

Datový model dle dané předlohy

Zpracovává Michal Dvořák.

Popis dané výšeče světa – zadání

Podle zadání jsem vypracoval datový model, který se zaměřuje na určitou výšeč světa. Vyplynulo z něj, že je zapotřebí připravit takový model, který bude zohledňovat vše, co by mohlo být potřebné pro vygenerování účtenky z restaurace Kolkovna. Účtenka obsahuje různé náležitosti a informace, které souvisejí jednak spolu navzájem a jednak také s informacemi, které na ní sice uvedeny nejsou, ale pro její vytvoření jsou potřebné a hrají svou roli na pozadí.

Protože se jedná o typ restaurace, která má více poboček či funguje jako franšíza, je zde tabulka **Pobočka**, ve které je pobočka evidována a jsou zde uvedeny adresní informace. Nejsou to však jediné informace, dále určitě potřebujeme evidovat mnohé další údaje specifické dané konkrétní pobočce, jako např. kontaktní údaje, proto tu jsou tabulky **Údaj** a **Typ údaje**.

Pokladna je další tabulka, se kterou je vhodné počítat, protože ve větších provozech je určitě využíváno více pokladen. Napojení na objednávku je potřebné vzhledem k tomu, že by se konkrétní objednávka zpracovávala právě na dané pokladně.

V rámci provozu restaurace je určitě dnes potřebné mít zorganizována dobře místa na sezení u různých stolů. Informace, kde zákazník seděl se svým pokrmem, je také uvedena na účtence. Z toho důvodu máme v modelu tabulku **Posezení zákazníků**, pro níž platí, že jedno posezení (stůl) se nachází na jedné konkrétní pobočce a pobočka rozhodně má více možností posezení, proto ta vazba od pobočky k posezení. Místo posezení tedy můžeme evidovat pro každou objednávku, ale nemusíme, pokud budeme předpokládat, že zákazník může využít takeaway.

Zaměstnanec evidujeme ve stejnojmenné tabulce, na níž je zde propojení s tabulkou **Pozice**. Touto vazbou znázorňuji, že zaměstnanec může v našem systému zastávat více pozic a zároveň stejná pozice může být využita pro více zaměstnanců. Dále propojujeme zaměstnance na danou **Objednávku** či více objednávek, za které je zodpovědný, to je nepochybně potřebné, navíc tato informace byla rovněž uvedena na účtence.

Pro objednávku dále evidujeme **Typ zákazníka**, který je pro ní přesně daný a není možné jinak. Díky tomuto typu jsme schopni ukládat informaci, jaký zákazník přišel, je například možné evidovat, zdali využije takeaway nebo si vybere místo apod.

Podstatné napojení na tabulku objednávka je s tabulkou **Položka objednávky**, v níž evidujeme jednotlivá objednaná jídla či pití, která ve výsledku pak tvoří i informace uvedené v účtence, protože ta je vlastně takovým výsledkem objednávky. Položek, které tvoří objednávku, může být samozřejmě mnoho, ale také může být tato objednávka čerstvě evidovaná nebo později zrušená, proto lze počítat i s variantou, že položky nemáme u objednávky žádné.

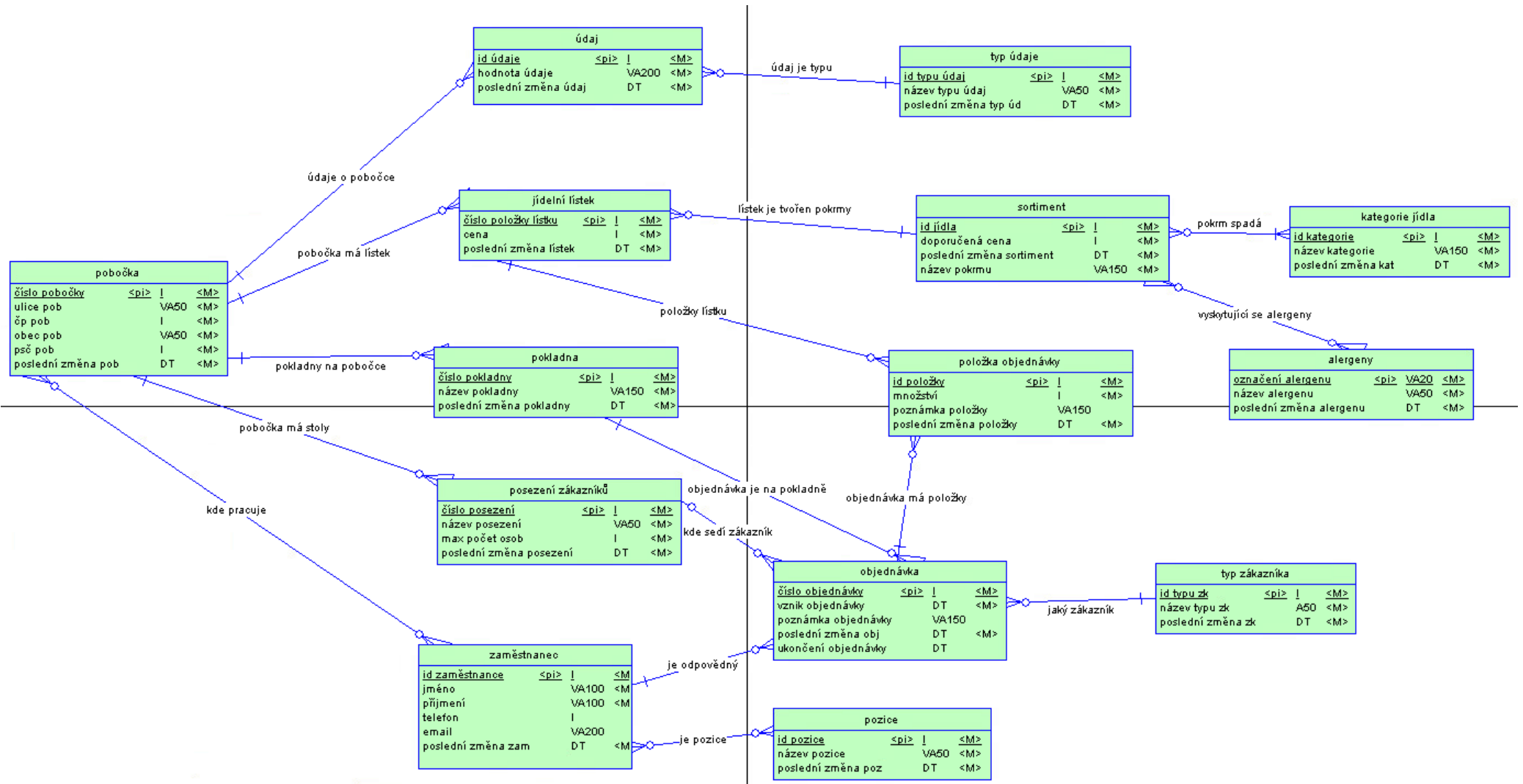
Položka objednávky vychází z **Jídelního lístku** a stejnojmenné tabulky, kde shromažďujeme nabídku jednotlivých poboček, kterou aktuálně mají. Položka tedy odpovídá právě jednomu záznamu, který se nachází v jídelním lístku. Jídelní lístek je ale takovou množinou, ze které zákazníci vybírají, proto jeden záznam z tabulky jídelní lístek může být napojen na mnoho položek mnoha objednávek.

Jídelní lístek je ale konkrétní nabídkou daných poboček, my však počítáme s tím, že se jedná o franšizu či podobně, takže shromažďujeme veškerá jídla, ze kterých by dle daných pravidel měly pobočky vybírat, v tabulce **Sortiment**. Ve výsledku pokrm evidovaný v tabulce sortiment může být dostupný na mnoha pobočkách, které tvoří síť, také ale nemusí být zrovna nikde v nabídce.

Pro restauraci je přínosem mít dobře pokrmy roztříděné, pro tento účel zde slouží tabulka **Kategorie jídla**, pro níž samozřejmě platí, že pokrm evidovaný v tabulce sortiment může spadat do více kategorií či zrovna, když ještě není takový záznam evidován, do žádné. Kategorie jídla je ale široký pojem, pod který spadá mnoho pokrmů.

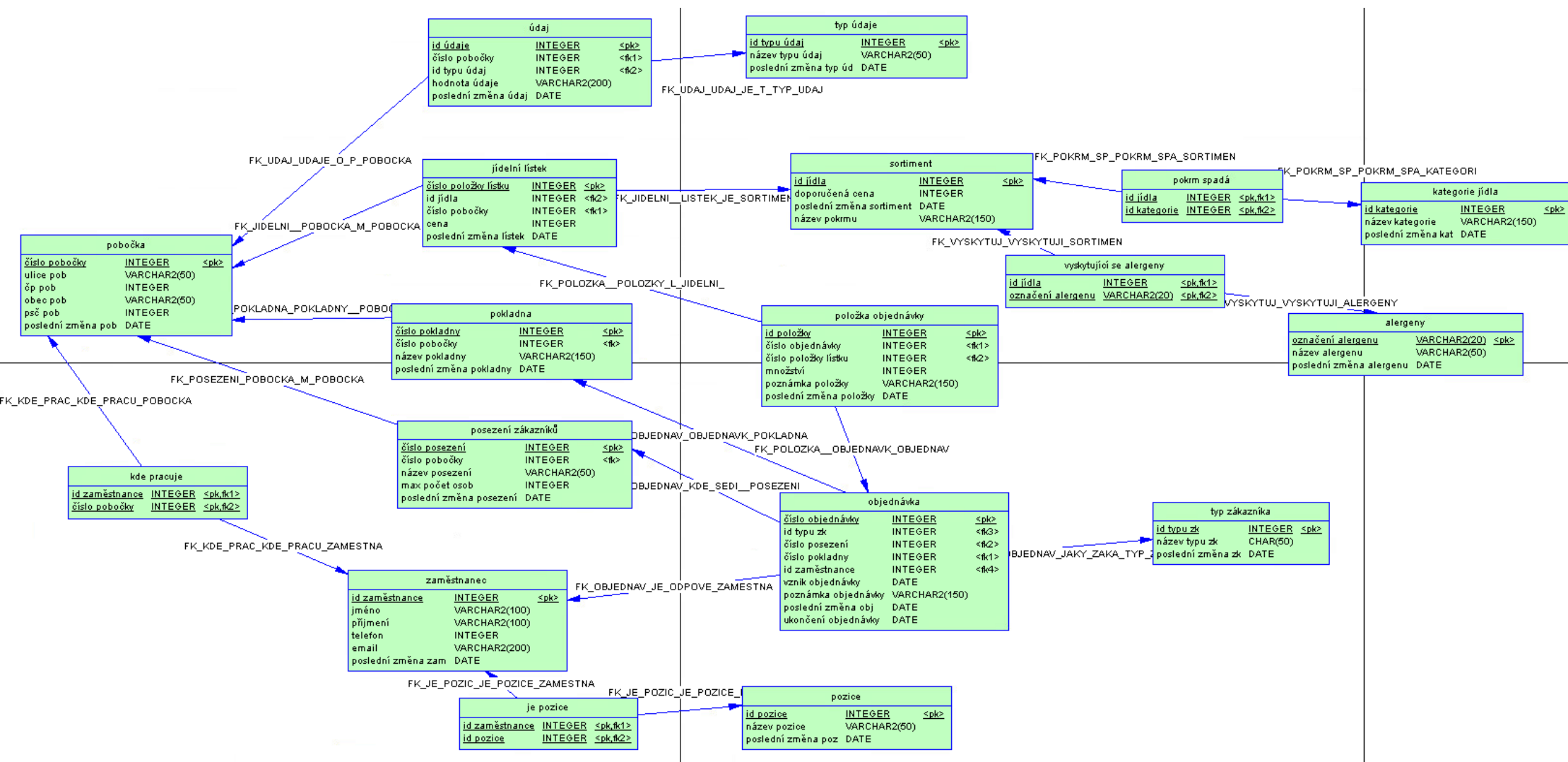
Zajímavostí, ale dnes již nutností, ale pro zákazníky velmi přínosnou a podstatnou informací, zajišťuje tabulka **Alergeny**. Je zapotřebí vědět, co restaurace nabízí, takže je nutné mít systém v tom, jaké alergenys jsou obsaženy v jakých pokrmech, které evidujeme v tabulce sortiment. Vazba mezi těmito dvěma tabulkami znázorňuje situaci, kdy v jednom pokrmu může být mnoho obsažených alergenů, možná také dle oficiálního seznamu žádný, a také alergen se vyskytuje v mnoha různých pokrmech, které se z potravin jeho obsahujících připravují.

Konceptuální datový model



Plnohodnotné zobrazení [zde](#).

Fyzický datový model



Plnohodnotné zobrazení [zde](#).

Definice relačních tabulek a souvisejících objektů

```
/*=====*/
/* DBMS name:      ORACLE Version 10g      */
/* Created on:     15.07.2022 13:11:04      */
/*=====*/

/*=====*/
/* Table: ALERGENY                          */
/*=====*/
create table ALERGENY (
  OZNACENI_ALERGENU   VARCHAR2(20)          not null,
  NAZEV_ALERGENU      VARCHAR2(50)          not null,
  POSLEDNI_ZMENA_ALERGENU DATE              not null,
  constraint PK_ALERGENY primary key (OZNACENI_ALERGENU)
);

/*=====*/
/* Table: JE_POZICE                          */
/*=====*/
create table JE_POZICE (
  ID_ZAMESTNANCE      INTEGER              not null,
  ID_POZICE           INTEGER              not null,
  constraint PK_JE_POZICE primary key (ID_ZAMESTNANCE, ID_POZICE)
);

/*=====*/
/* Index: JE_POZICE_FK                      */
/*=====*/
create index JE_POZICE_FK on JE_POZICE (
```

```

        ID_ZAMESTNANCE ASC
    );

/*=====*/
/* Index: JE_POZICE2_FK */
/*=====*/
create index JE_POZICE2_FK on JE_POZICE (
    ID_POZICE ASC
);

/*=====*/
/* Table: JIDELNI_LISTEK */
/*=====*/
create table JIDELNI_LISTEK (
    CISLO_POLOZKY_LISTKU INTEGER not null,
    ID_JIDLA             INTEGER not null,
    CISLO_POBOCKY        INTEGER not null,
    CENA                 INTEGER not null,
    POSLEDNI_ZMENA_LISTEK DATE not null,
    constraint PK_JIDELNI_LISTEK primary key (CISLO_POLOZKY_LISTKU)
);

/*=====*/
/* Index: POBOCKA_MA_LISTEK_FK */
/*=====*/
create index POBOCKA_MA_LISTEK_FK on JIDELNI_LISTEK (
    CISLO_POBOCKY ASC
);

/*=====*/
/* Index: LISTEK_JE_TVOREN_POKRMY_FK */
/*=====*/

```

```

/*=====*/
create index LISTEK_JE_TVOREN_POKRMY_FK on JIDELNI_LISTEK (
    ID_JIDLA ASC
);

/*=====*/
/* Table: KATEGORIE_JIDLA */
/*=====*/
create table KATEGORIE_JIDLA (
    ID_KATEGORIE          INTEGER          not null,
    NAZEV_KATEGORIE       VARCHAR2(150)    not null,
    POSLEDNI_ZMENA_KAT    DATE             not null,
    constraint PK_KATEGORIE_JIDLA primary key (ID_KATEGORIE)
);

/*=====*/
/* Table: KDE_PRACUJE */
/*=====*/
create table KDE_PRACUJE (
    ID_ZAMESTNANCE        INTEGER          not null,
    CISLO_POBOCKY         INTEGER          not null,
    constraint PK_KDE_PRACUJE primary key (ID_ZAMESTNANCE, CISLO_POBOCKY)
);

/*=====*/
/* Index: KDE_PRACUJE_FK */
/*=====*/
create index KDE_PRACUJE_FK on KDE_PRACUJE (
    ID_ZAMESTNANCE ASC
);

```



```

/*=====*/
/* Index: KDE_PRACUJE2_FK */
/*=====*/
create index KDE_PRACUJE2_FK on KDE_PRACUJE (
    CISLO_POBOCKY ASC
);

/*=====*/
/* Table: OBJEDNAVKA */
/*=====*/
create table OBJEDNAVKA (
    CISLO_OBJEDNAVKY      INTEGER          not null,
    ID_TYPU_ZK            INTEGER          not null,
    CISLO_POSEZENI        INTEGER,
    CISLO_POKLADNY        INTEGER          not null,
    ID_ZAMESTNANCE        INTEGER          not null,
    VZNIK_OBJEDNAVKY      DATE             not null,
    POZNAMKA_OBJEDNAVKY   VARCHAR2(150),
    POSLEDNI_ZMENA_OBJ    DATE             not null,
    UKONCENI_OBJEDNAVKY   DATE,
    constraint PK_OBJEDNAVKA primary key (CISLO_OBJEDNAVKY)
);

/*=====*/
/* Index: OBJEDNAVKA_JE_NA_POKLADNE_FK */
/*=====*/
create index OBJEDNAVKA_JE_NA_POKLADNE_FK on OBJEDNAVKA (
    CISLO_POKLADNY ASC
);

/*=====*/

```

```

/* Index: KDE_SEDI_ZAKAZNIK_FK */
/*=====*/
create index KDE_SEDI_ZAKAZNIK_FK on OBJEDNAVKA (
    CISLO_POSEZENI ASC
);

/*=====*/
/* Index: JAKY_ZAKAZNIK_FK */
/*=====*/
create index JAKY_ZAKAZNIK_FK on OBJEDNAVKA (
    ID_TYPU_ZK ASC
);

/*=====*/
/* Index: JE_ODPOVEDNY_FK */
/*=====*/
create index JE_ODPOVEDNY_FK on OBJEDNAVKA (
    ID_ZAMESTNANCE ASC
);

/*=====*/
/* Table: POBOCKA */
/*=====*/
create table POBOCKA (
    CISLO_POBOCKY          INTEGER          not null,
    ULICE_POB              VARCHAR2(50)      not null,
    CP_POB                 INTEGER          not null,
    OBEC_POB               VARCHAR2(50)      not null,
    PSC_POB                INTEGER          not null,
    POSLEDNI_ZMENA_POB     DATE              not null,
    constraint PK_POBOCKA primary key (CISLO_POBOCKY)

```

```

);

/*=====*/
/* Table: POKLADNA */
/*=====*/
create table POKLADNA (
    CISLO_POKLADNY      INTEGER          not null,
    CISLO_POBOCKY       INTEGER          not null,
    NAZEV_POKLADNY      VARCHAR2(150)    not null,
    POSLEDNI_ZMENA_POKLADNY DATE          not null,
    constraint PK_POKLADNA primary key (CISLO_POKLADNY)
);

/*=====*/
/* Index: POKLADNY_NA_POBOCCE_FK */
/*=====*/
create index POKLADNY_NA_POBOCCE_FK on POKLADNA (
    CISLO_POBOCKY ASC
);

/*=====*/
/* Table: POKRM_SPADA */
/*=====*/
create table POKRM_SPADA (
    ID_JIDLA            INTEGER          not null,
    ID_KATEGORIE        INTEGER          not null,
    constraint PK_POKRM_SPADA primary key (ID_JIDLA, ID_KATEGORIE)
);

/*=====*/
/* Index: POKRM_SPADA_FK */

```

```

/*=====*/
create index POKRM_SPADA_FK on POKRM_SPADA (
    ID_JIDLA ASC
);

/*=====*/
/* Index: POKRM_SPADA2_FK */
/*=====*/
create index POKRM_SPADA2_FK on POKRM_SPADA (
    ID_KATEGORIE ASC
);

/*=====*/
/* Table: POLOZKA_OBJEDNAVKY */
/*=====*/
create table POLOZKA_OBJEDNAVKY (
    ID_POLOZKY          INTEGER          not null,
    CISLO_OBJEDNAVKY    INTEGER          not null,
    CISLO_POLOZKY_LISTKU INTEGER          not null,
    MNOZSTVI           INTEGER          not null,
    POZNAMKA_POLOZKY    VARCHAR2(150),
    POSLEDNI_ZMENA_POLOZKY DATE          not null,
    constraint PK_POLOZKA_OBJEDNAVKY primary key (ID_POLOZKY)
);

/*=====*/
/* Index: OBJEDNAVKA_MA_POLOZKY_FK */
/*=====*/
create index OBJEDNAVKA_MA_POLOZKY_FK on POLOZKA_OBJEDNAVKY (
    CISLO_OBJEDNAVKY ASC
);

```

```

/*=====*/
/* Index: POLOZKY_LISTKU_FK */
/*=====*/
create index POLOZKY_LISTKU_FK on POLOZKA_OBJEDNAVKY (
    CISLO_POLOZKY_LISTKU ASC
);

/*=====*/
/* Table: POSEZENI_ZAKAZNIKU */
/*=====*/
create table POSEZENI_ZAKAZNIKU (
    CISLO_POSEZENI      INTEGER          not null,
    CISLO_POBOCKY       INTEGER          not null,
    NAZEV_POSEZENI      VARCHAR2(50)    not null,
    MAX_POCET_OSOB      INTEGER          not null,
    POSLEDNI_ZMENA_POSEZENI DATE          not null,
    constraint PK_POSEZENI_ZAKAZNIKU primary key (CISLO_POSEZENI)
);

/*=====*/
/* Index: POBOCKA_MA_STOLY_FK */
/*=====*/
create index POBOCKA_MA_STOLY_FK on POSEZENI_ZAKAZNIKU (
    CISLO_POBOCKY ASC
);

/*=====*/
/* Table: POZICE */
/*=====*/
create table POZICE (

```

```

ID_POZICE          INTEGER          not null,
NAZEV_POZICE       VARCHAR2(50)     not null,
POSLEDNI_ZMENA_POZ DATE             not null,
constraint PK_POZICE primary key (ID_POZICE)
);

/*=====*/
/* Table: SORTIMENT */
/*=====*/
create table SORTIMENT (
  ID_JIDLA          INTEGER          not null,
  DOPORUCENA_CENA   INTEGER          not null,
  POSLEDNI_ZMENA_SORTIMENT DATE      not null,
  NAZEV_POKRMU      VARCHAR2(150)   not null,
  constraint PK_SORTIMENT primary key (ID_JIDLA)
);

/*=====*/
/* Table: TYP_UDAJE */
/*=====*/
create table TYP_UDAJE (
  ID_TYPU_UDAJ      INTEGER          not null,
  NAZEV_TYPU_UDAJ   VARCHAR2(50)     not null,
  POSLEDNI_ZMENA_TYP_UD DATE        not null,
  constraint PK_TYP_UDAJE primary key (ID_TYPU_UDAJ)
);

/*=====*/
/* Table: TYP_ZAKAZNIKA */
/*=====*/
create table TYP_ZAKAZNIKA (

```

```

ID_TYPU_ZK          INTEGER          not null,
NAZEV_TYPU_ZK       CHAR(50)         not null,
POSLEDNI_ZMENA_ZK   DATE             not null,
constraint PK_TYP_ZAKAZNIKA primary key (ID_TYPU_ZK)
);

/*=====*/
/* Table: UDAJ */
/*=====*/
create table UDAJ (
    ID_UDAJE          INTEGER          not null,
    CISLO_POBOCKY     INTEGER          not null,
    ID_TYPU_UDAJ       INTEGER          not null,
    HODNOTA_UDAJE      VARCHAR2(200)   not null,
    POSLEDNI_ZMENA_UDAJ DATE           not null,
    constraint PK_UDAJ primary key (ID_UDAJE)
);

/*=====*/
/* Index: UDAJE_O_POBOCCE_FK */
/*=====*/
create index UDAJE_O_POBOCCE_FK on UDAJ (
    CISLO_POBOCKY ASC
);

/*=====*/
/* Index: UDAJ_JE_TYPU_FK */
/*=====*/
create index UDAJ_JE_TYPU_FK on UDAJ (
    ID_TYPU_UDAJ ASC
);

```

```

/*=====*/
/* Table: VYSKYTUJICI_SE_ALERGENY */
/*=====*/
create table VYSKYTUJICI_SE_ALERGENY (
    ID_JIDLA            INTEGER                not null,
    OZNACENI_ALERGENU   VARCHAR2(20)          not null,
    constraint PK_VYSKYTUJICI_SE_ALERGENY primary key (ID_JIDLA, OZNACENI_ALERGENU)
);

/*=====*/
/* Index: VYSKYTUJICI_SE_ALERGENY_FK */
/*=====*/
create index VYSKYTUJICI_SE_ALERGENY_FK on VYSKYTUJICI_SE_ALERGENY (
    ID_JIDLA ASC
);

/*=====*/
/* Index: VYSKYTUJICI_SE_ALERGENY2_FK */
/*=====*/
create index VYSKYTUJICI_SE_ALERGENY2_FK on VYSKYTUJICI_SE_ALERGENY (
    OZNACENI_ALERGENU ASC
);

/*=====*/
/* Table: ZAMESTNANEC */
/*=====*/
create table ZAMESTNANEC (
    ID_ZAMESTNANCE      INTEGER                not null,
    JMENO               VARCHAR2(100)          not null,
    PRIJMENI            VARCHAR2(100)          not null,

```



```

TELEFON            INTEGER,
EMAIL              VARCHAR2(200),
POSLEDNI_ZMENA_ZAM DATE                        not null,
constraint PK_ZAMESTNANEC primary key (ID_ZAMESTNANCE)
);

alter table JE_POZICE
  add constraint FK_JE_POZIC_JE_POZICE_ZAMESTNA foreign key (ID_ZAMESTNANCE)
    references ZAMESTNANEC (ID_ZAMESTNANCE)
ON DELETE CASCADE;

alter table JE_POZICE
  add constraint FK_JE_POZIC_JE_POZICE_POZICE foreign key (ID_POZICE)
    references POZICE (ID_POZICE)
ON DELETE CASCADE;

alter table JIDELNI_LISTEK
  add constraint FK_JIDELNI__LISTEK_JE_SORTIMEN foreign key (ID_JIDLA)
    references SORTIMENT (ID_JIDLA)
ON DELETE CASCADE;

alter table JIDELNI_LISTEK
  add constraint FK_JIDELNI__POBOCKA_M_POBOCKA foreign key (CISLO_POBOCKY)
    references POBOCKA (CISLO_POBOCKY)
ON DELETE CASCADE;

alter table KDE_PRACUJE
  add constraint FK_KDE_PRAC_KDE_PRACU_ZAMESTNA foreign key (ID_ZAMESTNANCE)
    references ZAMESTNANEC (ID_ZAMESTNANCE)
ON DELETE CASCADE;

```

```
alter table KDE_PRACUJE
  add constraint FK_KDE_PRAC_KDE_PRACU_POBOCKA foreign key (CISLO_POBOCKY)
    references POBOCKA (CISLO_POBOCKY)
ON DELETE CASCADE;

alter table OBJEDNAVKA
  add constraint FK_OBJEDNAV_JAKY_ZAKA_TYP_ZAKA foreign key (ID_TYPU_ZK)
    references TYP_ZAKAZNIKA (ID_TYPU_ZK)
ON DELETE SET NULL;

alter table OBJEDNAVKA
  add constraint FK_OBJEDNAV_JE_ODPOVE_ZAMESTNA foreign key (ID_ZAMESTNANCE)
    references ZAMESTNANEC (ID_ZAMESTNANCE);

alter table OBJEDNAVKA
  add constraint FK_OBJEDNAV_KDE_SEDI_POSEZENI foreign key (CISLO_POSEZENI)
    references POSEZENI_ZAKAZNIKU (CISLO_POSEZENI)
ON DELETE SET NULL;

alter table OBJEDNAVKA
  add constraint FK_OBJEDNAV_OBJEDNAVK_POKLADNA foreign key (CISLO_POKLADNY)
    references POKLADNA (CISLO_POKLADNY)
ON DELETE SET NULL;

alter table POKLADNA
  add constraint FK_POKLADNA_POKLADNY_POBOCKA foreign key (CISLO_POBOCKY)
    references POBOCKA (CISLO_POBOCKY)
ON DELETE CASCADE;

alter table POKRM_SPADA
  add constraint FK_POKRM_SP_POKRM_SPA_SORTIMEN foreign key (ID_JIDLA)
```

```

        references SORTIMENT (ID_JIDLA)
ON DELETE CASCADE;

alter table POKRM_SPADA
    add constraint FK_POKRM_SP_POKRM_SPA_KATEGORI foreign key (ID_KATEGORIE)
        references KATEGORIE_JIDLA (ID_KATEGORIE)
ON DELETE CASCADE;

alter table POLOZKA_OBJEDNAVKY
    add constraint FK_POLOZKA__OBJEDNAVK_OBJEDNAV foreign key (CISLO_OBJEDNAVKY)
        references OBJEDNAVKA (CISLO_OBJEDNAVKY)
ON DELETE CASCADE;

alter table POLOZKA_OBJEDNAVKY
    add constraint FK_POLOZKA__POLOZKY_L_JIDELNI_ foreign key (CISLO_POLOZKY_LISTKU)
        references JIDELNI_LISTEK (CISLO_POLOZKY_LISTKU);

alter table POSEZENI_ZAKAZNIKU
    add constraint FK_POSEZENI_POBOCKA_M_POBOCKA foreign key (CISLO_POBOCKY)
        references POBOCKA (CISLO_POBOCKY)
ON DELETE CASCADE;

alter table UDAJ
    add constraint FK_UDAJ_UDAJE_O_P_POBOCKA foreign key (CISLO_POBOCKY)
        references POBOCKA (CISLO_POBOCKY)
ON DELETE CASCADE;

alter table UDAJ
    add constraint FK_UDAJ_UDAJ_JE_T_TYP_UDAJ foreign key (ID_TYPU_UDAJ)
        references TYP_UDAJE (ID_TYPU_UDAJ)
ON DELETE CASCADE;

```

```
alter table VYSKYTUJICI_SE_ALERGENY
  add constraint FK_VYSKYTUJ_VYSKYTUJI_SORTIMEN foreign key (ID_JIDLA)
    references SORTIMENT (ID_JIDLA)
ON DELETE CASCADE;

alter table VYSKYTUJICI_SE_ALERGENY
  add constraint FK_VYSKYTUJ_VYSKYTUJI_ALERGENY foreign key (OZNACENI_ALERGENU)
    references ALERGENY (OZNACENI_ALERGENU)
ON DELETE CASCADE;
```