UNIVERZITA PARDUBICE  
Fakulta elektrotechniky a informatiky

*Informační systém psího útulku*

Seminární práce z předmětu Databázové systémy I

Radek Sahliger

V Ústí nad Orlicí dne 5.12.2021

Obsah

[Analýza 10](#_Toc70355807)

[ERD 13](#_Toc70355808)

[Relační model dat 16](#_Toc70355809)

[Fyzický model dat 17](#_Toc70355810)

[Výčet SQL dotazů 18](#_Toc70355811)

[Závěr 22](#_Toc70355812)

[Přílohy 23](#_Toc70355813)

Analýza

Scénář

Databáze je určená pro psí útulek.  
Tím pádem jsou základním stavebním kamenem databáze psi, se kterými budou uživatelé zacházet. Není od věci rozlišovat psi podle toho, jak byli přivedeni do útulku. Pes může být cíleně odchycen pokud se o něj například jeho dosavadní rodina nedokáže postarat. Dále je možné psa nalézt na ulici. Dále útulek provozuje “psí hotel”, což znamená, že majitelé psů mohou dočasně přenechat péči o psa útulku.  
Všechny osoby, které s útulkem nějak spolupracují jsou uvedené v databázi. Osoby mohou napomáhat útulku tím, že venčí psy, darují peníze nebo potřebné věci, či léčí a provádějí vyšetření psů.  
Aby nedošlo k problémům se zákonem, je nutné aby jednotlivé kliniky, které se starají o psy z útulku měli potřebné platné licence.  
Osoba, která venčí psy pravidelně a často si může oblíbit nějakého psa. Ne vždy má tato osoba možnosti psa adoptovat domů. Proto je možné rezervovat si psa na určité datum, aby měl uživatel jistotu, že pes neodejde na venčení s někým jiným.

Strukturální pravidla

* Pes
  + Pes musí být vložen s platným plemenem
  + Pes, který je nově zaveden v databázi musí mít vyplněnou karanténu
  + Pes může mít jméno a detail, ale není to nutné, pokud o psovi nic nevíme a je možno doplnit později (pokud psa pojmenujeme nebo pokud je problémový)
* Odchyceny a Nalezeny
  + U odchycený i nalezených psech je nutné uvést místo, kde byl pes chycen (Stačí slovní popis místa, popřípadě ulice s městem/vesnicí)
  + Také je potřeba uvést datum, kdy byl pes chycen
* Odlozeny
  + Odložený pes musí náležet jedné osobě z databáze
  + Při zavedení odloženého psa je nutné uvést předem domluvený časový interval, přes který má být pes ubytovaný v útulku (zavedeno datumem převzetí psa a konečným datumem, kdy bude pes navrácen)
* Osoba
  + Všechny sloupce musí být vyplněné při zavádění osoby do databáze (jméno, příjmení, adresa a telefonní číslo)
  + Všechny sloupce kromě id jsou ve formě textu
* VeterinarniKlinika
  + Kliniku musí vlastnit osoba již zavedená do databáze
  + Adresa kliniky je ve formě textu
* Licence
  + Klinice musí být přiřazená alespoň jedna licence
  + Licence mohou být zastaralé
  + Licence mají přesně dané období na kdy jsou platné
* Procedura
  + Procedura musí být spojena s klinikou, kde byla daná procedura prováděna
  + Vyšetřovat lze pouze zavedené psy
  + Musí obsahovat popis dané procedury a datum, kdy byla provedena
  + U každé procedury je třeba uvést výši ceny za proceduru
* Plemeno
  + Plemeno musí mít svůj název
  + Popis plemena není povinný, záleží pouze na iniciativu uživatele
* Incident
  + Incident může provést pouze pes z databáze
  + Je nutné zapsat datum kdy se incident děl
  + Jen nutné popsat, co přesně daný pes zavinil
* Umrti
  + Úmrtí lze připojit k zavedenému psu
  + Je nutné uvést datum úmrtí psa
  + Dále je zapotřebí uvést zda byl pes utracen nebo jestli zemřel přirozeně (Toto je zaváděno pomocí sloupce utra\_prir, kde 0 znamená to, že byl pes utracen a 1 znamená, že pes zemřel přirozeně)
* Dar
  + Útulek může přijmout dar pouze od osoby zavedené v databázi
  + Je nutné uvést datum převzetí daru
  + Sloupec obnos je nepovinný záznam peněžní částky daru
  + Sloupec ostatní uvádí ve formě textu popis daru, jiného než jsou peníze(např. krmivo)
* Rezervace
  + Rezervace může provádět pouze zavedená osoba
  + K rezervaci je nutné připojit určitého psa
  + Datum, na kdy je pes rezervován je povinný
* AdoptovanyNavraceny
  + Záznam v této tabulce lze spojovat pouze se zavedenými psy a osobami
  + Datum adopce/navrácení je povinné zapsat
  + Dále je nutné uvést, zda byl pes adoptován nebo navrácen původnímu majiteli (Toto se zavádí pomocí záznamu adop\_nav, kde 0 znamená adopci a 1 znamená navrácení)
* Vychazka
  + Venčit lze pouze zavedené psy zavedenými osobami
  + Datum je povinné zapsat

Procedurální pravidla

* Pes
  + Pes nově zavedený do databáze musí mít položku karanténa rovno 1
  + Pokud je pes v útulku déle jak 30 dní, tak je nutné změnit stav karantény na 0
* Incident a Umrti
  + Incident může zavinit pouze živý pes
  + Zemřít může pouze živý pes
* Rezervace, AdoptovanyNavraceny, Vychazka, Procedura
  + K těmto záznamům může být připojen pouze živý pes
  + K rezervaci, adopci a vycházce může být připojen pouze pes, který má položku karanténa rovno 0 (tudíž byl v útulku déle jak 30 dní)
  + Proceduru může provádět pouze klinika s platnou licencí
* Dar
  + U daru je nutné uvést alespoň jednu ze dvou položek obnos a ostatní
  + Pokud dar není peněžní je nutné zapsat ostatní hmotné darované věci (např. krmivo)
  + Pokud je dar pouze peněžní je nutné vyplnit obnos, který byl darován
  + Dar může být peněžní a zároveň lze darovat i jiné věci

Jednoduchá analýza

1.**Pes** - Tato entita představuje Psa samotného. Obsahuje atributy i*d-pes, jmeno, detail* a *karantena*. Pes nemusí mít *jméno*, pokud nebylo zjištěno nebo pokud byl nalezen.  
*Detail* obsahuje popis psa a jeho povahy, případně jaké opatření se musí dodržovat při zacházení se psem. Poslední atribut *karantena* udává, zda pes prošel 30-denní karanténou po převzetí psa do útulku, po této době je možné psa brát na vycházky.

2.**Nalezeny** - Tato entita označuje nalezeného psa a je podtypem entity Pes. Obsahuje atributy *misto* a *datum*. Atribut *misto* označuje místo, kde byl pes nalezen a *datum* označuje kdy byl nalezen.

3.**Odchyceny** - Tato entita označuje odchyceného psa.  
Obsahuje stejné atributy jako entita **Nalezeny**.

4.**Odlozeny** - Tato entita označuje psa, který byl odložen do útulku majitelem na nějakuou stanovenou dobu. Entita obsahuje atributy *datumOd* a *datumDo*, které obsahují dobu, kdy bude pes uchováván v útulku.

5.**Plemeno** - Entita udává plemeno psa. Entita obsahuje *id-plemeno, nazev* a *popis*. Atribut *popis* obsahuje popis plemena, případně povahu plemena.

6.**Osoba** - Entita osob, které interagují se psy a útulkem. Obsahuje atributy i*d\_osoba, jmeno, prijmeni, adresa*.

7.**Rezervace** - Osoby si mohou rezervovat nějakého psa na vycházku. Obsahuje aributy *id\_rezervace* a *datum*.

8.**AdoptovanyNavraceny** - Entita označuje navrácení nebo adopci psa osobou. Obsahuje *datum* a *adop\_nav* který označuje, zda byl pes navrácen majiteli nebo adoptovaný osobou.

9.**Vychazky** - Pes může jít s osobou na vycházku. Obsahuje *id\_vychazka* a *datum*.

10.**VeterinarniKlinika** - Osoba může vlastnit veterinární kliniku. Obsahuje *id-klinika* a *adresa*.

11.**Licence** - Každá klinika musí mít platnou licenci pro ošetřování psů a ostatních zvířat. Klinika si obnovuje licenci, pokud dojde její platnost. Obsahuje *id\_licence, datumOd* a *datumDo*.

12.**Dar** - Osoba může darovat peníze útulku. Obsahuje *id\_dar, obnos, ostatni* a *datum*. *Obnos* určuje množství peněz, které osoba darovala. *Ostatni* určuje jiné materiální dary.

13.**Procedura** - Pokud je pes nemocný nebo má úraz, tak klinika může provést proceduru na daném psovi. Obsahuje *id\_procedura, popis, datum* a *cena*. Do *popisu* je nutné zapsat co bylo danému psu provedeno. *Cena* určuje jaký obnos peněz stála daná procedura

14.**Incident** - Psi mohou být nevyzpytatelní a mohou provést nějaký incident.  
Incident obsahuje *id\_incident, datum* a popis *incidentu*.

15.**Umrti** - Psi také mohou zemřít v době pobytu v útulku. Mohou zemřít přirozeně nebo se zaměstnanci útulku shodnou, že je potřeba psa utratit.  
Obsahuje atributy *id\_umrti, datum* a *utra\_prir,* který udává jakým způsobem pes zemřel.

CRUD analýza

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | C | R | U | D |
| Pes | Ano | Ano | Ano | Ano |
| Nalezeny | Ano | Ano | Ano | Ano |
| Odchyceny | Ano | Ano | Ano | Ano |
| Odlozeny | Ano | Ano | Ano | Ano |
| Plemeno | Ano | Ano | --- | --- |
| Osoba | Ano | Ano | Ano | --- |
| Rezervace | Ano | Ano | --- | Ano |
| AdoptovanyNavraceny | Ano | Ano | --- | Ano |
| Vychazky | Ano | Ano | --- | Ano |
| VeterinarniKlinika | Ano | Ano | Ano | --- |
| Licence | Ano | Ano | --- | Ano |
| Dar | Ano | Ano | --- | --- |
| Procedura | Ano | Ano | --- | Ano |
| Incident | Ano | Ano | --- | Ano |
| Umrti | Ano | Ano | --- | Ano |

**Pes**

Psa můžeme vytvořit, zobrazit a upravovat ho, pokud je k psovi potřeba dopsat povahu nebo pokud je pojmenován. Pokud je pes dlouho po smrti, mohou zaměstnanci psa z databáze smazat, ale zároveň musí smazat i s ním spojené záznamy a jeho subtyp.

**Nalezeny**

Nalezený pes může být přidán do databáze a zobrazen. Také je možná úprava záznamu, pokud se stejný pes vrací do útulku a je potřeba upravit datum nalezení. Nalezeného psa můžeme smazat z databáze, pokud smažeme i s ním spojené záznamy.

**Odchyceny**

U odchyceného psa jsou podmínky stejné jako u nalezeného psa.

**Odlozeny**

U odloženého psa také platí stejné podmínky jako u nalezeného a odchyceného psa.

**Plemeno**

Plemeno můžeme přidat a zobrazovat. Plemeno se již nemůže měnit a zůstává stejné. Také není důvod plemeno odstraňovat z databáze.

**Osoba**

Osobu můžeme přidávat a zobrazovat. Záznam osoby lze měnit, pokud si osoba například změnila jméno nebo adresu. Osoby nelze odstraňovat z databáze.

**Rezervace**

Rezervace můžeme přidávat a zobrazovat. Rezervace se nemůže měnit. Je možné odstranit rezervaci, pokud již připojený pes nebude udržován v databázi.

**AdoptovanyNavraceny**

U adopce platí stejné podmínky jako u rezervací.

**Vychazky**

Uvycházek platí stejné podmínky jako u rezervací a adopcí.

**VeterinarniKlinika**

Veterinární kliniky můžeme přidávat a zobrazovat. Kliniku lze také měnit, pokud se klinika přesune na jinou adresu. Kliniku není možné odstranit, kvůli dokumentaci procedur.

**Licence**

Licence můžeme přidávat a zobrazovat. Licence ale nelze měnit a zůstávají stejné. Také nelze odstraňovat licence kvůli dokumentaci klinik.

**Dar**

Dary lez přidávat a zobrazovat. Dar ale nelze změnit a také odstranit.

**Procedura**

Procedury můžeme přidávat a zobrazovat. Proceduru nelze změnit. Proceduru je ale možné odstranit, pokud se připojený pes bude odstraňovat z databáze.

**Incident**

Incidenty lze přidávat a zobrazovat. Incident nelze změnit. Také jej lze odstranit, pokud se bude připojený pes mazat z databáze.

**Umrti**

U úmrtí psů platí stejné podmínky jako u incidentů.

Smyčky

V mém databázovém modelu se nachází smyčky. Tabulka PSI a tabulka OSOBY jsou spojeny tabulkami REZERVACE, VYCHAZKY, ODLOZENI a ADOPTOVANINAVRACENI. Přes tyto tabulky se lze dostat od psa k osobě a opět zpět k psovi. Také je osoba spojena s tabulkou VETRINARNIKLINIKY a pes s tabulkou PROCEDURA. Tyto spoje také umožňují smyčky. Všechny tyto tabulky považuji za důležitou součást mého databázového modelu a proto jsem se rozhodnul smyčky ponechat.

Integritní omezení

IO1: Nově zavedený pes musí mít karanténu rovno 1

IO2: Pes může být venčen, pokud není spojen s úmrtím

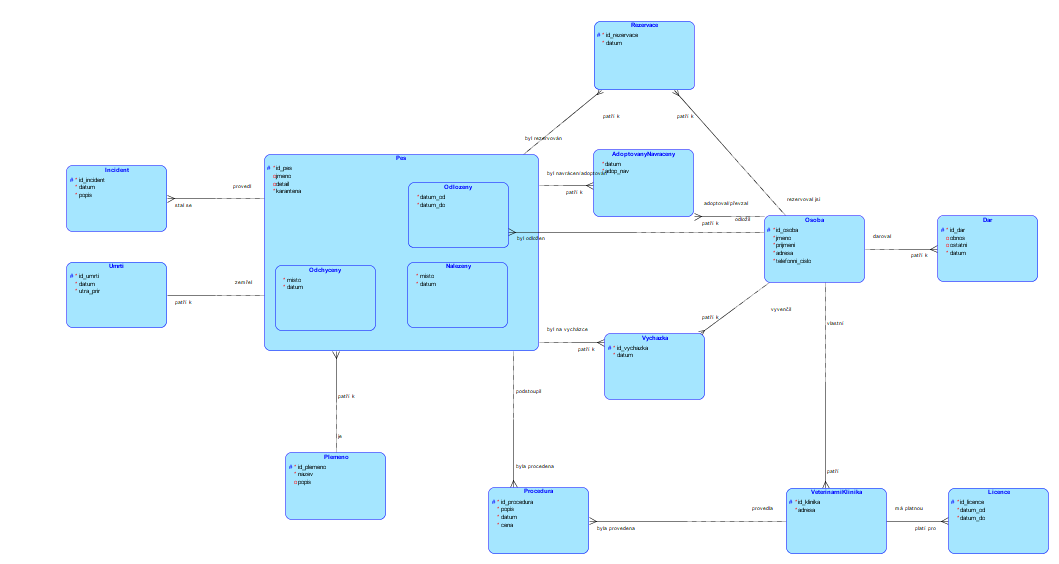
IO3: Pes může být rezervován, pokud není spojen s úmrtím

IO4: Pes může být adoptován, pokud není spojen s úmrtím

IO5: Dar musí být buď peněžní a majetkový nebo jen peněžní nebo jen majetkový

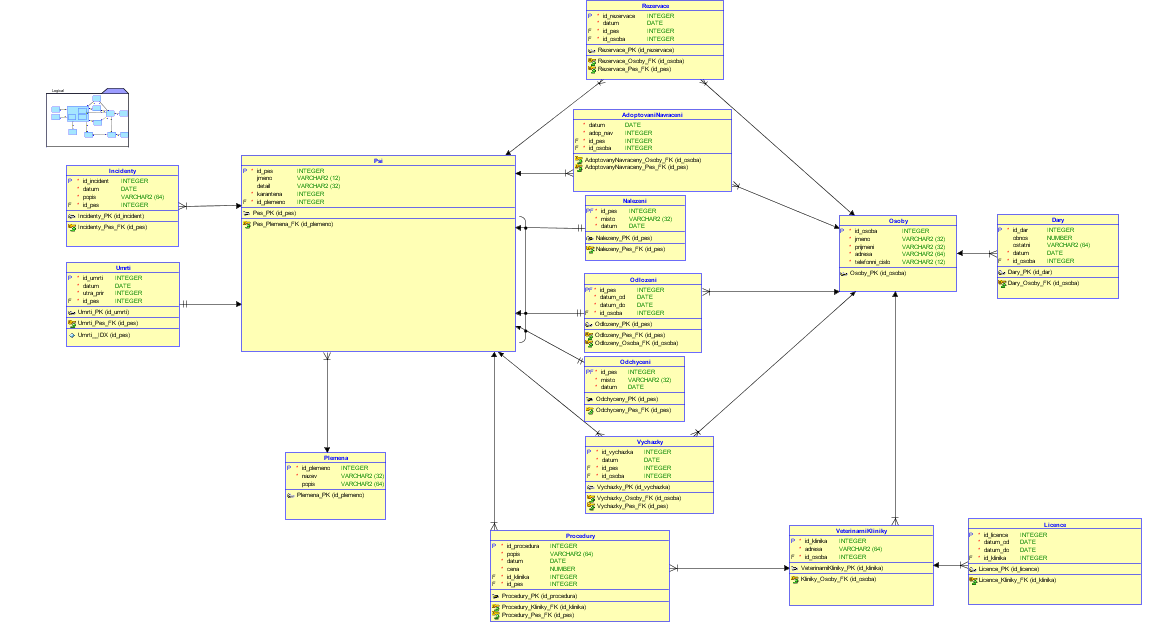
IO5: Aby klinika mohla provádět procedury, musí mít platnou licenci

## ERD



* Pes
  + Pes může zavinit více incidentů
  + Pes může jednou zemřít
  + Pes musí mít jedno plemeno
  + Pes může podstoupit více procedur
  + Pes může být několikrát vyvenčen
  + Pes může být několikrát rezervován
  + Pes může být několikrát adoptován
* Odlozeny
  + Odložený pes musí být spojen s jednou osobou
* Incident
  + Incident musí provést jeden pes z databáze
* Umrti
  + Zemřít musí jeden pes z databáze
* Rezervace
  + Rezervace musí mít jednoho přiřazeného psa
  + Rezervace musí mít jednu přiřazenou osobu
* AdoptovanyNavraceny
  + Adopce musí být spojena s jedním psem
  + Adopce musí být spojena s jednou osobou
* Vychazka
  + Vycházka musí být spojena s jedním psem
  + Vycházka musí být spojena s jednou osobou
* Procedura
  + Procedura musí být spojena s jedním psem
  + Procedura musí být spojena s jednou klinikou
* Osoba
  + Osoba může rezervovat několik psů
  + Osoba může venčit několik psů
  + Osoba může adoptovat několik psů
  + Osoba může odložit několik psů
  + Osoba může vlastnit několik klinik
  + Osoba může darovat několik darů
* VeterinarniKlinika
  + Klinika musí být spojena s jednou osobou
  + Klinika musí být spojena s jednou nebo několika licencemi
* Dar
  + Dar musí být spojený s jednou osobou
* Licence
  + Licence musí být spojena s jednou klinikou

## Relační model dat

Po vytvoření relačního modelu jsem provedl několik malých úprav. Cizí klíče vytvořené data modelerem obsahovaly prefixy se jmény odkazovaných tabulek ve jméně. Tyto prefixy jsem odstranil, aby se jména cizích klíčů shodovaly s primárními klíči odkazovaných tabulek.  
Protože jména některých tabulek jsou poměrně dlouhá, bylo nutné zkrátit názvy některých constraintů cizích klíčů.

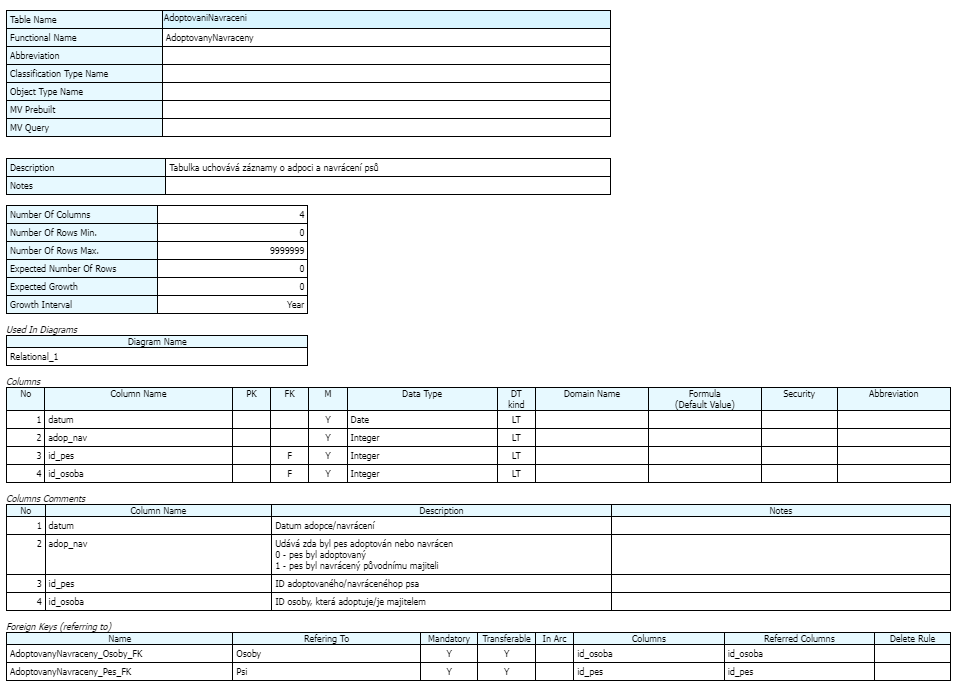
Subtypy psů jsem se rozhodnul realizovat pomocí arcu. Každý typ psa obsahuje klíč id\_psa.  
Tabulka pes obsahuje diskriminátor typ\_psa, který pomocí jednoho písmene (N, H nebo O) určuje typ psa, do kterého záznam patří.

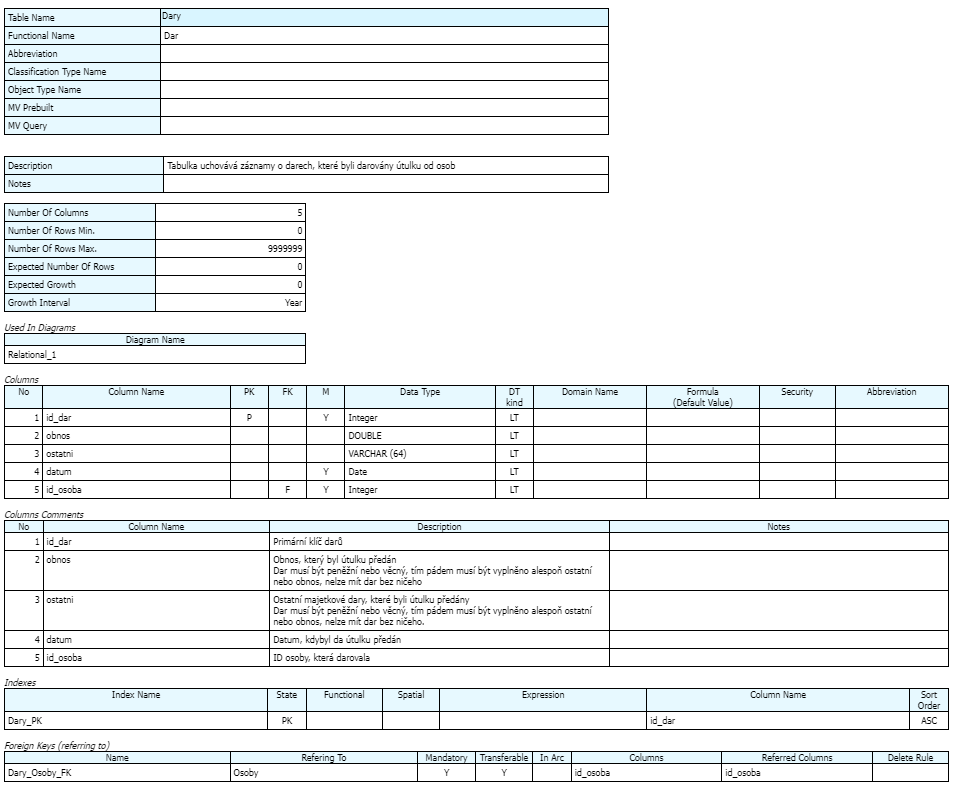
Sekvence + Triggery

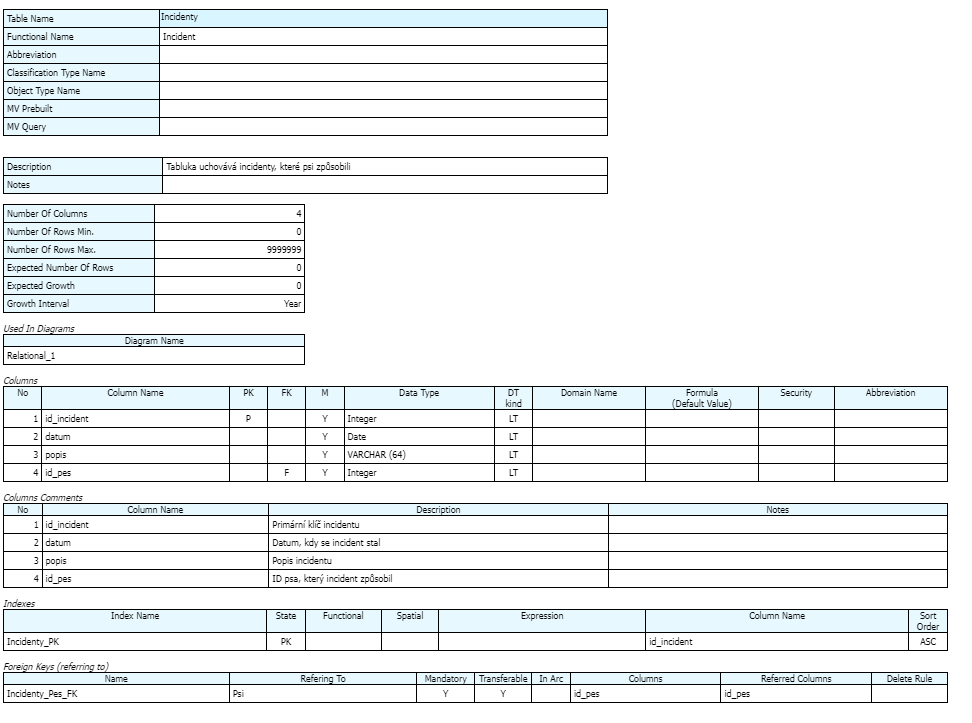
Ve svém projektu nepoužívám sekvence, jelikož se lépe orientuji v hodnotách, které sám vytvořím. Také je pro takto malý projekt zbytečné vytvářet sekvence.

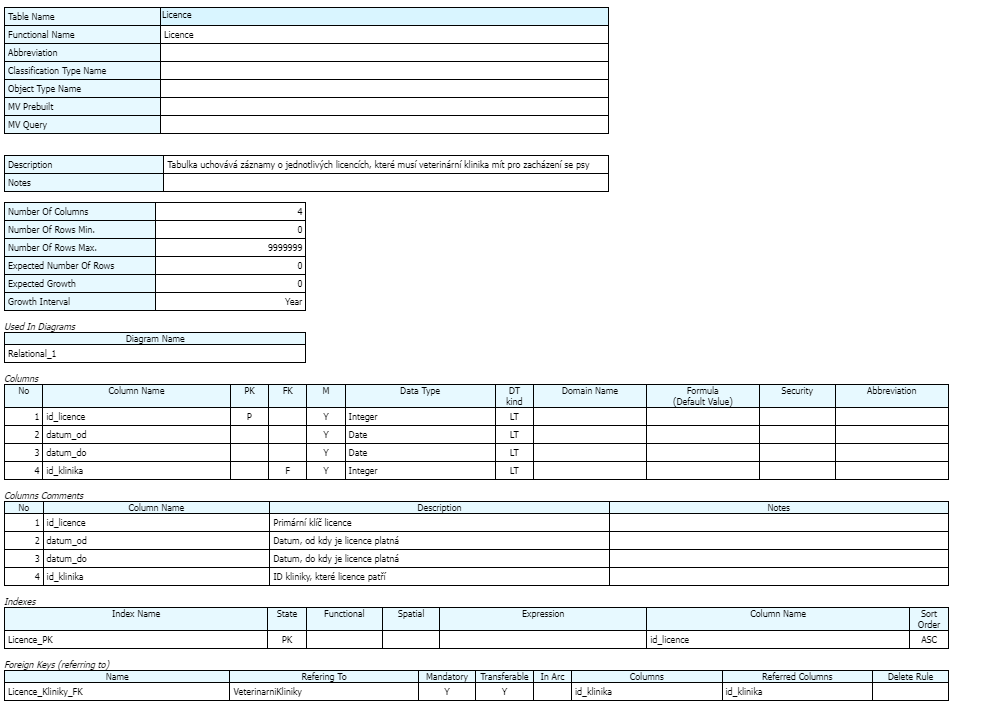
Triggery jsem sám žádné nevytvářel, ale používám triggery vytvořené automaticky při generování DDL skripu data modelerem.

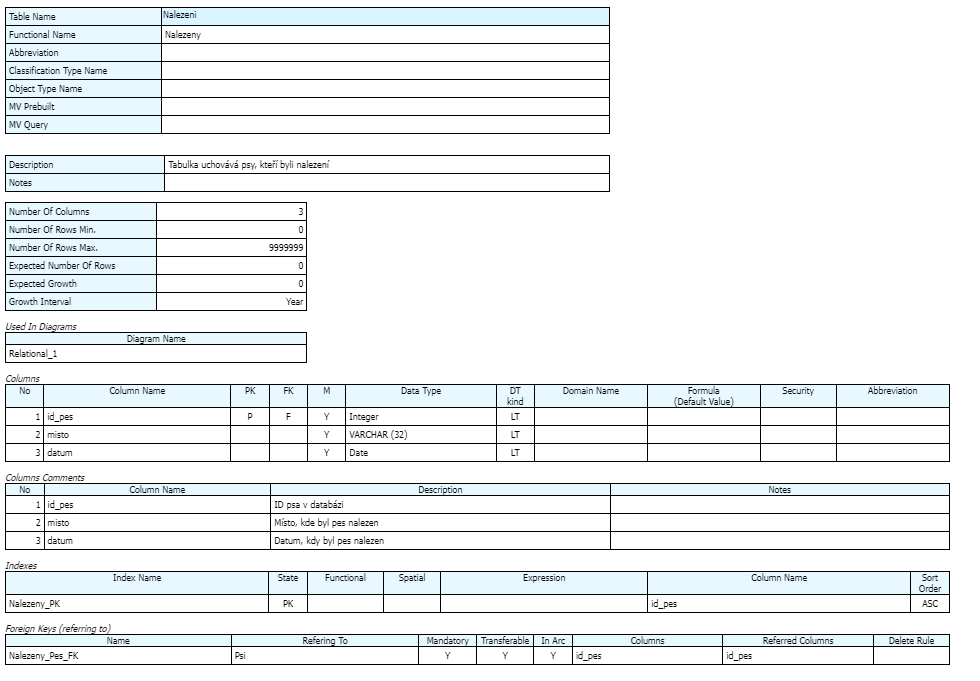
Fyzický model dat

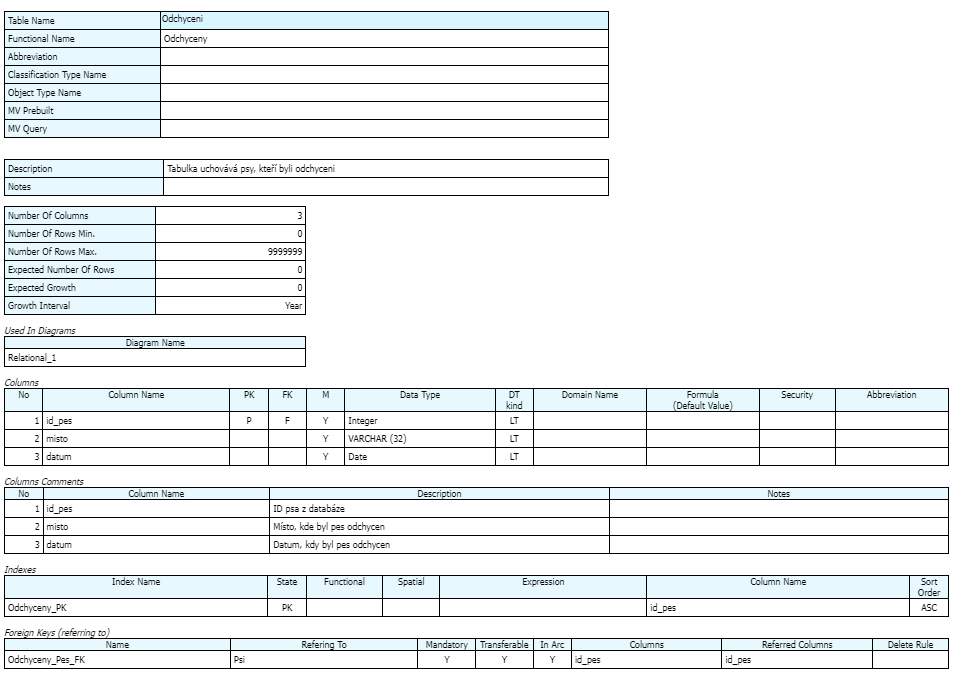


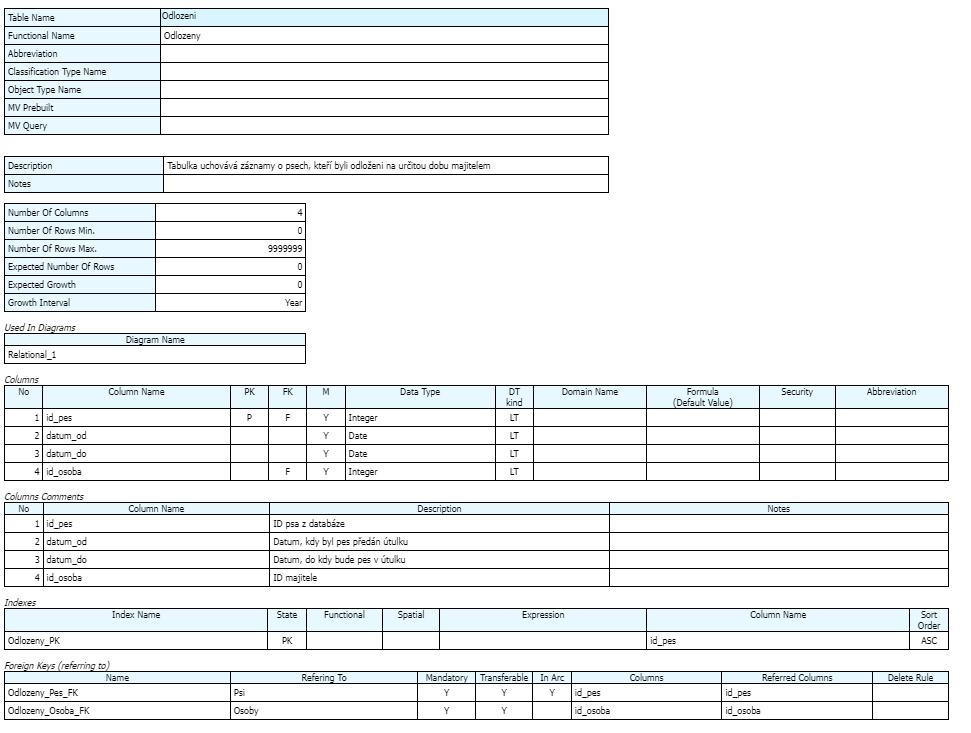


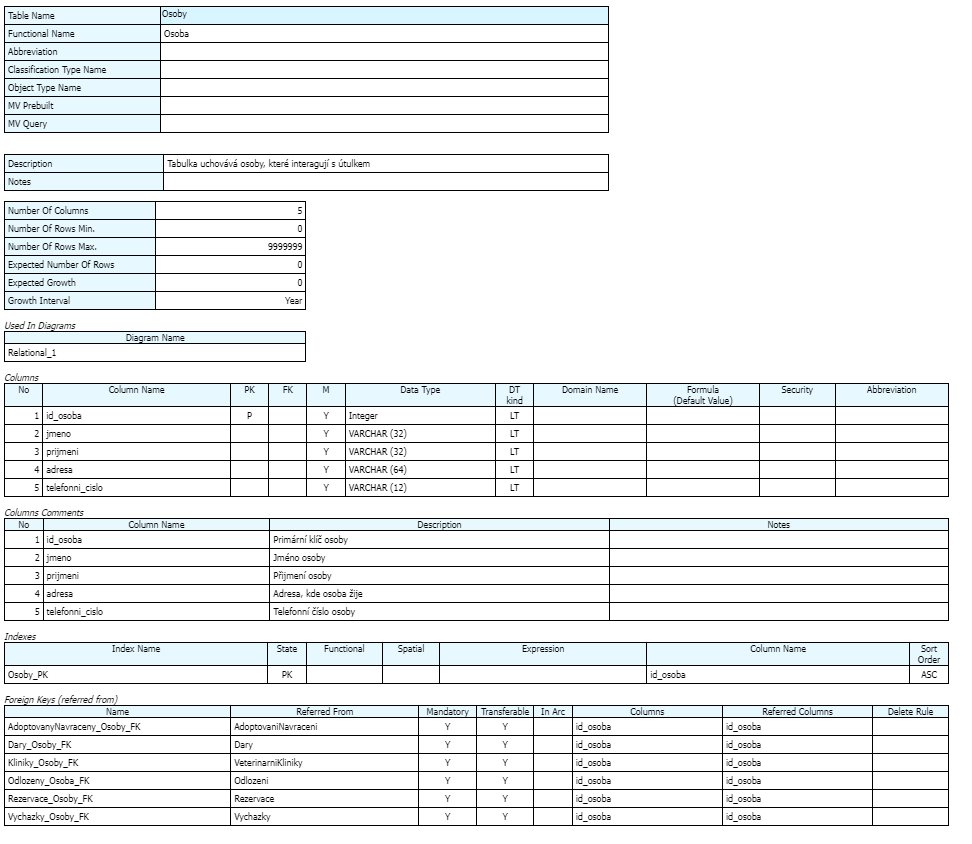


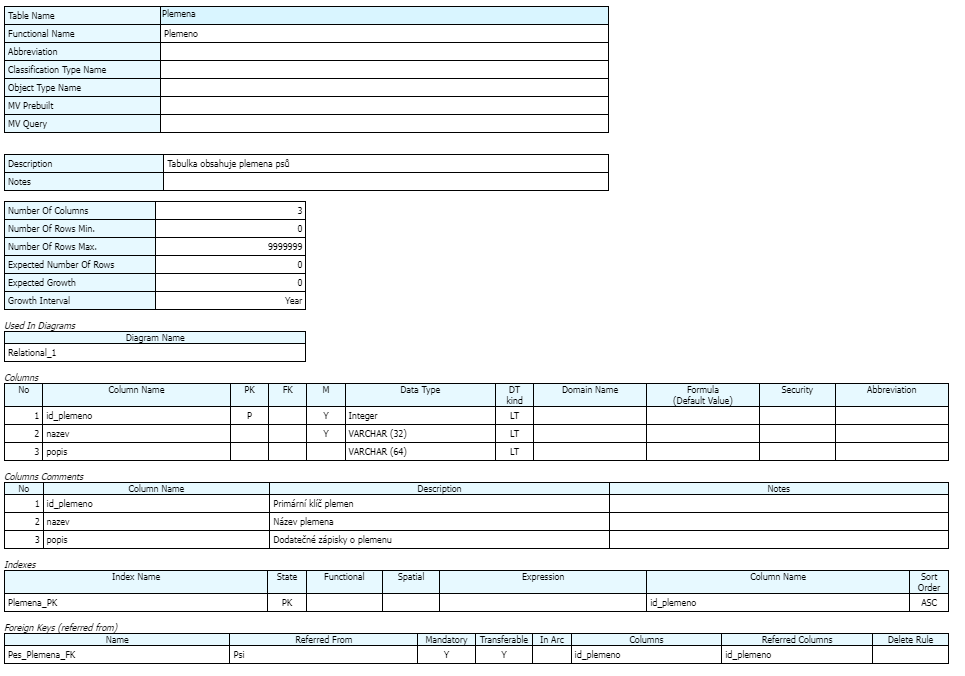


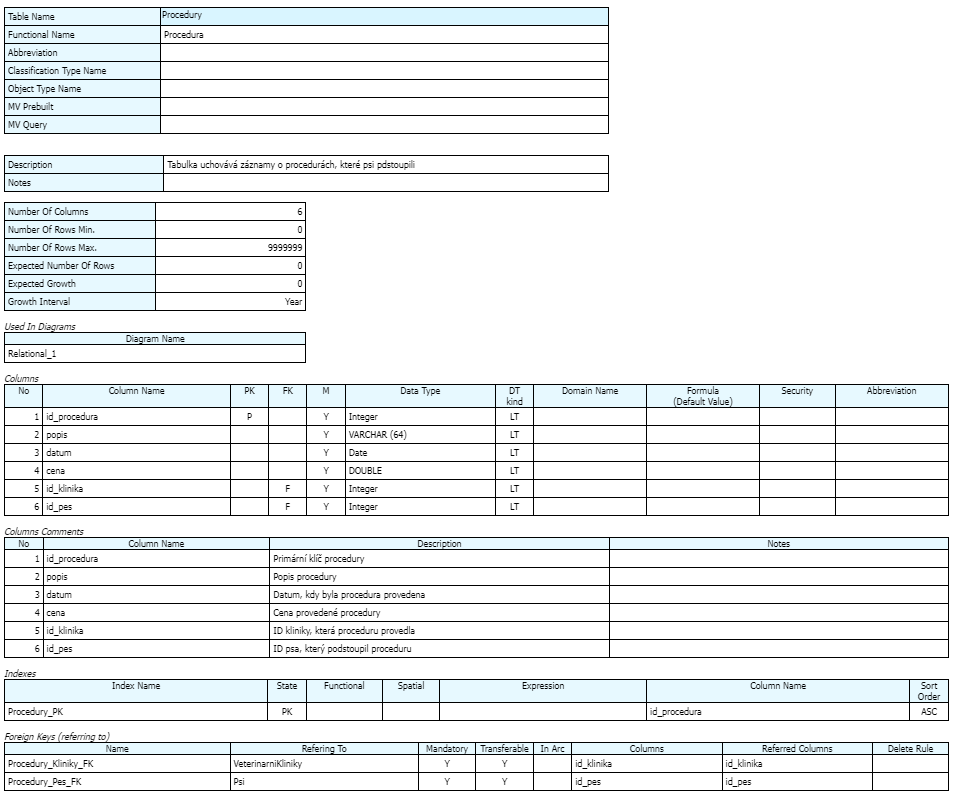


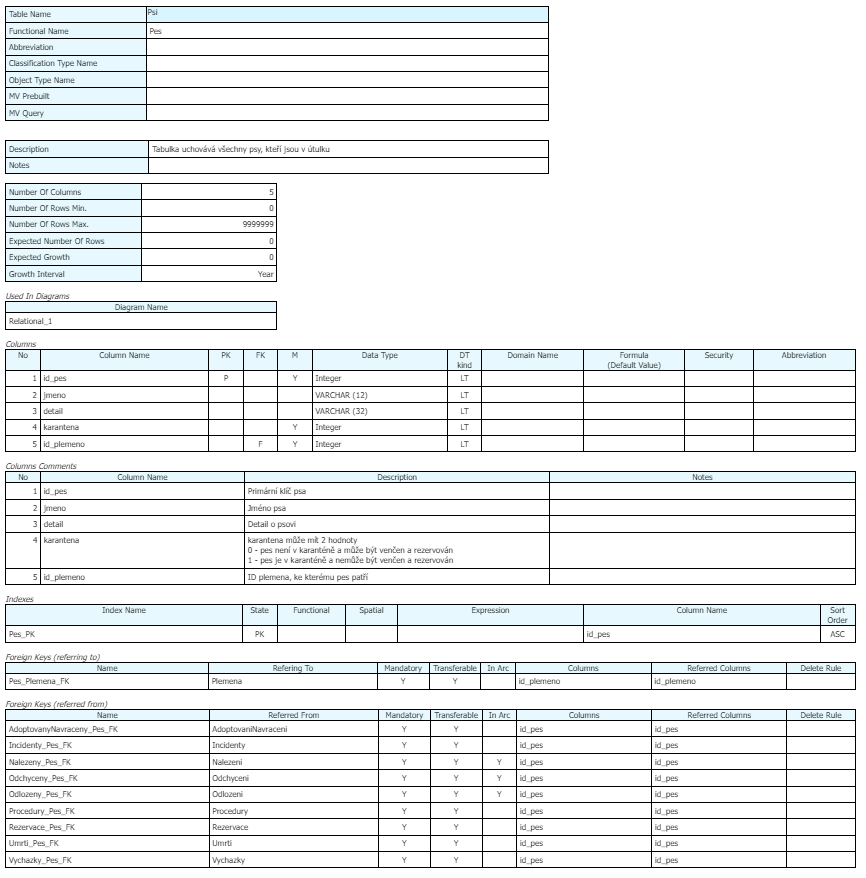


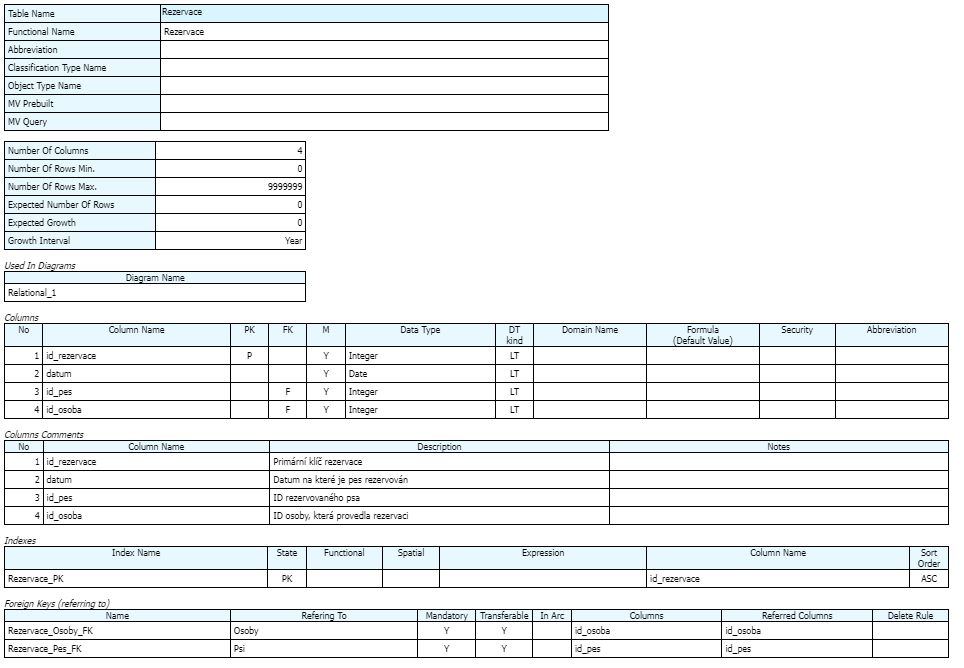


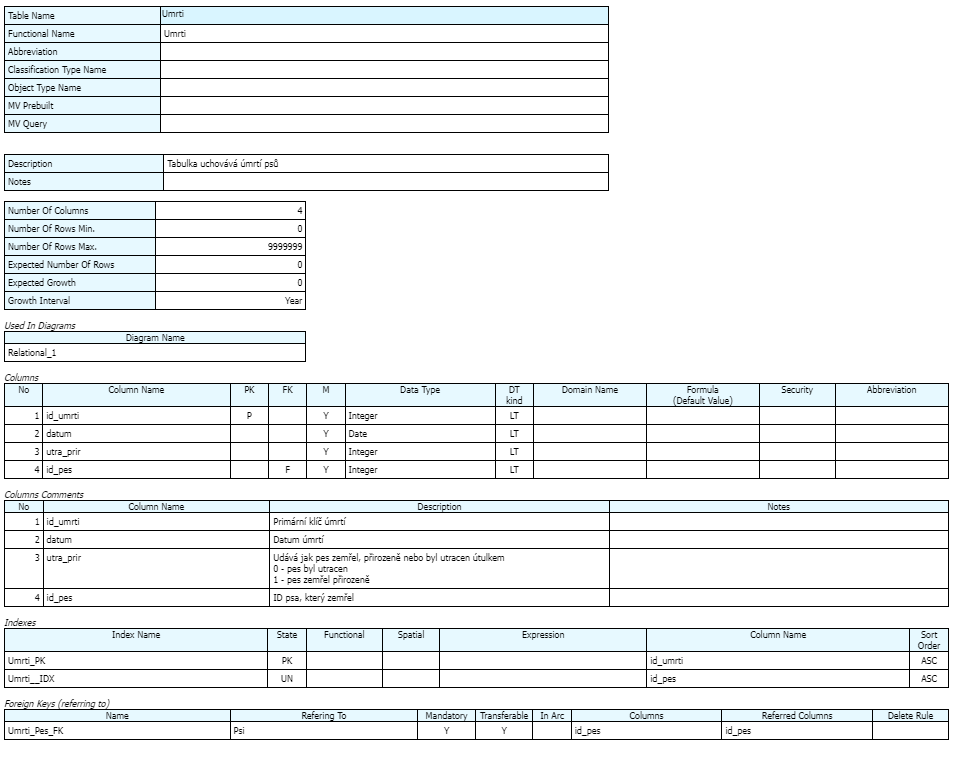


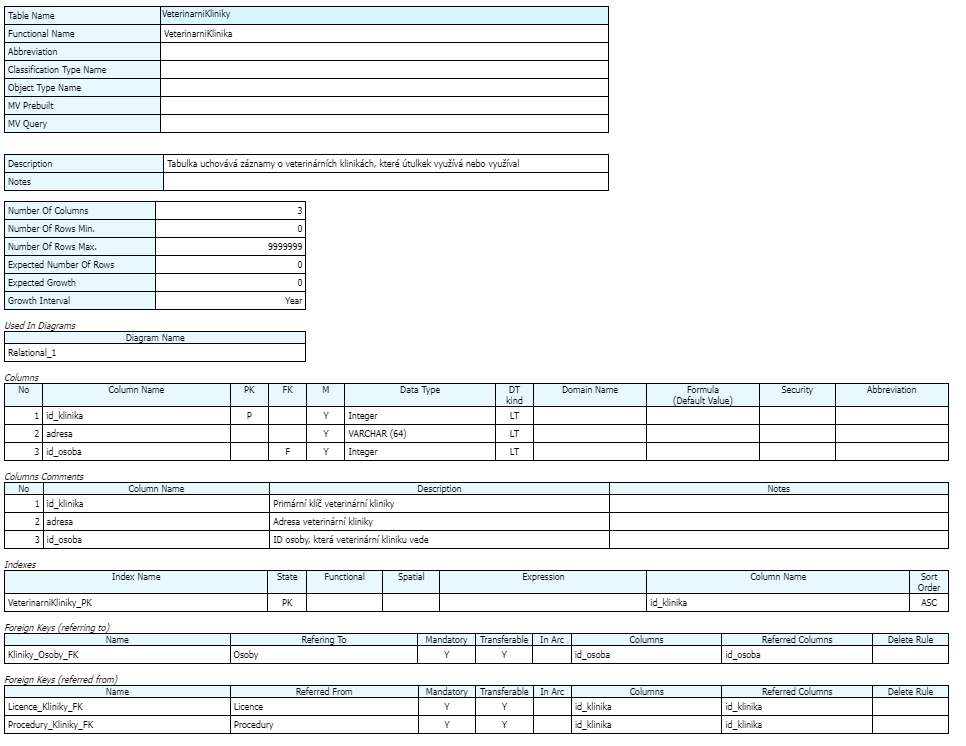


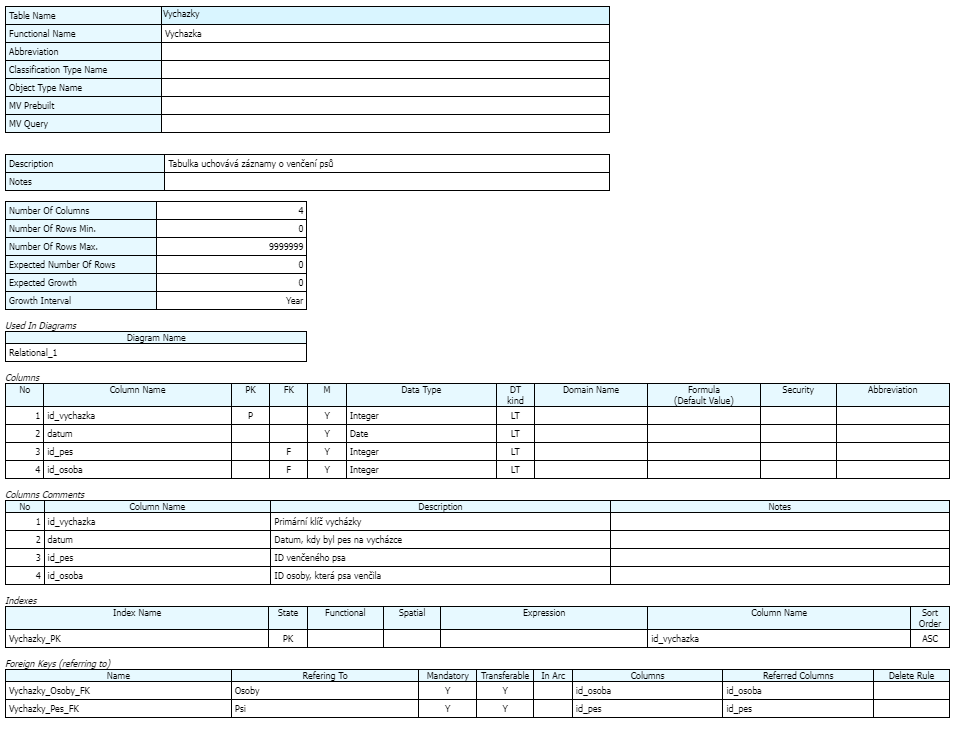












Výčet SQL dotazů

A1 Seznam psů, kteří byli odloženi Petrem Novým HK:D1 VK:D6  
SELECT p.id\_pes, p.jmeno  
FROM psi p  
JOIN odlozeni d  
ON p.id\_pes = d.id\_pes  
JOIN osoby o  
ON d.id\_osoba = o.id\_osoba  
WHERE o.jmeno LIKE 'Petr' AND o.prijmeni LIKE 'Nový';

A2 Seznam psů, kteří nebyli odloženi Petrem Novým HK:D2 VK:D6  
SELECT p.id\_pes, p.jmeno  
FROM psi p  
JOIN odlozeni d  
ON p.id\_pes = d.id\_pes  
JOIN osoby o  
ON d.id\_osoba = o.id\_osoba  
WHERE o.jmeno NOT LIKE 'Petr' AND o.prijmeni NOT LIKE 'Nový';

A3 Seznam psů, kteří podstoupili procedury pouze v klinice s id 2 HK:D3 VK:D17  
SELECT DISTINCT p.id\_pes, p.jmeno  
FROM psi p  
JOIN procedury r  
ON p.id\_pes = r.id\_pes  
MINUS (SELECT DISTINCT p.id\_pes, p.jmeno  
FROM psi p  
JOIN procedury r  
ON p.id\_pes = r.id\_pes  
WHERE id\_klinika = 1 OR id\_klinika = 2);

A4 Seznam psů, kteří byli oštřováni ve všech klinikách HK:D4 VK:D6  
SELECT DISTINCT p.id\_pes, p.jmeno  
FROM psi p  
JOIN procedury r   
ON p.id\_pes = r.id\_pes  
WHERE id\_klinika = 1 AND id\_klinika = 2 AND id\_klinika = 3;

A5 Seznam osob, které vlastní kliniky a adresy klinik HK:5  
SELECT id\_osoba, jmeno, prijmeni, veterinarnikliniky.adresa  
FROM osoby  
JOIN veterinarnikliniky USING (id\_osoba);

A6 Seznam klinik a jejich spojených licencí HK:D6  
SELECT k.id\_klinika, k.adresa, l.id\_licence, l.datum\_do  
FROM veterinarnikliniky k  
JOIN licence l  
ON l.id\_klinika = k.id\_klinika;

A7 Seznam všech vycházek a jejich spojených psů HK:D7  
SELECT \*  
FROM psi  
NATURAL JOIN vychazky;

A8 Cross join psů a vycházek HK:D8  
SELECT \*  
FROM psi  
CROSS JOIN vychazky;

A9 Seznam všech osob včetně jejich přiřazených klinik HK:D9  
SELECT o.id\_osoba, o.jmeno, o.prijmeni, NVL(k.adresa, 'Žádná klinika')  
FROM osoby o  
LEFT OUTER JOIN veterinarnikliniky k ON k.id\_osoba = o.id\_osoba;

A10 Seznam osob, které vlastní kliniku HK:D10  
SELECT o.id\_osoba, o.jmeno, o.prijmeni, NVL(k.adresa, 'Žádná klinika')  
FROM osoby o  
RIGHT OUTER JOIN veterinarnikliniky k ON k.id\_osoba = o.id\_osoba;

A11 Seznam psů a jejich chování HK:D11  
SELECT p.id\_pes, p.jmeno, i.id\_incident, NVL(i.popis, 'Nic neprovedl')  
FROM psi p  
FULL OUTER JOIN incidenty i  
ON p.id\_pes = i.id\_pes;

A12 Seznam darů, kde byl darovaný obnos vyšší než průměr HK:D12  
SELECT id\_dar, obnos, ostatni  
FROM dary  
WHERE obnos > (SELECT AVG(obnos) FROM dary);

A13 Seznam nalezených psů HK:D13 VK:D6  
SELECT id\_pes, jmeno  
FROM (SELECT psi.id\_pes, jmeno, misto FROM psi  
JOIN nalezeni ON psi.id\_pes = nalezeni.id\_pes);

A14 Pocty všech kategorií psů v databázi HK:D14 VK:D22  
SELECT DISTINCT (SELECT COUNT(\*) FROM nalezeni) AS Pocet\_nalezenych,  
(SELECT COUNT (\*) FROM odchyceni) AS Pocet\_odchycenych,  
(SELECT COUNT (\*) FROM odlozeni) AS Pocet\_odlozenych  
FROM psi;

A15 Seznam osob, které nedarovaly peníze, ale jiné věci HK:D15 VK:D6  
SELECT DISTINCT o.jmeno, o.prijmeni  
FROM osoby o  
JOIN dary d ON o.id\_osoba = d.id\_osoba  
WHERE EXISTS (SELECT id\_osoba FROM dary WHERE d.obnos IS NULL);

A16 Seznam všech adres z databáze a id osob k nim přiřazených HK:D16  
SELECT adresa, id\_osoba FROM osoby  
UNION  
SELECT adresa, id\_osoba FROM veterinarnikliniky  
ORDER BY ADRESA;

A17 Seznam psů, kteří jsou rezervováni a nebyli ještě venčeni HK:D17 VK:D6  
SELECT p.id\_pes, p.jmeno  
FROM psi p  
JOIN rezervace r   
ON p.id\_pes = r.id\_pes  
MINUS (SELECT DISTINCT p.id\_pes, p.jmeno  
FROM psi p  
JOIN vychazky v ON p.id\_pes = v.id\_pes);

A18 Seznam psů, kteří způsobili alespoň jeden incident a jsou po smrti HK:D18 VK:D6  
SELECT psi.id\_pes, psi.jmeno  
FROM psi  
JOIN incidenty  
ON psi.id\_pes = incidenty.id\_pes  
INTERSECT  
SELECT psi.id\_pes, psi.jmeno  
FROM psi  
JOIN umrti  
ON psi.id\_pes = umrti.id\_pes;

A19 Seznam psů, kteří mají jméno delší jak 5 písmen HK:D19  
SELECT \*  
FROM psi  
WHERE LENGTH(jmeno) > 5;

A20 Rozdíl součtu všech finančních darů a cen provedených procedur HK:D20 VK:D22  
SELECT ABS(x - y) FROM  
(SELECT SUM(dary.obnos) AS x FROM dary),  
(SELECT SUM(procedury.cena) AS y FROM procedury);

A21 Seznam psů, kteří byli nalezeni mezi roky 2010 a 2015 HK:D21 VK:D6  
SELECT p.id\_pes, p.jmeno, n.datum  
FROM psi p  
JOIN nalezeni n ON p.id\_pes = n.id\_pes  
WHERE n.datum <= TO\_DATE('1.1.2015', 'dd-mm-yyyy') AND n.datum >= TO\_DATE('1.1.2010', 'dd-mm-yyyy');

A22 Pocčet darů, minimální finanční dar, maximální finanční dar, součet všech finančních darů a průměrný finanční dar HK:D22  
SELECT COUNT(\*), MIN(dary.obnos), MAX(dary.obnos), SUM(dary.obnos), AVG(dary.obnos)  
FROM dary;

A23 Seznam osob, které venčily psi a kolikrát HK:D23  
SELECT o.id\_osoba, o.jmeno, o.prijmeni, COUNT(\*)  
FROM osoby o  
JOIN vychazky v ON o.id\_osoba = v.id\_osoba  
GROUP BY o.id\_osoba, o.jmeno, o.prijmeni;

A24 Seznam osob, které do útulku neuložily psa HK:D24 VK:D9, D17, D12  
SELECT DISTINCT o.id\_osoba, o.jmeno, o.prijmeni  
FROM osoby o  
LEFT JOIN odlozeni d ON o.id\_osoba = d.id\_osoba  
WHERE d.id\_osoba IS NULL;

SELECT DISTINCT o.id\_osoba, o.jmeno, o.prijmeni  
FROM osoby o  
MINUS(SELECT DISTINCT o.id\_osoba, o.jmeno, o.prijmeni  
FROM osoby o  
LEFT JOIN odlozeni d ON o.id\_osoba = d.id\_osoba  
WHERE d.id\_osoba IS NOT NULL);

SELECT DISTINCT o.id\_osoba, o.jmeno, o.prijmeni  
FROM osoby o  
WHERE o.id\_osoba NOT IN (SELECT d.id\_osoba FROM odlozeni d);

A25 Seznam psů a počet podstoupených procedur nad 4 HK:D25  
SELECT p.id\_pes, p.jmeno, COUNT(\*)  
FROM psi p  
JOIN procedury r ON p.id\_pes = r.id\_pes  
WHERE p.id\_plemeno = 7  
GROUP BY p.id\_pes, p.jmeno  
HAVING COUNT(\*) > 3  
ORDER BY COUNT(\*) DESC;

A26 Seznam všech prohlídek s se všemi psy, adresami a jmén správců klinik HK:D26 VK:D6  
CREATE OR REPLACE VIEW moje\_view AS  
SELECT k.id\_klinika AS id\_kliniky, k.adresa AS adresa\_kliniky, o.jmeno || ' ' || o.prijmeni AS jmeno\_spravce, d.popis AS popis\_procedury, d.cena AS cena\_procedury, p.jmeno AS jmeno\_psa FROM veterinarnikliniky k  
JOIN osoby o ON k.id\_osoba = o.id\_osoba  
JOIN procedury d ON k.id\_klinika = d.id\_klinika  
JOIN psi p ON d.id\_pes = p.id\_pes;

A27 Výpis seznamu z předchozího view, s podmínkou ceny nad 1000 Kč HK:27  
SELECT \* FROM moje\_view  
WHERE cena\_procedury > 1000;

A28 Vložení rezervace pro psa neznámý HK:D28  
INSERT INTO rezervace(id\_rezervace, datum, id\_pes, id\_osoba)  
SELECT p.id\_pes, CURRENT\_DATE, p.id\_pes, 2  
FROM psi p  
Where p.jmeno LIKE 'Neznámý';

A29 Změna jména psa, který se jmenuje Brok HK:D29  
UPDATE psi SET jmeno = 'Neznámý'  
WHERE id\_pes = (SELECT id\_pes FROM psi WHERE jmeno LIKE 'Brok');

A30 Smazání psa, který se jmenuje Odstranit HK:D30  
DELETE psi WHERE id\_pes = (SELECT id\_pes FROM psi WHERE jmeno LIKE 'Odstranit');

Závěr

Na závěr bych si dovolil říci, že má databáze psího útulku je vcelku dobře navržená. Jsem si vědom toho, že nemusí být dokonale implementována, což lze vylepšit časem. Díky tomuto projektu jsem zjistil jak se databáze navrhují, co závisí na analýze požadavků a jak tyto poznatky formují celou databázi. Tento projekt měl vcelku široký záběr, co se týče praktických věcí a jsem zvědavý co přinese projekt v předmětu na tento navazující.

Přílohy

1. DDL skript pro vytvoření tabulek, omezení, sekvencí, indexů, triggerů, apod.
2. DML skript pro naplnění tabulek daty.
3. Soubor dmd včetně složky (Datový model musí souhlasit s DDL skriptem)
4. Skript se všemi dotazy.