter table ZAMESTNANEC

drop constraint ZAMESTNANEC\_NA\_POBOCKA

/

drop index EMAIL\_POB

/

drop index TEL\_POB

/

drop table POBOCKA cascade constraints

/

drop index POSKYTOVANE\_SLUZBY2\_FK

/

drop index POSKYTOVANE\_SLUZBY\_FK

/

drop table POSKYTOVANE\_SLUZBY cascade constraints

/

drop table SLUZBA cascade constraints

/

drop index POSKYTNUTE\_SLUZBY\_FK

/

drop index KDE\_VYHOTOVENA\_ZAKAZKA\_FK

/

drop table ZAKAZKA cascade constraints

/

drop index NAZEV\_FIRMA

/

```
drop index EMAIL_ZAK
```

```
/
```

```
drop index TEL_ZAK
```

```
/
```

```
drop index COP_ZAK
```

```
/
```

```
drop table ZAKAZNIK cascade constraints
```

```
/
```

```
drop index EMAIL_ZAM
```

```
/
```

```
drop index TEL_ZAM
```

```
/
```

```
drop index COP_ZAM
```

```
/
```

```
drop index ZAMESTNANY_FK
```

```
/
```

```
drop table ZAMESTNANEC cascade constraints
```

```
/
```

```
/*=====*/
```

```
/* Table: POBOCKA */
```

```
/*=====*/
```

```
create table POBOCKA (
```

```
    ID_POBOCKA    CHAR(10)    not null
```

```
constraint CKC_ID_POBOCKA_POBOCKA check (regexp_like (ID_POBOCKA, '^[0-9]{10}$')),
CP          NUMBER(6)          not null
constraint CKC_CP_POBOCKA check (regexp_like (CP, '^[0-9]+$')),
ULICE       VARCHAR2(50)       not null,
MESTO       VARCHAR2(50)       not null,
TEL_POB     CHAR(9)            not null
constraint CKC_TEL_POB_POBOCKA check (TEL_POB >= '0' and regexp_like (TEL_POB, '^[0-9]{9}')),
EMAIL_POB   VARCHAR2(70)       not null,
ZRUSENA     CHAR(1)            default 'N' not null
constraint CKC_ZRUSENA_POBOCKA check (ZRUSENA in ('A','N')),
constraint PK_POBOCKA primary key (ID_POBOCKA)
)
/

/*=====*/
/* Index: TEL_POB          */
/*=====*/
create unique index TEL_POB on POBOCKA (
    TEL_POB ASC
)
/

/*=====*/
/* Index: EMAIL_POB       */
/*=====*/
create unique index EMAIL_POB on POBOCKA (
    EMAIL_POB ASC
)
/

/*=====*/
/* Table: POSKYTOVANE_SLUZBY */
/*=====*/
```

```
create table POSKYTOVANE_SLUZBY (  
  ID_ZAKAZKA      CHAR(10)          not null  
    constraint CKC_ID_ZAKAZKA_POSKYTOV check (regexp_like (ID_ZAKAZKA, '^[0-9]{10}$')),  
  ID_SLUZBA       CHAR(10)          not null  
    constraint CKC_ID_SLUZBA_POSKYTOV check (regexp_like (ID_SLUZBA, '^[0-9]{10}$')),  
  constraint PK_POSKYTOVANE_SLUZBY primary key (ID_ZAKAZKA, ID_SLUZBA)  
)  
/
```

```
/*=====*/  
/* Index: POSKYTOVANE_SLUZBY_FK          */  
/*=====*/  
create index POSKYTOVANE_SLUZBY_FK on POSKYTOVANE_SLUZBY (  
  ID_ZAKAZKA ASC  
)  
/
```

```
/*=====*/  
/* Index: POSKYTOVANE_SLUZBY2_FK        */  
/*=====*/  
create index POSKYTOVANE_SLUZBY2_FK on POSKYTOVANE_SLUZBY (  
  ID_SLUZBA ASC  
)  
/
```

```
/*=====*/  
/* Table: SLUZBA                        */  
/*=====*/  
create table SLUZBA (  
  ID_SLUZBA       CHAR(10)          not null  
    constraint CKC_ID_SLUZBA_SLUZBA check (regexp_like (ID_SLUZBA, '^[0-9]{10}$')),  
  CENA_SLUZBA_KC   INTEGER          not null,  
  POPIS_SLUZBA     VARCHAR2(256)    not null,
```

```
VYRAZENO      CHAR(1)      default 'N' not null
    constraint CKC_VYRAZENO_SLUZBA check (VYRAZENO in ('A','N')),
    constraint PK_SLUZBA primary key (ID_SLUZBA)
)
/

/*=====*/
/* Table: ZAKAZKA                                */
/*=====*/

create table ZAKAZKA (
    ID_ZAKAZKA      CHAR(10)      not null
        constraint CKC_ID_ZAKAZKA_ZAKAZKA check (regexp_like (ID_ZAKAZKA, '^[0-9]{10}$')),
    ID_POBOCKA      CHAR(10)      not null,
    ID_ZAK           CHAR(10)      not null,
    DOB_TRV         VARCHAR2(3)    not null,
    DAT_PRIJETI      DATE          not null
        constraint CKC_DAT_PRIJETI_ZAKAZKA check (DAT_PRIJETI >= '01-MAR-2013'),
    PRACE_TECHNIKA_KC integer       not null,
    constraint PK_ZAKAZKA primary key (ID_ZAKAZKA)
)
/

/*=====*/
/* Index: KDE_VYHOTOVENA_ZAKAZKA_FK              */
/*=====*/

create index KDE_VYHOTOVENA_ZAKAZKA_FK on ZAKAZKA (
    ID_POBOCKA ASC
)
/

/*=====*/
/* Index: POSKYTNUTE_SLUZBY_FK                    */
/*=====*/
```

```
create index POSKYTNUTE_SLUZBY_FK on ZAKAZKA (
  ID_ZAK ASC
)
/

/*=====*/
/* Table: ZAKAZNIK */
/*=====*/

create table ZAKAZNIK (
  ID_ZAK      CHAR(10)      not null
    constraint CKC_ID_ZAK_ZAKAZNIK check (regexp_like (ID_ZAK, '^[0-9]{10}$')),
  COP_ZAK     CHAR(9)       not null
    constraint CKC_COP_ZAK_ZAKAZNIK check (COP_ZAK >= '0' and regexp_like (COP_ZAK, '^[0-9]{9}$')),
  TEL_ZAK     CHAR(9)       not null
    constraint CKC_TEL_ZAK_ZAKAZNIK check (TEL_ZAK >= '0' and regexp_like (TEL_ZAK, '^[0-9]{9}$')),
  EMAIL_ZAK   VARCHAR2(70)  not null,
  FIRMA       CHAR(1)       default 'N' not null
    constraint CKC_FIRMA_ZAKAZNIK check (FIRMA in ('A','N')),
  NAZEV_FIRMA VARCHAR2(256),
  constraint PK_ZAKAZNIK primary key (ID_ZAK)
)
/

/*=====*/
/* Index: COP_ZAK */
/*=====*/

create unique index COP_ZAK on ZAKAZNIK (
  COP_ZAK ASC
)
/

/*=====*/
```



```
/* Index: TEL_ZAK                                */
/*=====*/
create unique index TEL_ZAK on ZAKAZNIK (
    TEL_ZAK ASC
)
/

/*=====*/
/* Index: EMAIL_ZAK                                */
/*=====*/
create unique index EMAIL_ZAK on ZAKAZNIK (
    EMAIL_ZAK ASC
)
/

/*=====*/
/* Index: NAZEV_FIRMA                                */
/*=====*/
create unique index NAZEV_FIRMA on ZAKAZNIK (
    NAZEV_FIRMA ASC
)
/

/*=====*/
/* Table: ZAMESTNANEC                                */
/*=====*/
create table ZAMESTNANEC (
    ID_ZAM          CHAR(10)          not null
        constraint CKC_ID_ZAM_ZAMESTNA check (regexp_like (ID_ZAM, '^[0-9]{10}$')),
    ID_POBOCKA      CHAR(10)          not null
        constraint CKC_ID_POBOCKA_ZAMESTNA check (regexp_like (ID_POBOCKA, '^[0-9]{10}$')),
    JMENO           VARCHAR2(50)      not null,
    PRIJMENI        VARCHAR2(50)      not null,
```

```
COP_ZAM      CHAR(9)      not null
    constraint CKC_COP_ZAM_ZAMESTNA check (regexp_like (COP_ZAM, '^[0-9]{9}$')),
EMAIL_ZAM     VARCHAR2(70),
TEL_ZAM       CHAR(9)      not null
    constraint CKC_TEL_ZAM_ZAMESTNA check (regexp_like (TEL_ZAM, '^[0-9]{9}$')),
ZAMESTNANY    CHAR(1)      default 'A' not null
    constraint CKC_ZAMESTNANY_ZAMESTNA check (ZAMESTNANY in ('A','N')),
constraint PK_ZAMESTNANEC primary key (ID_ZAM)
)
/

/*=====*/
/* Index: ZAMESTNANY_FK                */
/*=====*/
create index ZAMESTNANY_FK on ZAMESTNANEC (
    ID_POBOCKA ASC
)
/

/*=====*/
/* Index: COP_ZAM                      */
/*=====*/
create unique index COP_ZAM on ZAMESTNANEC (
    COP_ZAM ASC
)
/

/*=====*/
/* Index: TEL_ZAM                      */
/*=====*/
create unique index TEL_ZAM on ZAMESTNANEC (
    TEL_ZAM ASC
)
/
```

/

/\*=====\*/

/\* Index: EMAIL\_ZAM \*/

/\*=====\*/

create unique index EMAIL\_ZAM on ZAMESTNANEC (

EMAIL\_ZAM ASC

)

/

alter table POSKYTOVANE\_SLUZBY

add constraint FK\_ZAKAZKA\_POSKYTOVANA\_SLUZBA foreign key (ID\_ZAKAZKA)

references ZAKAZKA (ID\_ZAKAZKA)

/

alter table POSKYTOVANE\_SLUZBY

add constraint FK\_SLUZBA\_POSKYTOVANA\_SLUZBA foreign key (ID\_SLUZBA)

references SLUZBA (ID\_SLUZBA)

/

alter table ZAKAZKA

add constraint KDE\_VYHOTOVENA\_ZAKAZKA foreign key (ID\_POBOCKA)

references POBOCKA (ID\_POBOCKA)

/

alter table ZAKAZKA

add constraint FK\_ZAKAZKA\_POSKYT\_SL\_ZAKAZNIK foreign key (ID\_ZAK)

references ZAKAZNIK (ID\_ZAK)

/

alter table ZAMESTNANEC

add constraint ZAMESTNANEC\_NA\_POBOCKA foreign key (ID\_POBOCKA)

references POBOCKA (ID\_POBOCKA)

/

```
create or replace TRIGGER TRIG3
BEFORE INSERT OR UPDATE ON POSKYTOVANE_SLUZBY
FOR EACH ROW
DECLARE
SluzbaVYR CHAR(1);
BEGIN
    SELECT SLUZBA."VYRAZENO" into SluzbaVYR
    FROM SLUZBA
    WHERE SLUZBA."ID_SLUZBA"=:new."ID_SLUZBA";
    IF(SluzbaVYR='A')THEN
        RAISE_APPLICATION_ERROR(-20000,'Nelze poskytovat vyrazenou sluzbu');
    END IF;
END;
```

/

```
create or replace TRIGGER TRIG5
BEFORE INSERT OR UPDATE ON ZAKAZKA
FOR EACH ROW
DECLARE
PobockaZRUS CHAR(1);
BEGIN
    SELECT POBOCKA."ZRUSENA" into PobockaZRUS
    FROM POBOCKA
    WHERE POBOCKA."ID_POBOCKA"=:new."ID_POBOCKA";
    IF(PobockaZRUS='A')THEN
        RAISE_APPLICATION_ERROR(-20000,'Nelze provést službu na zrušené pobočce');
    END IF;
END;
```

/

```
create or replace TRIGGER TRIG7
BEFORE INSERT OR UPDATE ON ZAKAZKA
FOR EACH ROW
BEGIN
  IF( :new.DAT_PRIJETI > sysdate )
  THEN
    RAISE_APPLICATION_ERROR( -20001, 'Nelze zadat budoucí den' );
  END IF;
END;
/
```

```
create or replace TRIGGER TRIG1
BEFORE INSERT OR UPDATE ON ZAKAZNIK
FOR EACH ROW
BEGIN
  IF(:new.FIRMA='A' and :new.NAZEV_FIRMA is NULL) THEN
    RAISE_APPLICATION_ERROR(-20000, 'Pro firmu musí být vyplněný název');
  END IF;
END;
/
```

```
create or replace TRIGGER TRIG2
BEFORE UPDATE OR INSERT ON ZAKAZNIK
FOR EACH ROW
BEGIN
  IF(:new.FIRMA='N' AND :new.NAZEV_FIRMA IS NOT NULL) THEN
    RAISE_APPLICATION_ERROR(-20000, 'Člověk co nezastupuje firmu nemůže mít vyplněné pole
název firmy');
  END IF;
```

END;

/

```
create or replace TRIGGER TRIG6
BEFORE INSERT OR UPDATE ON ZAMESTNANEC
FOR EACH ROW
DECLARE
PobockaCIS CHAR(1);
BEGIN
    SELECT ZRUSENA into PobockaCIS
    FROM POBOCKA
    WHERE POBOCKA."ID_POBOCKA"=:new."ID_POBOCKA";
    IF(PobockaCIS='A') THEN
        RAISE_APPLICATION_ERROR(-20000,'Nelze zaměstnat zaměstnance na zrušené pobočce');
    END IF;
END;
/
```

#### 4.3 Integritní omezení

Tabulka ZAKAZKA

Primární klíč je tvořen atributem ID\_ZAKAZKA.

- constraint PK\_ZAKAZKA primary key (ID\_ZAKAZKA)

*Doménová integrita*

**Nelze vyhotovit zakázku na zrušené pobočce.**

- create or replace TRIGGER TRIG5  
BEFORE INSERT OR UPDATE ON ZAKAZKA  
FOR EACH ROW  
DECLARE  
PobockaZRUS CHAR(1);  
BEGIN  
 SELECT POBOCKA."ZRUSENA" into PobockaZRUS  
 FROM POBOCKA  
 WHERE POBOCKA."ID\_POBOCKA"=:new."ID\_POBOCKA";  
 IF(PobockaZRUS='A')THEN  
 RAISE\_APPLICATION\_ERROR(-20000,'Nelze provést službu na zrušené pobočce');  
 END IF;

END;

#### **Nelze přijmout zakázku později než v dnešní den**

- create or replace TRIGGER TRIG7  
BEFORE INSERT OR UPDATE ON ZAKAZKA  
FOR EACH ROW  
BEGIN  
IF( :new.DAT\_PRIJETI > sysdate )  
THEN  
RAISE\_APPLICATION\_ERROR( -20001, 'Nelze zadat budoucí den' );  
END IF;  
END;

#### **Zakázku nelze přijmout dříve než v datum založení firmy**

- constraint CKC\_DAT\_PRIJETI\_ZAKAZKA check (DAT\_PRIJETI >= '01-MAR-2013')

#### **ID\_ZAKAZKA je omezeno na 10 číselných znaků**

- constraint CKC\_ID\_ZAKAZKA\_ZAKAZKA check (regexp\_like (ID\_ZAKAZKA, '^[0-9]{10}\$'))

#### *Referenční integrita*

Lze smazat řádek v tabulce zakázka, ale služba, zákazník ani pobočka se nesmažou. Cizími klíči v tabule zakázka jsou ID\_SLUZBA, ID\_ZAK a ID\_POBOCKA. Referenční integrita je RESTRICT.

- alter table ZAKAZKA  
add constraint FK\_ZAKAZKA\_POSKYT\_SL\_ZAKAZNIK foreign key (ID\_ZAK)  
references ZAKAZNIK (ID\_ZAK)
- alter table ZAKAZKA  
add constraint KDE\_VYHOTOVENA\_ZAKAZKA foreign key (ID\_POBOCKA)  
references POBOCKA (ID\_POBOCKA)

#### *Tabulka ZAKAZNIK*

Primární klíč je tvořen atributem ID\_ZAK.

- constraint PK\_ZAKAZNIK primary key (ID\_ZAK)

#### *Doménová integrita*

#### **ID\_ZAK je omezeno na 10 číselných znaků**

- constraint CKC\_ID\_ZAK\_ZAKAZNIK check (regexp\_like (ID\_ZAK, '^[0-9]{10}\$'))

#### **COP\_ZAK je omezeno na 9 číselných znaků**

- constraint CKC\_COP\_ZAK\_ZAKAZNIK check (COP\_ZAK >= '0' and regexp\_like (COP\_ZAK, '^[0-9]{9}\$'))

#### **TEL\_ZAK je omezeno na 9 číselných znaků**

- constraint CKC\_TEL\_ZAK\_ZAKAZNIK check (TEL\_ZAK >= '0' and regexp\_like (TEL\_ZAK, '^[0-9]{9}\$'))

**Atribut FIRMA, který zabezpečuje identifikaci, zda zákazník zastupuje firmu či nikoli je omezen na A, nebo N**

- constraint CKC\_FIRMA\_ZAKAZNIK check (FIRMA in ('A','N'))

**Pokud zákazník zastupuje firmu musí být vyplněno pole název firmy a pokud nezastupuje zákazník firmu tak nesmí být vyplněno pole název firmy**

- create or replace TRIGGER TRIG1  
BEFORE INSERT OR UPDATE ON ZAKAZNIK  
FOR EACH ROW  
BEGIN  
IF(:new.FIRMA='A' and :new.NAZEV\_FIRMA is NULL) THEN  
RAISE\_APPLICATION\_ERROR(-20000, 'Pro firmu musí být vyplněný název');  
END IF;  
END;
- create or replace TRIGGER TRIG2  
BEFORE UPDATE OR INSERT ON ZAKAZNIK  
FOR EACH ROW  
BEGIN  
IF(:new.FIRMA='N' AND :new.NAZEV\_FIRMA IS NOT NULL) THEN  
RAISE\_APPLICATION\_ERROR(-20000, 'Člověk co nezastupuje firmu nemůže mít vyplněné pole název firmy');  
END IF;  
END;

#### *Referenční integrita*

Nelze smazat zákazníka, pokud je přiřazený k zakázce. Referenční integrita je RESTRICT.

- alter table ZAKAZKA  
add constraint FK\_ZAKAZKA\_POSKYT\_SL\_ZAKAZNIK foreign key (ID\_ZAK)  
references ZAKAZNIK (ID\_ZAK)

#### *Tabulka POBOCKA*

Primární klíč je tvořen atributem ID\_POBOCKA.

- constraint PK\_POBOCKA primary key (ID\_POBOCKA)

#### *Doménová integrita*

**ID\_POBOCKA je omezen na 10 číselných znaků**

- constraint CKC\_ID\_POBOCKA\_POBOCKA check (regexp\_like (ID\_POBOCKA, '^[0-9]{10}\$'))

**CP je omezen pouze na číselné znaky**

- constraint CKC\_CP\_POBOCKA check (regexp\_like (CP, '^[0-9]+\$'))

**TEL\_POB je omezen na 9 číselných znaků**

- constraint CKC\_TEL\_POB\_POBOCKA check (TEL\_POB >= '0' and regexp\_like (TEL\_POB, '^[0-9]{9}\$'))

**Atribut ZRUSENA je omezen na dvě hodnoty A a N**

- constraint CKC\_ZRUSENA\_POBOCKA check (ZRUSENA in ('A','N'))



### *Referenční integrita*

Nelze smazat pobočku, pokud na ní jsou zaměstnanci. Referenční integrita je RESTRICT.

- alter table ZAMESTNANEC  
add constraint ZAMESTNANEC\_NA\_POBOCKA foreign key (ID\_POBOCKA)  
references POBOCKA (ID\_POBOCKA)
- alter table ZAKAZKA  
add constraint KDE\_VYHOTOVENA\_ZAKAZKA foreign key (ID\_POBOCKA)  
references POBOCKA (ID\_POBOCKA)

### *Tabulka ZAMESTNANEC*

Primární klíč je tvořen atributem ID\_ZAM.

- constraint PK\_ZAMESTNANEC primary key (ID\_ZAM)

### *Doménová integrita*

**ID\_ZAM je omezen na 10 číselných znaků**

- constraint CKC\_ID\_ZAM\_ZAMESTNA check (regexp\_like (ID\_ZAM, '^[0-9]{10}\$'))

**COP\_ZAM je omezen na 9 číselných znaků**

- constraint CKC\_COP\_ZAM\_ZAMESTNA check (regexp\_like (COP\_ZAM, '^[0-9]{9}\$'))

**TEL\_ZAM je omezen na 9 číselných znaků**

- constraint CKC\_TEL\_ZAM\_ZAMESTNA check (regexp\_like (TEL\_ZAM, '^[0-9]{9}\$'))

**Atribut ZAMESTNANY je omezen na dvě hodnoty A a N**

- constraint CKC\_ZAMESTNANY\_ZAMESTNA check (ZAMESTNANY in ('A','N'))

**Zaměstnanec nemůže být zaměstnán na zrušené pobočce**

- create or replace TRIGGER TRIG6  
BEFORE INSERT OR UPDATE ON ZAMESTNANEC  
FOR EACH ROW  
DECLARE  
PobockaCIS CHAR(1);  
BEGIN  
SELECT ZRUSENA into PobockaCIS  
FROM POBOCKA  
WHERE POBOCKA."ID\_POBOCKA"=:new."ID\_POBOCKA";  
IF(PobockaCIS='A') THEN  
RAISE\_APPLICATION\_ERROR(-20000,'Nelze zaměstnat zaměstnance na zrušené pobočce');  
END IF;  
END;

### *Referenční integrita*

Zaměstnanec lze smazat. Při smazání zaměstnance se nesmaže pobočka. Cizí klíč je ID\_POBOCKA. Referenční integrita je RESTRICT.

- alter table ZAMESTNANEC  
add constraint ZAMESTNANEC\_NA\_POBOCKA foreign key (ID\_POBOCKA)  
references POBOCKA (ID\_POBOCKA)

### *Tabulka SLUZBY*

Primární klíč je tvořen atributem ID\_SLUZBA.

- constraint PK\_SLUZBA primary key (ID\_SLUZBA)

### *Doménová integrita*

**ID\_SLUZBA je omezen na 10 číselných znaků**

- constraint CKC\_ID\_SLUZBA\_SLUZBA check (regexp\_like (ID\_SLUZBA, '^[0-9]{10}\$'))

**Atribut ZRUSENA je omezen na dvě hodnoty A a N**

- constraint CKC\_VYRAZENO\_SLUZBA check (VYRAZENO in ('A','N'))

### *Referenční integrita*

Nelze smazat službu která je přiřazena k zakázce. Referenční integrita je RESTRICT.

- alter table POSKYTOVANE\_SLUZBY  
add constraint FK\_SLUZBA\_POSKYTOVANA\_SLUZBA foreign key (ID\_SLUZBA)  
references SLUZBA (ID\_SLUZBA)

### *Tabulka POSKYTOVANE\_SLUZBY*

Primární klíč je tvořen kombinací dvou cizích klíčů ID\_ZAKAZKA a ID\_SLUZBA.

- constraint PK\_POSKYTOVANE\_SLUZBY primary key (ID\_ZAKAZKA, ID\_SLUZBA)

### *Doménová integrita*

**ID\_SLUZBA je omezen na 10 číselných znaků**

- constraint CKC\_ID\_SLUZBA\_POSKYTOV check (regexp\_like (ID\_SLUZBA, '^[0-9]{10}\$'))

**ID\_ZAKAZKA je omezena na 10 číselných znaků**

- constraint CKC\_ID\_ZAKAZKA\_POSKYTOV check (regexp\_like (ID\_ZAKAZKA, '^[0-9]{10}\$'))

**Nelze poskytovat službu která je vyřazena**

```
create or replace TRIGGER TRIG3
BEFORE INSERT OR UPDATE ON POSKYTOVANE_SLUZBY
FOR EACH ROW
DECLARE
SluzbaVYR CHAR(1);
BEGIN
SELECT SLUZBA."VYRAZENO" into SluzbaVYR
FROM SLUZBA
WHERE SLUZBA."ID_SLUZBA"=:new."ID_SLUZBA";
IF(SluzbaVYR='A')THEN
```

```
RAISE_APPLICATION_ERROR(-20000,'Nelze poskytovat vyrazenou službu');  
END IF;  
END;
```

#### *Referenční integrita*

Lze smazat celý řádek. Při smazání záznamu se nesmaže služba ani zakázka. Referenční integrita je RESTRICT.

- alter table POSKYTOVANE\_SLUZBY  
add constraint FK\_SLUZBA\_POSKYTOVANA\_SLUZBA foreign key (ID\_SLUZBA)  
references SLUZBA (ID\_SLUZBA)
- alter table POSKYTOVANE\_SLUZBY  
add constraint FK\_ZAKAZKA\_POSKYTOVANA\_SLUZBA foreign key (ID\_ZAKAZKA)  
references ZAKAZKA (ID\_ZAKAZKA)

#### 4.4 Definice přístupových práv

```
GRANT SELECT, UPDATE, INSERT, DELETE ON POBOCKA TO IT218;
```

```
GRANT SELECT, UPDATE, INSERT, DELETE ON POSKYTOVANE_SLUZBY TO IT218;
```

```
GRANT SELECT, UPDATE, INSERT, DELETE ON SLUZBA TO IT218;
```

```
GRANT SELECT, UPDATE, INSERT, DELETE ON ZAKAZKA TO IT218;
```

```
GRANT SELECT, UPDATE, INSERT, DELETE ON ZAKAZNIK TO IT218;
```

```
GRANT SELECT, UPDATE, INSERT, DELETE ON ZAMESTNANEC TO IT218;
```

```
GRANT SELECT ON POBOCKA TO STUDENT;
```

```
GRANT SELECT ON POSKYTOVANE_SLUZBY TO STUDENT;
```

```
GRANT SELECT ON SLUZBA TO STUDENT;
```

```
GRANT SELECT ON ZAKAZKA TO STUDENT;
```

```
GRANT SELECT ON ZAKAZNIK TO STUDENT;
```

```
GRANT SELECT ON ZAMESTNANEC TO STUDENT;
```

## 5 Obsah databáze

### 5.1 SQL příkazy pro naplnění databáze daty

#### **POBOCKA**

```
INSERT INTO "VITJ07"."POBOCKA" (ID_POBOCKA, CP, ULICE, MESTO, TEL_POB, EMAIL_POB, ZRUSENA) VALUES ('0000000000', '0', 'NULTA', 'NULTA', '0000000000', 'NULTA', 'N')
```

```
INSERT INTO "VITJ07"."POBOCKA" (ID_POBOCKA, CP, ULICE, MESTO, TEL_POB, EMAIL_POB, ZRUSENA) VALUES ('0000000001', '887', 'Šumavská', 'Kaplice', '727833929', 'kapliceServis@auto.cz', 'N')
```

```
INSERT INTO "VITJ07"."POBOCKA" (ID_POBOCKA, CP, ULICE, MESTO, TEL_POB, EMAIL_POB, ZRUSENA) VALUES ('0000000002', '955', 'Chemická', 'Praha', '981837224', 'kunraticeServis@auto.cz', 'N')
```

```
INSERT INTO "VITJ07"."POBOCKA" (ID_POBOCKA, CP, ULICE, MESTO, TEL_POB, EMAIL_POB, ZRUSENA) VALUES ('0000000003', '114', 'Lanova', 'České Budějovice', '324345189', 'ceskebudejoviceServis@auto.cz', 'N')
```

```
INSERT INTO "VITJ07"."POBOCKA" (ID_POBOCKA, CP, ULICE, MESTO, TEL_POB, EMAIL_POB, ZRUSENA) VALUES ('0000000004', '1651', 'Budějovická', 'Písek', '887364220', 'pisekServis@auto.cz', 'N')
```

#### **SLUZBA**

```
INSERT INTO "VITJ07"."SLUZBA" (ID_SLUZBA, CENA_SLUZBA_KC, POPIS_SLUZBA, VYRAZENO) VALUES ('0000000001', '1000', 'Výměna oleje', 'N')
```

```
INSERT INTO "VITJ07"."SLUZBA" (ID_SLUZBA, CENA_SLUZBA_KC, POPIS_SLUZBA, VYRAZENO) VALUES ('0000000002', '3000', 'Výměna brzdového kotouče a brzdových destiček', 'N')
```

```
INSERT INTO "VITJ07"."SLUZBA" (ID_SLUZBA, CENA_SLUZBA_KC, POPIS_SLUZBA, VYRAZENO) VALUES ('0000000003', '15000', 'Oprava zadřeného motoru', 'N')
```

```
INSERT INTO "VITJ07"."SLUZBA" (ID_SLUZBA, CENA_SLUZBA_KC, POPIS_SLUZBA, VYRAZENO) VALUES ('0000000004', '3500', 'Servisní kontrola', 'N')
```

```
INSERT INTO "VITJ07"."SLUZBA" (ID_SLUZBA, CENA_SLUZBA_KC, POPIS_SLUZBA, VYRAZENO) VALUES ('0000000005', '700', 'Přezutí pneumatik', 'N')
```

#### **ZAKAZNIK**

```
INSERT INTO "VITJ07"."ZAKAZNIK" (ID_ZAK, COP_ZAK, TEL_ZAK, EMAIL_ZAK, FIRMA) VALUES ('0000000001', '827346521', '896532182', 'JosefNovak@seznam.cz', 'N')
```

```
INSERT INTO "VITJ07"."ZAKAZNIK" (ID_ZAK, COP_ZAK, TEL_ZAK, EMAIL_ZAK, FIRMA) VALUES ('0000000002', '000999222', '722833192', 'Alois.Omacka@seznam.cz', 'N')
```

```
INSERT INTO "VITJ07"."ZAKAZNIK" (ID_ZAK, COP_ZAK, TEL_ZAK, EMAIL_ZAK, FIRMA, NAZEV_FIRMA)
VALUES ('0000000003', '129865342', '606866161', 'andreasKehrer@kft.cz', 'A', 'KFT')
```

```
INSERT INTO "VITJ07"."ZAKAZNIK" (ID_ZAK, COP_ZAK, TEL_ZAK, EMAIL_ZAK, FIRMA) VALUES
('0000000004', '777666444', '783645234', 'Petr.Belohlavy@gmail.com', 'N')
```

```
INSERT INTO "VITJ07"."ZAKAZNIK" (ID_ZAK, COP_ZAK, TEL_ZAK, EMAIL_ZAK, FIRMA, NAZEV_FIRMA)
VALUES ('0000000005', '634523463', '887955782', 'PetrBazo@komercpoj.cz', 'A', 'Komerční
pojišťovna')
```

## ZAMESTNANEC

```
INSERT INTO "VITJ07"."ZAMESTNANEC" (ID_ZAM, ID_POBOCKA, JMENO, PRIJMENI, COP_ZAM,
EMAIL_ZAM, TEL_ZAM, ZAMESTNANY) VALUES ('0000000001', '0000000001', 'Jan', 'Vitu',
'201818233', 'vitj07@vse.cz', '721892822', 'A')
```

```
INSERT INTO "VITJ07"."ZAMESTNANEC" (ID_ZAM, ID_POBOCKA, JMENO, PRIJMENI, COP_ZAM,
EMAIL_ZAM, TEL_ZAM, ZAMESTNANY) VALUES ('0000000002', '0000000001', 'Petr', 'Habásko',
'982887343', 'habasko@auto.cz', '788933421', 'A')
```

```
INSERT INTO "VITJ07"."ZAMESTNANEC" (ID_ZAM, ID_POBOCKA, JMENO, PRIJMENI, COP_ZAM,
EMAIL_ZAM, TEL_ZAM, ZAMESTNANY) VALUES ('0000000003', '0000000002', 'Yurij', 'Vovchuk',
'776283998', 'yurij@auto.cz', '990823745', 'A')
```

```
INSERT INTO "VITJ07"."ZAMESTNANEC" (ID_ZAM, ID_POBOCKA, JMENO, PRIJMENI, COP_ZAM,
EMAIL_ZAM, TEL_ZAM, ZAMESTNANY) VALUES ('0000000004', '0000000003', 'Andrej',
'Nejmenovaný', '223872891', 'andrje@sautbo.cz', '111111111', 'A')
```

```
INSERT INTO "VITJ07"."ZAMESTNANEC" (ID_ZAM, ID_POBOCKA, JMENO, PRIJMENI, COP_ZAM,
EMAIL_ZAM, TEL_ZAM, ZAMESTNANY) VALUES ('0000000005', '0000000004', 'Tomáš', 'Hrančík',
'992384111', 'TomHrancik@auto.cz', '878263454', 'A')
```

## ZAKAZKA

```
INSERT INTO "VITJ07"."ZAKAZKA" (ID_ZAKAZKA, ID_POBOCKA, ID_ZAK, DOB_TRV, DAT_PRIJETI,
PRACE_TECHNIKA_KC) VALUES ('0000000001', '0000000002', '0000000003', '10', TO_DATE('2013-04-
01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), '3000')
```

```
INSERT INTO "VITJ07"."ZAKAZKA" (ID_ZAKAZKA, ID_POBOCKA, ID_ZAK, DOB_TRV, DAT_PRIJETI,
PRACE_TECHNIKA_KC) VALUES ('0000000002', '0000000001', '0000000001', '2', TO_DATE('2013-03-
15 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), '800')
```

```
INSERT INTO "VITJ07"."ZAKAZKA" (ID_ZAKAZKA, ID_POBOCKA, ID_ZAK, DOB_TRV, DAT_PRIJETI,
PRACE_TECHNIKA_KC) VALUES ('0000000003', '0000000002', '0000000002', '4', TO_DATE('2013-03-
19 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), '1200')
```

```
INSERT INTO "VITJ07"."ZAKAZKA" (ID_ZAKAZKA, ID_POBOCKA, ID_ZAK, DOB_TRV, DAT_PRIJETI,
PRACE_TECHNIKA_KC) VALUES ('0000000004', '0000000004', '0000000005', '8', TO_DATE('2013-03-
21 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), '2400')
```

```
INSERT INTO "VITJ07"."ZAKAZKA" (ID_ZAKAZKA, ID_POBOCKA, ID_ZAK, DOB_TRV, DAT_PRIJETI,
PRACE_TECHNIKA_KC) VALUES ('0000000005', '0000000003', '0000000004', '20', TO_DATE('2013-05-
15 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), '6000')
```

### POSKYTOVANE\_SLUZBY

```
INSERT INTO "VITJ07"."POSKYTOVANE_SLUZBY" (ID_ZAKAZKA, ID_SLUZBA) VALUES ('0000000001',
'0000000005')
```

```
INSERT INTO "VITJ07"."POSKYTOVANE_SLUZBY" (ID_ZAKAZKA, ID_SLUZBA) VALUES ('0000000002',
'0000000001')
```

```
INSERT INTO "VITJ07"."POSKYTOVANE_SLUZBY" (ID_ZAKAZKA, ID_SLUZBA) VALUES ('0000000003',
'0000000002')
```

```
INSERT INTO "VITJ07"."POSKYTOVANE_SLUZBY" (ID_ZAKAZKA, ID_SLUZBA) VALUES ('0000000004',
'0000000004')
```

```
INSERT INTO "VITJ07"."POSKYTOVANE_SLUZBY" (ID_ZAKAZKA, ID_SLUZBA) VALUES ('0000000005',
'0000000003')
```

## 5.2 Opis vložených dat

### POBOCKA

ID_POBOCKA	CP	ULICE	MESTO	TEL_POB	EMAIL_POB	ZRUSENA
0000000000	0	NULTA	NULTA	0000000000	NULTA	N
0000000001	887	Šumavská	Kaplice	727833929	kapliceServis@auto.cz	N
0000000002	955	Chemická	Praha	981837224	kunraticeServis@auto.cz	N
0000000003	114	Lanova	České Budějovice	324345189	ceskebudejoviceServis@auto.cz	N
0000000004	1651	Budějovická	Písek	887364220	pisekServis@auto.cz	N

### ZAMESTNANEC

ID_ZAM	ID_POBOCKA	JMENO	PRIJMENI	COP_ZAM	EMAIL_ZAM	TEL_ZAM	ZAMESTNANY
0000000001	0000000001	Jan	Vitu	201818233	vitj07@vse.cz	721892822	A
0000000002	0000000001	Petr	Habásko	982887343	habasko@auto.cz	788933421	A
0000000003	0000000002	Yurij	Vovchuk	776283998	yurij@auto.cz	990823745	A
0000000004	0000000003	Andrej	Nejmenovaný	223872891	andrje@sautbo.cz	111111111	A
0000000005	0000000004	Tomáš	Hrančík	992384111	TomHrancik@auto.cz	878263454	A

### ZAKAZNIK

ID_ZAK	COP_ZAK	TEL_ZAK	EMAIL_ZAK	FIRMA	NAZEV_FIRMA
0000000001	827346521	896532182	JosefNovak@seznam.cz	N	
0000000002	000999222	722833192	Alois.Omacka@seznam.cz	N	
0000000003	129865342	606866161	andreasKehrer@kft.cz	A	KFT
0000000004	777666444	783645234	Petr.Belohlavy@gmail.com	N	

0000000005	634523463	887955782	PetrBazo@komercpoj.cz	A	Komerční pojišťovna
------------	-----------	-----------	-----------------------	---	------------------------

## SLUZBA

ID_SLUZBA	CENA_SLUZBA_KC	POPIS_SLUZBA	VYRAZENO
0000000001	1000	Výměna oleje	N
0000000002	3000	Výměna brzdového kotouče a brzdových destiček	N
0000000003	15000	Oprava zadřeného motoru	N
0000000004	3500	Servisní kontrola	N
0000000005	700	Přezutí pneumatik	N

## ZAKAZKA

ID_ZAKAZKA	ID_POBOCKA	ID_ZAK	DOB_TRV	DAT_PRIJETI	PRACE_TECHNIKA_KC
0000000001	0000000002	0000000003	10	01.04.13	3000
0000000002	0000000001	0000000001	2	15.03.13	800
0000000003	0000000002	0000000002	4	19.03.13	1200
0000000004	0000000004	0000000005	8	21.03.13	2400
0000000005	0000000003	0000000004	20	15.05.13	6000

## POSKYTOVANE\_SLUZBY

ID_ZAKAZKA	ID_SLUZBA
0000000001	0000000005
0000000002	0000000001
0000000003	0000000002
0000000004	0000000004
0000000005	0000000003