

Dokumentace databáze

Zdeněk Šejvl, Roman Skřivan, Marek Lešetický

Obsah

1. Určení a očekávání databáze

2. Zadavatel/ zákazník

- i. Funkční požadavky
- ii. Testovací data
- iii. Konceptuální model

3. ICT správce firmy

- i. Požadavky na HW, SW
- ii. Postup instalace DB

4. Analytika a vývoj

- i. E-R model
- ii. Integritní omezení, popis jednotlivých tabulek

5. Závěr

1. Určení a očekávání databáze.

- Databáze je určena pro prostředí nemocnice, přesněji pro nemocniční personál.
- Od databáze se očekává efektivita využití, rychlost a jednoduchost.

2.Zadavatel/ Zákazník

i. Funkční požadavky

- Materiál
 - a) Systém eviduje rozdělení materiálu v nemocnici
 - b) Celkový počet jednotlivého materiálu
 - c) Na každé odvěti materiálu je soukromá tabulka, jelikož můžou mít různé atributy
- Léky
 - a) Evidence druhu léku, výrobce léku a samotného léku
 - b) Evidence jakému léku byl podán jaký lék
- Lůžka
 - a) Evidence lůžek, pokojů a oddělení v nemocnici
 - b) Také eviduje jednotlivé druhy lůžek
- Pacienti
 - a) Systém u pacientů se eviduje jejich osobní údaje
 - b) Také se eviduje jaký doktor, se stará o jakého pacienta. To samé se sestřičkami.
- Doktoři
 - a) Systém u doktorů se eviduje jejich osobní údaje
- Sestřičky
 - b) Systém u sestřiček eviduje jejich osobní údaje
- Stravu
 - a) Systém eviduje stravu rozdělenou na snídani, oběd a večeři.
 - b) Pacient má potom složen snídani, oběd a večeří na jeden den

ii. Testovací data

- Testovací data jsou ve formátu MS Excel, zatím nezveřejněna

- ### iii. Konceptuální model



3. ICT správce firmy

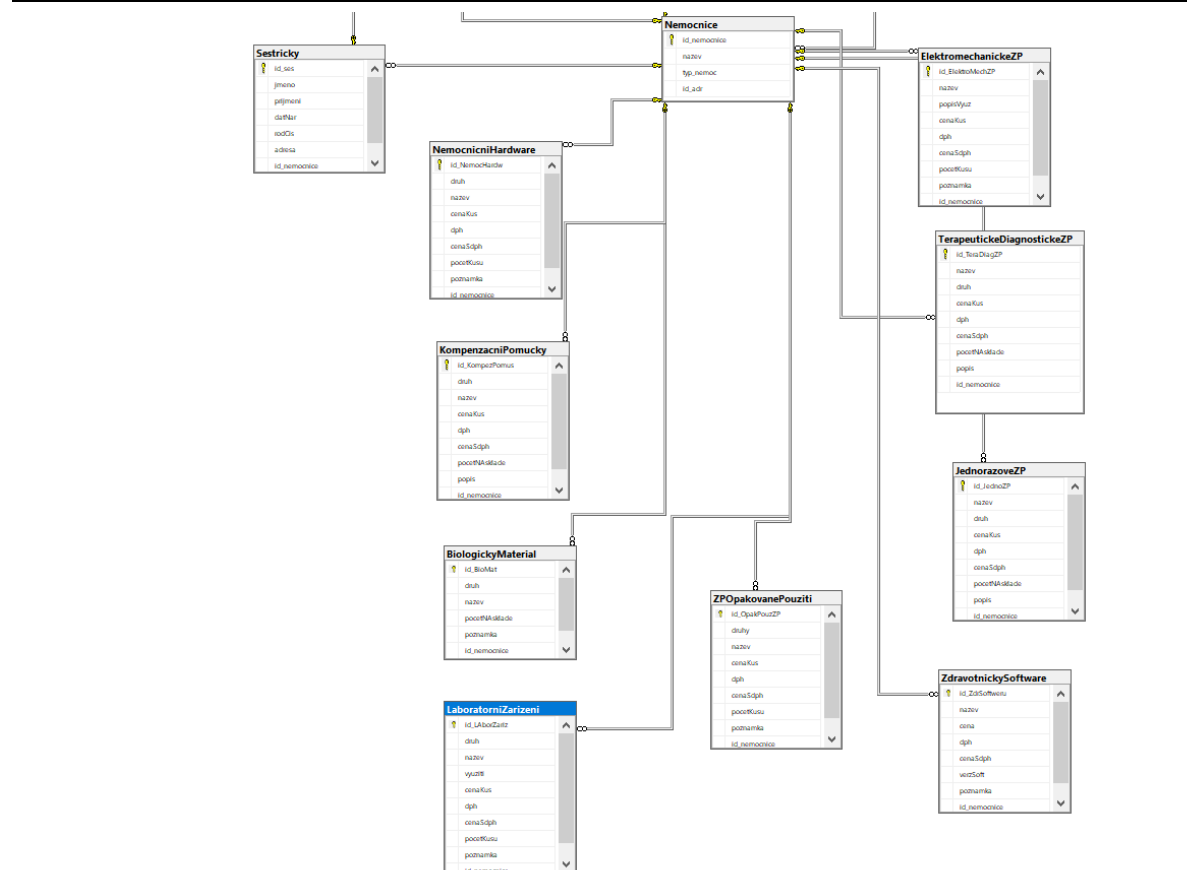
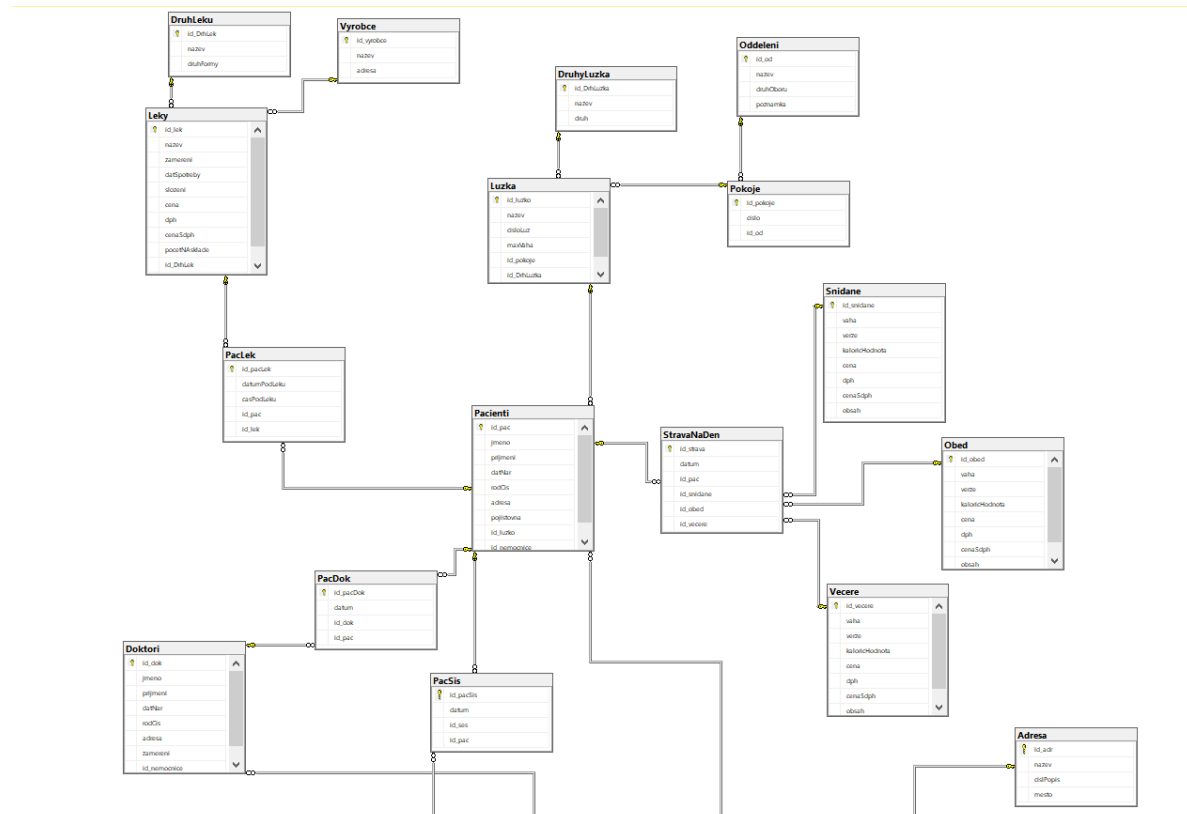
i. Požadavky na HW a SW

- MS SQL Server, minimální verze 2014
- Internetové připojení, minimálně 2Mb/ s
- Diskový prostor, minimálně 20 MB

ii. Postup instalace DB

- Databáze se instaluje jednoduše, zde je krok po kroku:
 - 1) Otevřete kod databáze, který je v příloze databáze/script database.sql, musí být nainstalován MS SQL Server
 - 2) Otevřete MS SQL Server a vytvořte databázi Nemocnice
 - 3) Nakonec stačí dát execute u modu, který jste otevřeli

4. Analytika a vývoj



- i. Er – model
 - Na předchozí stránce
- ii. Integritní omezení, popis jednotlivých tabulek
 - Databáze se nachází v 2. Normálové formě

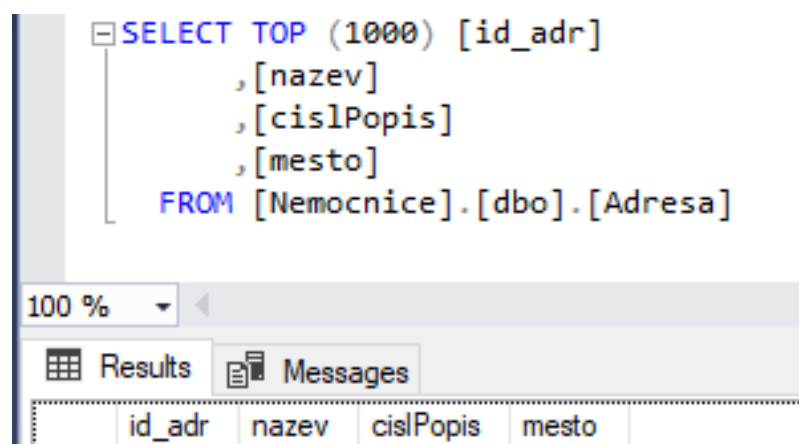
Tabulky:

➤ Adresa

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky adresa */
create table Adresa(
    id_adr int identity(1,1) primary key,
    nazev nvarchar(50) not null,
    cislPopis int not null,
    mesto nvarchar(50) not null
);
```

- Print – screen aplikace tabuoky v MS SQL Serveru



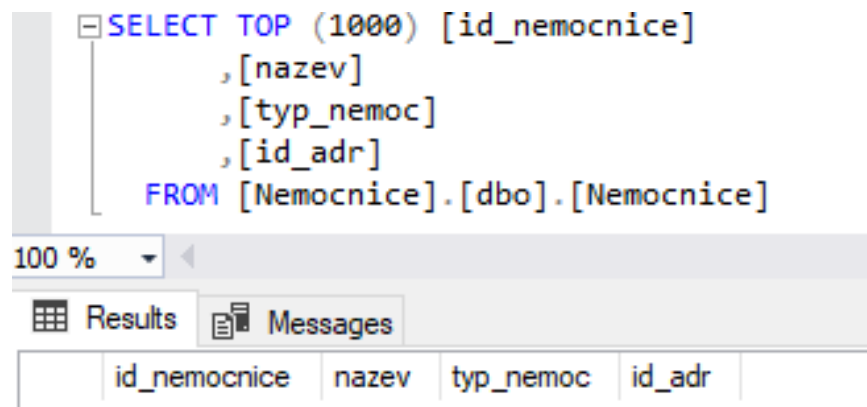
- Do tabulky se zapisuje název adresy, číslo popisné a měst

➤ Nemocnice

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky nemocnice a fk adresy */
create table Nemocnice(
    id_nemocnice int identity(1,1) primary key,
    nazev nvarchar(50) not null,
    typ_nemoc nvarchar(100) not null,
    id_adr int not null,
    constraint fk_nemocnice_adresa foreign key(id_adr) references Adresa(id_adr)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru



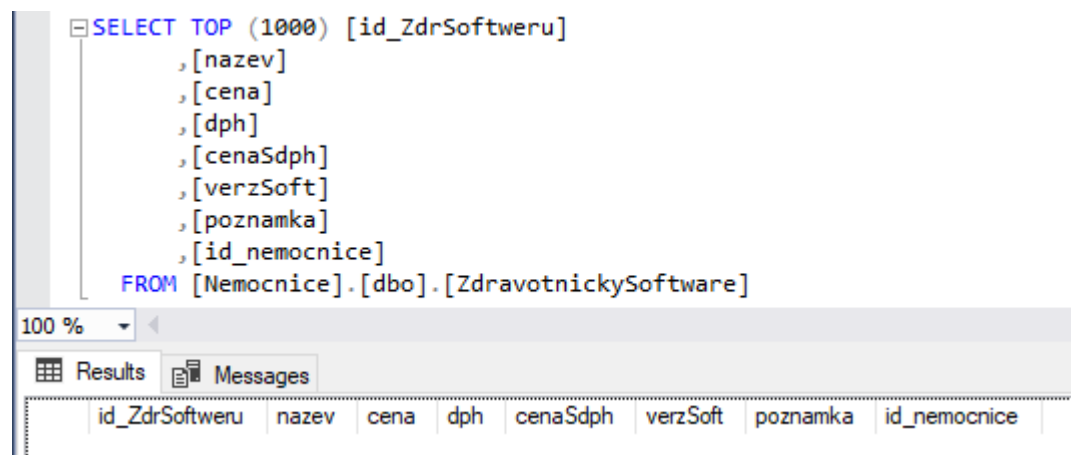
- Do tabulky se zapisuje název nemocnice, typ nemocnice a cizí klíč adresy

➤ ZdravotnickýSoftware

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky ZdravotnickýSoftware a fk nemocnice */
create table ZdravotnickýSoftware(
    id_ZdrSoftweru int identity(1,1) primary key,
    nazev nvarchar(100) not null,
    cena float not null,
    dph nvarchar(10) not null,
    cenaSdph float not null,
    verzSoft nvarchar(12) not null,
    poznamka nvarchar(500) not null,
    id_nemocnice int not null,
    constraint fk_zdravotnickýSoftware_nemocnice foreign key(id_nemocnice) references Nemocnice(id_nemocnice)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru



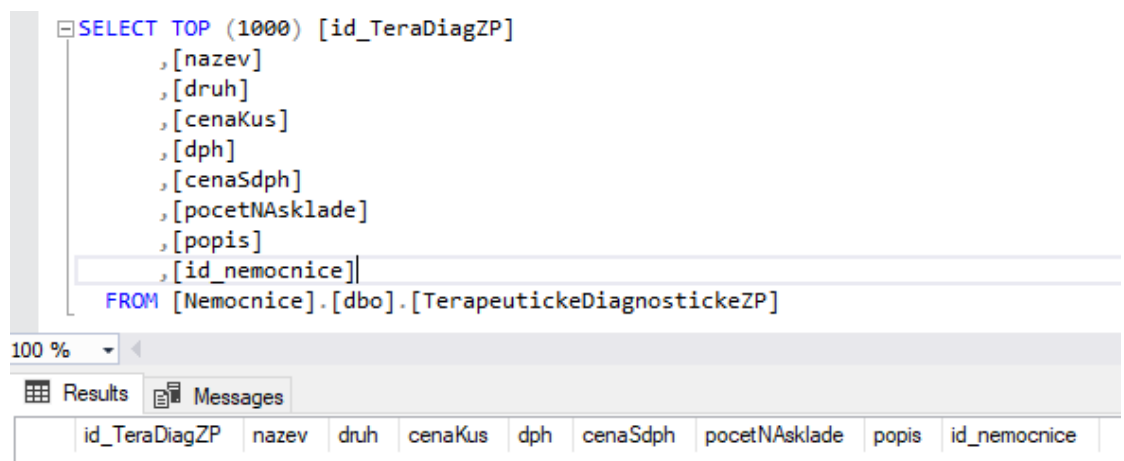
- Do tabulky se zapisuje název softweru, cena, dph, cena s dph, verze softweru, poznámka (není povinná) a cizí klíč nemocnice

➤ TerapeutickéDignostickéZP

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky TerapeutickeDiagnostickeZP a fk nemocnice */
create table TerapeutickeDiagnostickeZP(
    id_TeraDiagZP int identity(1,1) primary key,
    nazev nvarchar(100) not null,
    druh nvarchar(40) not null,
    cenaKus float not null,
    dph nvarchar(10) not null,
    cenaSdph float not null,
    pocetNAsklade int not null,
    popis nvarchar(500) not null,
    id_nemocnice int not null,
    constraint fk_terapeutickeDiagnostickeZP_nemocnice foreign key(id_nemocnice) references Nemocnice(id_nemocnice)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru



The screenshot shows a SQL query window with the following query:

```
SELECT TOP (1000) [id_TeraDiagZP]
, [nazev]
, [druh]
, [cenaKus]
, [dph]
, [cenaSdph]
, [pocetNAsklade]
, [popis]
, [id_nemocnice]
FROM [Nemocnice].[dbo].[TerapeutickeDiagnostickeZP]
```

Below the query, the 'Results' tab is active, displaying a table with the following columns:

id_TeraDiagZP	nazev	druh	cenaKus	dph	cenaSdph	pocetNAsklade	popis	id_nemocnice
---------------	-------	------	---------	-----	----------	---------------	-------	--------------

- Do tabulky se zapisuje název pomůcky, druh pomůcek, cena na kus, dph, cena s dph, počet na skladě, popis (není povinný) a cizí klíč nemocnice

➤ JednorazoveZP

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky JednorazoveZP a fk nemocnice */
create table JednorazoveZP(
    id_JednoZP int identity(1,1) primary key,
    nazev nvarchar(100) not null,
    druh nvarchar(50) not null,
    cenaKus float not null,
    dph nvarchar(10) not null,
    cenaSdph float not null,
    pocetNasklade int not null,
    popis nvarchar(500) not null,
    id_nemocnice int not null,
    constraint fk_jednorazoveZP_nemocnice foreign key(id_nemocnice) references Nemocnice(id_nemocnice)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_JednoZP]
, [nazev]
, [druh]
, [cenaKus]
, [dph]
, [cenaSdph]
, [pocetNasklade]
, [popis]
, [id_nemocnice]
FROM [Nemocnice].[dbo].[JednorazoveZP]
```

100 %

Results

Messages

id_JednoZP	nazev	druh	cenaKus	dph	cenaSdph	pocetNasklade	popis	id_nemocnice
------------	-------	------	---------	-----	----------	---------------	-------	--------------

- Do tabulky se zapisuje název pomůcky, druh pomůcek, cena na kus, dph, cena s dph, počet na skladě, popis (není povinný) a cizí klíč nemocnice

➤ BiologickyMaterial

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky BiologickyMaterial a fk nemocnice */
create table BiologickyMaterial(
    id_BioMat int identity(1,1) primary key,
    druh nvarchar(50) not null,
    nazev nvarchar(100) not null,
    pocetNasklade int not null,
    poznamka nvarchar(500) not null,
    id_nemocnice int not null,
    constraint fk_BiologickyMaterial_nemocnice foreign key(id_nemocnice) references Nemocnice(id_nemocnice)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_BioMat]
, [druh]
, [nazev]
, [pocetNasklade]
, [poznamka]
, [id_nemocnice]
FROM [Nemocnice].[dbo].[BiologickyMaterial]
```

00 %

Results Messages

id_BioMat	druh	nazev	pocetNasklade	poznamka	id_nemocnice
-----------	------	-------	---------------	----------	--------------

- Do tabulky se zapisuje druh materiálu, název materiálu, počet kusů na skladě, poznámka (není povinný) a cizí klíč nemocnice

➤ KompezacniPomucky

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky KompenzacniPomucky a fk nemocnice */
create table KompenzacniPomucky(
    id_KompezPomus int identity(1,1) primary key,
    druh nvarchar(50) not null,
    nazev nvarchar(100) not null,
    cenaKus float not null,
    dph nvarchar(10) not null,
    cenaSdph float not null,
    pocetNasklade int not null,
    popis nvarchar(500) not null,
    id_nemocnice int not null,
    constraint fk_kompenzacniPomucky_nemocnice foreign key(id_nemocnice) references Nemocnice(id_nemocnice)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_KompezPomus]
, [druh]
, [nazev]
, [cenaKus]
, [dph]
, [cenaSdph]
, [pocetNasklade]
, [popis]
, [id_nemocnice]
FROM [Nemocnice].[dbo].[KompenzacniPomucky]
```

100 %

Results Messages

id_KompezPomus	druh	nazev	cenaKus	dph	cenaSdph	pocetNasklade	popis	id_nemocnice
----------------	------	-------	---------	-----	----------	---------------	-------	--------------

- Do tabulky se zapisuje druh pomůcky, název pomůcky, cena na kus, dph, cena s dph, počet na skladě, popis (není povinný) a cizí klíč nemocnice

➤ LaboratniZarizeni

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky LaboratniZarizeni a fk nemocnice */
create table LaboratniZarizeni(
    id_LaborZariz int identity(1,1) primary key,
    druh nvarchar(100) not null,
    nazev nvarchar(100) not null,
    vyuziti nvarchar(500) not null,
    cenaKus float not null,
    dph nvarchar(10) not null,
    cenaSdph float not null,
    pocetKusu int not null,
    poznamka nvarchar(500) not null,
    id_nemocnice int not null,
    constraint fk_laboratniZarizeniy_nemocnice foreign key(id_nemocnice) references Nemocnice(id_nemocnice)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_LaborZariz]
, [druh]
, [nazev]
, [vyuziti]
, [cenaKus]
, [dph]
, [cenaSdph]
, [pocetKusu]
, [poznamka]
, [id_nemocnice]
FROM [Nemocnice].[dbo].[LaboratniZarizeni]
```

0 %

Results

Messages

id_LaborZariz	druh	nazev	vyuziti	cenaKus	dph	cenaSdph	pocetKusu	poznamka	id_nemocnice
---------------	------	-------	---------	---------	-----	----------	-----------	----------	--------------

- Do tabulky se zapisuje druh zařízení, název zařízení, využití, cena na kus, dph, cena s dph, počet kusů, poznámka (není povinný) a cizí klíč nemocnice

➤ NemocnicniHardware

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky NemocnicniHardware a fk nemocnice */
create table NemocnicniHardware(
    id_NemocHardw int identity(1,1) primary key,
    druh nvarchar(70) not null,
    nazev nvarchar(100) not null,
    cenaKus float not null,
    dph nvarchar(10) not null,
    cenaSdph float not null,
    pocetKusu int not null,
    poznamka nvarchar(500) not null,
    id_nemocnice int not null,
    constraint fk_nemocnicniHardware_nemocnice foreign key(id_nemocnice) references Nemocnice(id_nemocnice)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_NemocHardw]
, [druh]
, [nazev]
, [cenaKus]
, [dph]
, [cenaSdph]
, [pocetKusu]
, [poznamka]
, [id_nemocnice]
FROM [Nemocnice].[dbo].[NemocnicniHardware]
```

100 %

Results Messages

id_NemocHardw	druh	nazev	cenaKus	dph	cenaSdph	pocetKusu	poznamka	id_nemocnice
---------------	------	-------	---------	-----	----------	-----------	----------	--------------

- Do tabulky se zapisuje druh hardware, název hardwaru, cena na kus, dph, cena s dph, počet kusů, poznámka (není povinný) a cizí klíč nemocnice

➤ ZPOpakovanehoPouziti

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky ZPOpakovanePouziti a fk nemocnice */
create table ZPOpakovanePouziti(
    id_OpakPouzZP int identity(1,1) primary key,
    druhy nvarchar(100) not null,
    nazev nvarchar(100) not null,
    cenaKus float not null,
    dph nvarchar(10) not null,
    cenaSdph float not null,
    pocetKusu int not null,
    poznamka nvarchar(500) not null,
    id_nemocnice int not null,
    constraint fk_zPOpakovanePouziti_nemocnice foreign key(id_nemocnice) references Nemocnice(id_nemocnice)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_OpakPouzZP]
      ,[druhy]
      ,[nazev]
      ,[cenaKus]
      ,[dph]
      ,[cenaSdph]
      ,[pocetKusu]
      ,[poznamka]
      ,[id_nemocnice]
FROM [Nemocnice].[dbo].[ZPOpakovanePouziti]
```

0 %

Results Messages

id_OpakPouzZP	druhy	nazev	cenaKus	dph	cenaSdph	pocetKusu	poznamka	id_nemocnice
---------------	-------	-------	---------	-----	----------	-----------	----------	--------------

- Do tabulky se zapisuje druh pomůcky, název pomůcky, cena na kus, dph, cena s dph, počet kusů, poznámka (není povinný) a cizí klíč nemocnice

➤ ElektromechanickeZP

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky ElektromechanickeZP a fk nemocnice */
create table ElektromechanickeZP(
    id_ElektroMechZP int identity(1,1) primary key,
    nazev nvarchar(100) not null,
    popisVyuz nvarchar(500) not null,
    cenaKus float not null,
    dph nvarchar(10) not null,
    cenaSdph float not null,
    pocetKusu int not null,
    poznamka nvarchar(500) not null,
    id_nemocnice int not null,
    constraint fk_elektromechanickeZP_nemocnice foreign key(id_nemocnice) references Nemocnice(id_nemocnice)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_ElektroMechZP]
, [nazev]
, [popisVyuz]
, [cenaKus]
, [dph]
, [cenaSdph]
, [pocetKusu]
, [poznamka]
, [id_nemocnice]
FROM [Nemocnice].[dbo].[ElektromechanickeZP]
```

00 %									
Results		Messages							
id_ElektroMechZP	nazev	popisVyuz	cenaKus	dph	cenaSdph	pocetKusu	poznamka	id_nemocnice	

- Do tabulky se zapisuje název pomůcky, popis využití pomůcky, cena na kus, dph, cena s dph, počet kusů, poznámka (není povinný) a cizí klíč nemocnice

➤ Oddeleni

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky Oddeleni */  
create table Oddeleni(  
    id_od int identity(1,1) primary key,  
    nazev nvarchar(100) not null,  
    druhOboru nvarchar(100) not null,  
    poznamka nvarchar(500) not null  
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_od]  
    , [nazev]  
    , [druhOboru]  
    , [poznamka]  
FROM [Nemocnice].[dbo].[Oddeleni]
```

100 %

Results Messages

id_od	nazev	druhOboru	poznamka
-------	-------	-----------	----------

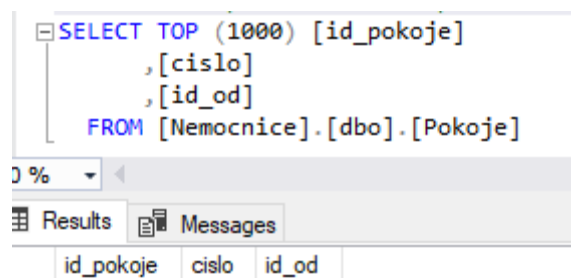
- Do tabulky se zapisuje název oddělení, druh oddělení, poznámka (není povinná)

➤ Pokoje

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky Pokoje a fk oddeleni */
create table Pokoje(
    id_pokoje int identity(1,1) primary key,
    cislo int not null,
    id_od int not null,
    constraint fk_pokoje_oddeleni foreign key(id_od) references Oddeleni(id_od)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a SQL query: `SELECT TOP (1000) [id_pokoje], [cislo], [id_od] FROM [Nemocnice].[dbo].[Pokoje]`. The bottom pane shows the 'Results' tab with a table containing three columns: `id_pokoje`, `cislo`, and `id_od`. The 'Messages' tab is also visible but empty.

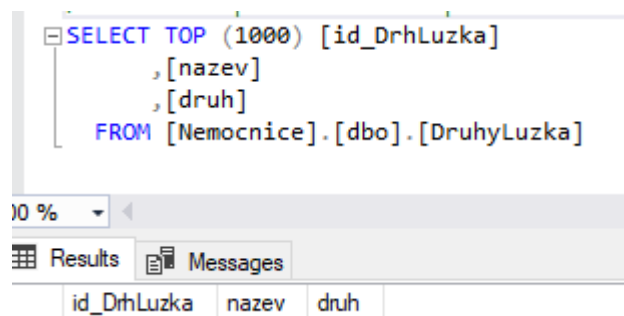
- Do tabulky se zapisuje číslo pokoje a cizí klíč nemocnice

➤ DruhyLuzka

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky DruhyLuzka */
create table DruhyLuzka(
    id_DrhLuzka int identity(1,1) primary key,
    nazev nvarchar(100) not null,
    druh nvarchar(100) not null
);
```

- Print – screen aplikace tabulky v MS SQL Serveru



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a SQL query: `SELECT TOP (1000) [id_DrhLuzka], [nazev], [druh] FROM [Nemocnice].[dbo].[DruhyLuzka]`. Below the query, the 'Results' tab is active, showing a table with three columns: `id_DrhLuzka`, `nazev`, and `druh`. The table is currently empty.

- Do tabulky se zapisuje název a druh lůžka

➤ Luzka

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky Luzka a fk Pokoje a DruhyLuzka */
create table Luzka(
    id_luzko int identity(1,1) primary key,
    nazev nvarchar(100) not null,
    cisloLuz int not null,
    maxVaha int not null,
    id_pokoje int not null,
    id_DrhLuzka int not null,
    constraint fk_luzka_pokoje foreign key(id_pokoje) references Pokoje(id_pokoje),
    constraint fk_luzka_druhyLuzka foreign key(id_DrhLuzka) references DruhyLuzka(id_DrhLuzka)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_luzko]
, [nazev]
, [cisloLuz]
, [maxVaha]
, [id_pokoje]
, [id_DrhLuzka]
FROM [Nemocnice].[dbo].[Luzka]
```

0 %					
Results Messages					
id_luzko	nazev	cisloLuz	maxVaha	id_pokoje	id_DrhLuzka

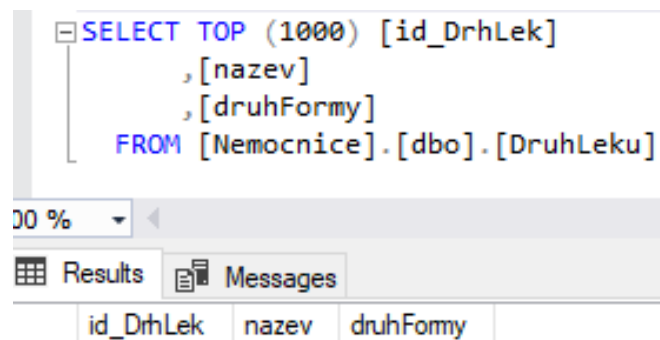
- Do tabulky se zapisuje název lůžka, číslo lůžka, maximální váha lůžka (v kg), cizí klíč pokoje a cizí klíč druhu lůžka

➤ DruhLeku

- Print – screen vytvoření tabulky a integrity

```
/*   Vytvoreni tabulky DruhLeku   */  
create table DruhLeku(  
    id_DrhLek int identity(1,1) primary key,  
    nazev nvarchar(50) not null,  
    druhFormy nvarchar(50) not null  
);
```

- Print – screen aplikace tabulky v MS SQL Serveru



The screenshot shows a SQL query window with the following text:

```
SELECT TOP (1000) [id_DrhLek]  
    , [nazev]  
    , [druhFormy]  
FROM [Nemocnice].[dbo].[DruhLeku]
```

Below the query window, there is a results pane showing the first few rows of data. The columns are labeled 'id_DrhLek', 'nazev', and 'druhFormy'. The first row shows '1', 'Aspirin', and 'Tablety'.

id_DrhLek	nazev	druhFormy
1	Aspirin	Tablety
2	Paracetamol	Tablety
3	Ibuprofen	Tablety
4	Acetaminofen	Tablety
5	Salicylová kyselina	Tablety

- Do tabulky se zapisuje název druhu léku, forma léku

➤ Vyrobce

- Print – screen vytvoření tabulky a integrity

```
/*   Vytvoreni tabulky Vyrobce   */  
create table Vyrobce(  
    id_vyrobce int identity(1,1) primary key,  
    nazev nvarchar(50) not null,  
    adresa nvarchar(50) not null  
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_vyrobce]  
                , [nazev]  
                , [adresa]  
FROM [Nemocnice].[dbo].[Vyrobce]
```

0 %

Results Messages

id_vyrobce	nazev	adresa
------------	-------	--------

- Do tabulky se zapisuje název výrobce, adresa výrobce

➤ Leky

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky Leky a fk DruhLeku a Vyrobcce */
create table Leky(
    id_lek int identity(1,1) primary key,
    nazev nvarchar(50) not null,
    zamereni nvarchar(100) not null,
    datSpotreby date not null,
    slozeni nvarchar(500) not null,
    cena float not null,
    dph nvarchar(10) not null,
    cenaSdph float not null,
    pocetNasklade int not null,
    id_DrhLek int not null,
    id_vyrobcce int not null,
    constraint fk_leky_druhLeku foreign key(id_DrhLek) references DruhLeku(id_DrhLek),
    constraint fk_leky_vyrobcce foreign key(id_vyrobcce) references Vyrobcce(id_vyrobcce)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_lek]
, [nazev]
, [zamereni]
, [datSpotreby]
, [slozeni]
, [cena]
, [dph]
, [cenaSdph]
, [pocetNasklade]
, [id_DrhLek]
, [id_vyrobcce]
FROM [Nemocnice].[dbo].[Leky]
```

00 %

Results Messages

id_lek	nazev	zamereni	datSpotreby	slozeni	cena	dph	cenaSdph	pocetNasklade	id_DrhLek	id_vyrobcce
--------	-------	----------	-------------	---------	------	-----	----------	---------------	-----------	-------------

- Do tabulky se zapisuje název léku, zaměření léku, datum spotřeby, složení, cenu, dph, cenu s dph, počet na skladě a cizí klíč DruhLeku a Vyrobcce

➤ Snidane

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky Snidane */
create table Snidane(
    id_snidane int identity(1,1) primary key,
    vaha int not null,
    verze nvarchar(100) not null,
    kaloricHodnota int not null,
    cena float not null,
    dph nvarchar(10) not null,
    cenaSdph float not null,
    obsah nvarchar(500)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_snidane]
, [vaha]
, [verze]
, [kaloricHodnota]
, [cena]
, [dph]
, [cenaSdph]
, [obsah]
FROM [Nemocnice].[dbo].[Snidane]
```

100 %

Results Messages

id_snidane	vaha	verze	kaloricHodnota	cena	dph	cenaSdph	obsah
------------	------	-------	----------------	------	-----	----------	-------

- Do tabulky se zapisuje váha snídaně (v gramech), verze jídla, kalorická hodnota, cena dph, cena s dph, obsah jídla

➤ Obed

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky Obed */
create table Obed(
    id_obed int identity(1,1) primary key,
    vaha int not null,
    verze nvarchar(100) not null,
    kaloricHodnota int not null,
    cena float not null,
    dph nvarchar(10) not null,
    cenaSdph float not null,
    obsah nvarchar(500)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_obed]
, [vaha]
, [verze]
, [kaloricHodnota]
, [cena]
, [dph]
, [cenaSdph]
, [obsah]
FROM [Nemocnice].[dbo].[Obed]
```

%							
Results Messages							
id_obed	vaha	verze	kaloricHodnota	cena	dph	cenaSdph	obsah

- Do tabulky se zapisuje váha snídaně (v gramech), verze jídla, kalorická hodnota, cena dph, cena s dph, obsah jídla

➤ Vecere

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky Vecere */
create table Vecere(
    id_vecere int identity(1,1) primary key,
    vaha int not null,
    verze nvarchar(100) not null,
    kaloricHodnota int not null,
    cena float not null,
    dph nvarchar(10) not null,
    cenaSdph float not null,
    obsah nvarchar(500)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_vecere]
, [vaha]
, [verze]
, [kaloricHodnota]
, [cena]
, [dph]
, [cenaSdph]
, [obsah]
FROM [Nemocnice].[dbo].[Vecere]
```

0 %

Results Messages

id_vecere	vaha	verze	kaloricHodnota	cena	dph	cenaSdph	obsah
-----------	------	-------	----------------	------	-----	----------	-------

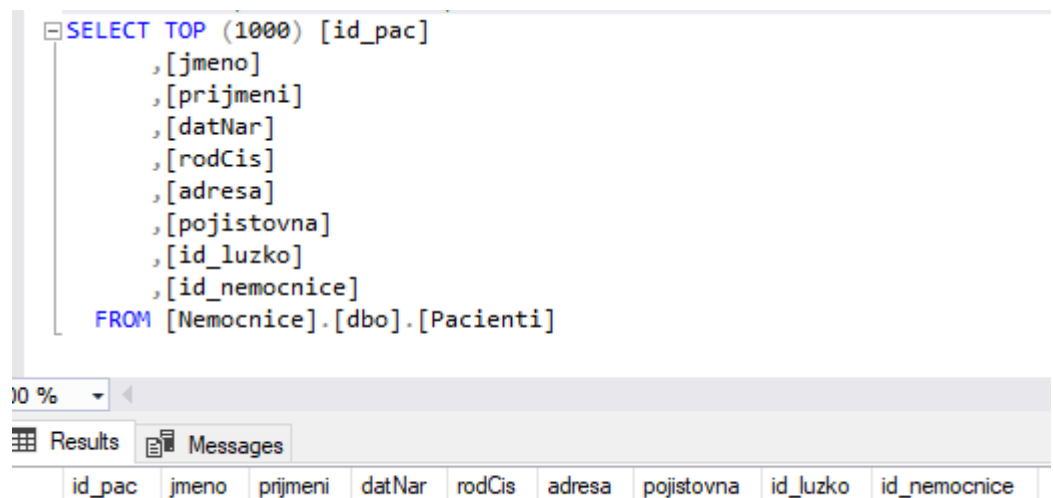
- Do tabulky se zapisuje váha snídaně (v gramech), verze jídla, kalorická hodnota, cena dph, cena s dph, obsah jídla

➤ Pacienti

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky Pacienti a fk Luzka a Nemocnice */
create table Pacienti(
    id_pac int identity(1,1) primary key,
    jmeno nvarchar(50) not null,
    prijmeni nvarchar(50) not null,
    datNar date not null,
    rodCis nvarchar(11) not null,
    adresa nvarchar(50) not null,
    pojistovna nvarchar(50) not null,
    id_luzko int null,
    id_nemocnice int not null,
    constraint fk_pacienti_luzka foreign key(id_luzko) references Luzka(id_luzko),
    constraint fk_pacienti_nemocnice foreign key(id_nemocnice) references Nemocnice(id_nemocnice)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a SQL query: `SELECT TOP (1000) [id_pac], [jmeno], [prijmeni], [datNar], [rodCis], [adresa], [pojistovna], [id_luzko], [id_nemocnice] FROM [Nemocnice].[dbo].[Pacienti]`. The bottom pane shows the 'Results' tab with a table containing 9 columns: id_pac, jmeno, prijmeni, datNar, rodCis, adresa, pojistovna, id_luzko, and id_nemocnice. The zoom level is set to 100%.

- Do tabulky se zapisuje jméno pacienta, příjmení, datum narození, rodné číslo, adresa, pojišťovna, cizí klíč lůžka (není povinné), cizí klíč nemocnice

➤ StravaNaDen

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky StravaNaDen a fk Pacienta, snidane, obeda a vecere*/
create table StravaNaDen(
    id_strava int identity(1,1) primary key,
    datum date not null,
    id_pac int not null,
    id_snidane int not null,
    id_obed int not null,
    id_vecere int not null,
    constraint fk_stravaNaDen_pacienti foreign key(id_pac) references Pacienti(id_pac),
    constraint fk_stravaNaDen_snidane foreign key(id_snidane) references Snidane(id_snidane),
    constraint fk_stravaNaDen_obed foreign key(id_obed) references Obed(id_obed),
    constraint fk_stravaNaDen_vecere foreign key(id_vecere) references Vecere(id_vecere)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_strava]
, [datum]
, [id_pac]
, [id_snidane]
, [id_obed]
, [id_vecere]
FROM [Nemocnice].[dbo].[StravaNaDen]
```

00 %

Results Messages

id_strava	datum	id_pac	id_snidane	id_obed	id_vecere
-----------	-------	--------	------------	---------	-----------

- Do tabulky se zapisuje datum, cizí klíč pacienta, cizí klíč snídaně, oběda a večeře

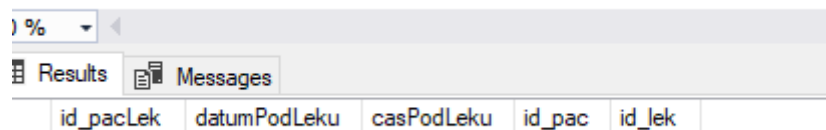
➤ PacLek

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky PacLek a fk Pacienta a leku*/  
create table PacLek(  
    id_pacLek int identity(1,1) primary key,  
    datumPodLeku date not null,  
    casPodLeku time not null,  
    id_pac int not null,  
    id_lek int not null,  
    constraint fk_pacLek_pacienti foreign key(id_pac) references Pacienti(id_pac),  
    constraint fk_pacLek_lek foreign key(id_lek) references Leky(id_lek)  
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_pacLek]  
    , [datumPodLeku]  
    , [casPodLeku]  
    , [id_pac]  
    , [id_lek]  
FROM [Nemocnice].[dbo].[PacLek]
```



The screenshot shows the SQL Server Enterprise Manager interface. At the top, there is a search bar with a dropdown menu set to '%'. Below it, there are two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying the structure of the 'PacLek' table. The table has five columns: 'id_pacLek', 'datumPodLeku', 'casPodLeku', 'id_pac', and 'id_lek'.

id_pacLek	datumPodLeku	casPodLeku	id_pac	id_lek
-----------	--------------	------------	--------	--------

- Do tabulky se zapisuje datum podání léku, čas podání léku a cizí klíč pacienta a léku

➤ Doktori

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky Doktori a fk nemocnice*/  
create table Doktori(  
    id_dok int identity(1,1) primary key,  
    jmeno nvarchar(50) not null,  
    prijmeni nvarchar(50) not null,  
    datNar date not null,  
    rodCis nvarchar(11) not null,  
    adresa nvarchar(50) not null,  
    zamereni nvarchar(100) not null,  
    id_nemocnice int not null,  
    constraint fk_doktori_nemocnice foreign key(id_nemocnice) references Nemocnice(id_nemocnice)  
);
```

- Print – screen aplikace tabulky v MS SQL Serveru

```
SELECT TOP (1000) [id_dok]  
    ,[jmeno]  
    ,[prijmeni]  
    ,[datNar]  
    ,[rodCis]  
    ,[adresa]  
    ,[zamereni]  
    ,[id_nemocnice]  
FROM [Nemocnice].[dbo].[Doktori]
```

0 %							
Results	Messages						
id_dok	jmeno	prijmeni	datNar	rodCis	adresa	zamereni	id_nemocnice

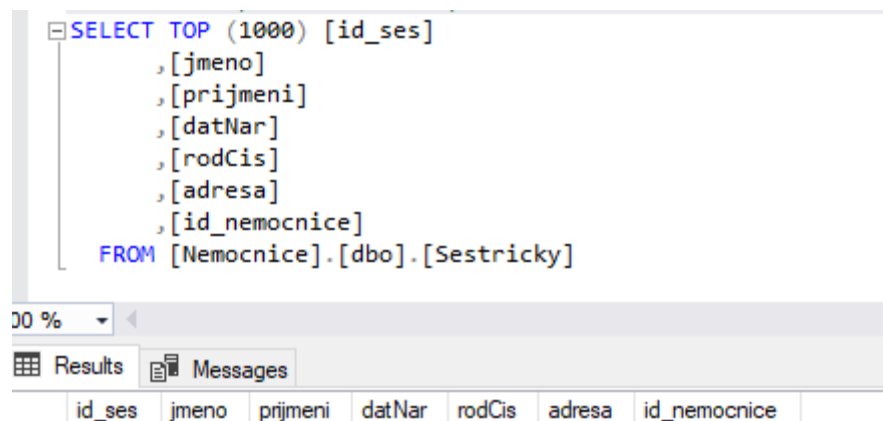
- Do tabulky se zapisuje jméno doktora, příjmení datum narození, rodné číslo, adresu bydliště, zaměření a cizí klíč nemocnice

➤ Sestricky

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky Sestricky a fk nemocnice*/
create table Sestricky(
    id_ses int identity(1,1) primary key,
    jmeno nvarchar(50) not null,
    prijmeni nvarchar(50) not null,
    datNar date not null,
    rodCis nvarchar(11) not null,
    adresa nvarchar(50) not null,
    id_nemocnice int not null,
    constraint fk_sestricky_nemocnice foreign key(id_nemocnice) references Nemocnice(id_nemocnice)
);
```

- Print – screen aplikace tabulky v MS SQL Serveru



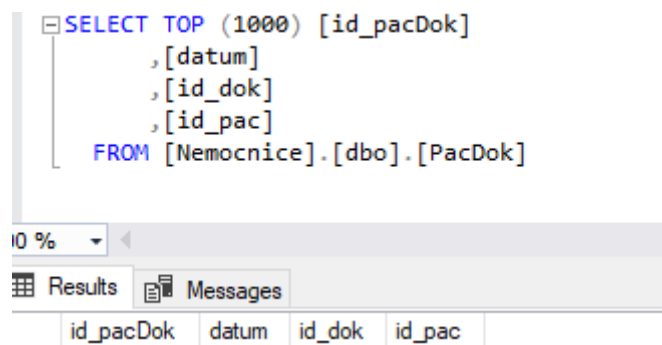
- Do tabulky se zapisuje jméno sestřičky, příjmení datum narození, rodné číslo, adresu bydliště a cizí klíč nemocnice

➤ PacDok

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky PacDok a fk doktoru a pacientu*/
create table PacDok(
    id_pacDok int identity(1,1) primary key,
    datum date not null,
    id_dok int not null,
    id_pac int not null,
    constraint fk_pacDok_doktori foreign key(id_dok) references Doktori(id_dok),
    constraint fk_pacDok_pacient foreign key(id_pac) references Pacienti(id_pac),
);
```

- Print – screen aplikace tabulky v MS SQL Serveru



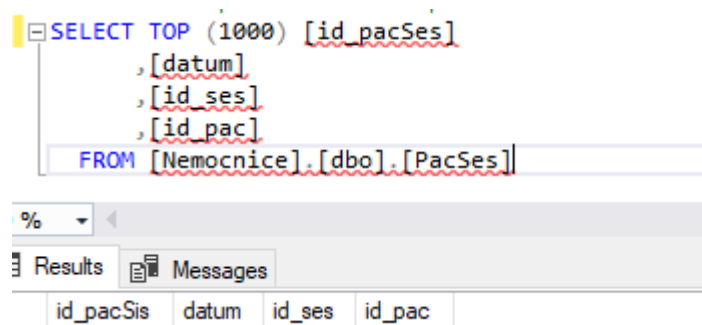
- Do tabulky se zapisuje datum a cizí klíč doktora a pacienta

➤ PacSes

- Print – screen vytvoření tabulky a integrity

```
/* Vytvoreni tabulky PacSis a fk sestricek a pacientu*/  
create table PacSes(  
    id_pacSes int identity(1,1) primary key,  
    datum date not null,  
    id_ses int not null,  
    id_pac int not null,  
    constraint fk_pacDok_sestricky foreign key(id_ses) references Sestricky(id_ses),  
    constraint fk_pacDok_pacienti foreign key(id_pac) references Pacienti(id_pac),  
);
```

- Print – screen aplikace tabulky v MS SQL Serveru



The screenshot shows a SQL query in the Enterprise Manager query window:

```
SELECT TOP (1000) [id_pacSes]  
    , [datum]  
    , [id_ses]  
    , [id_pac]  
FROM [Nemocnice].[dbo].[PacSes]
```

Below the query window, the 'Results' tab is active, displaying the following columns:

id_pacSes	datum	id_ses	id_pac
-----------	-------	--------	--------

- Do tabulky se zapisuje datum a cizí klíč sestřičky a pacienta

4. Závěr

- Změny provést nejsou problém, stačí nás kontaktovat a nějak se domluvíme
- Děkujeme za přečtení tohoto dokumentu