

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Dokumentácia popisujúca finálnu schému databázy

Veterinárna klinika

1. mája 2018

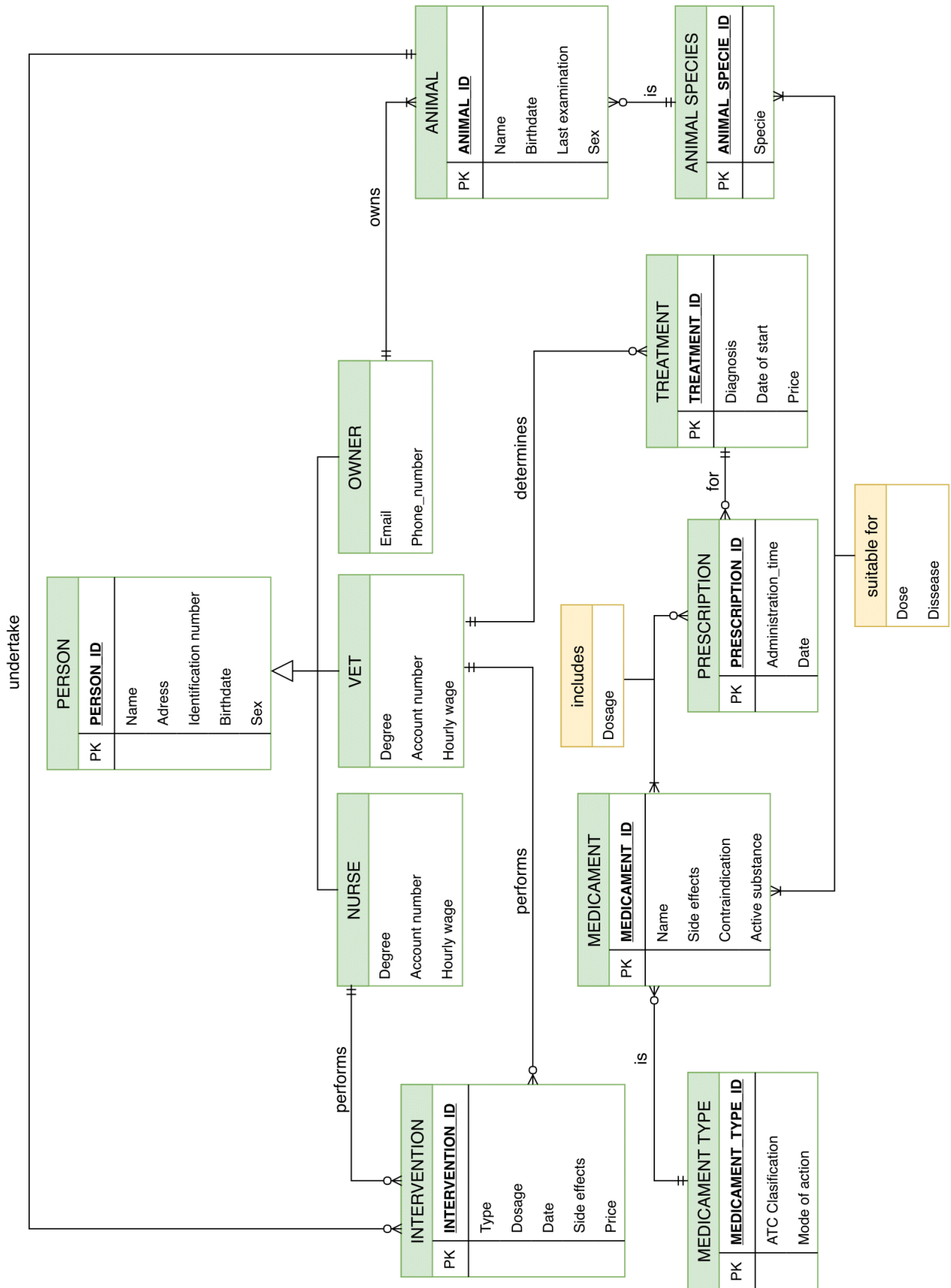
Matúš Liščinský
Vladimír Marcin

(xlisci02)
(xmarci10)

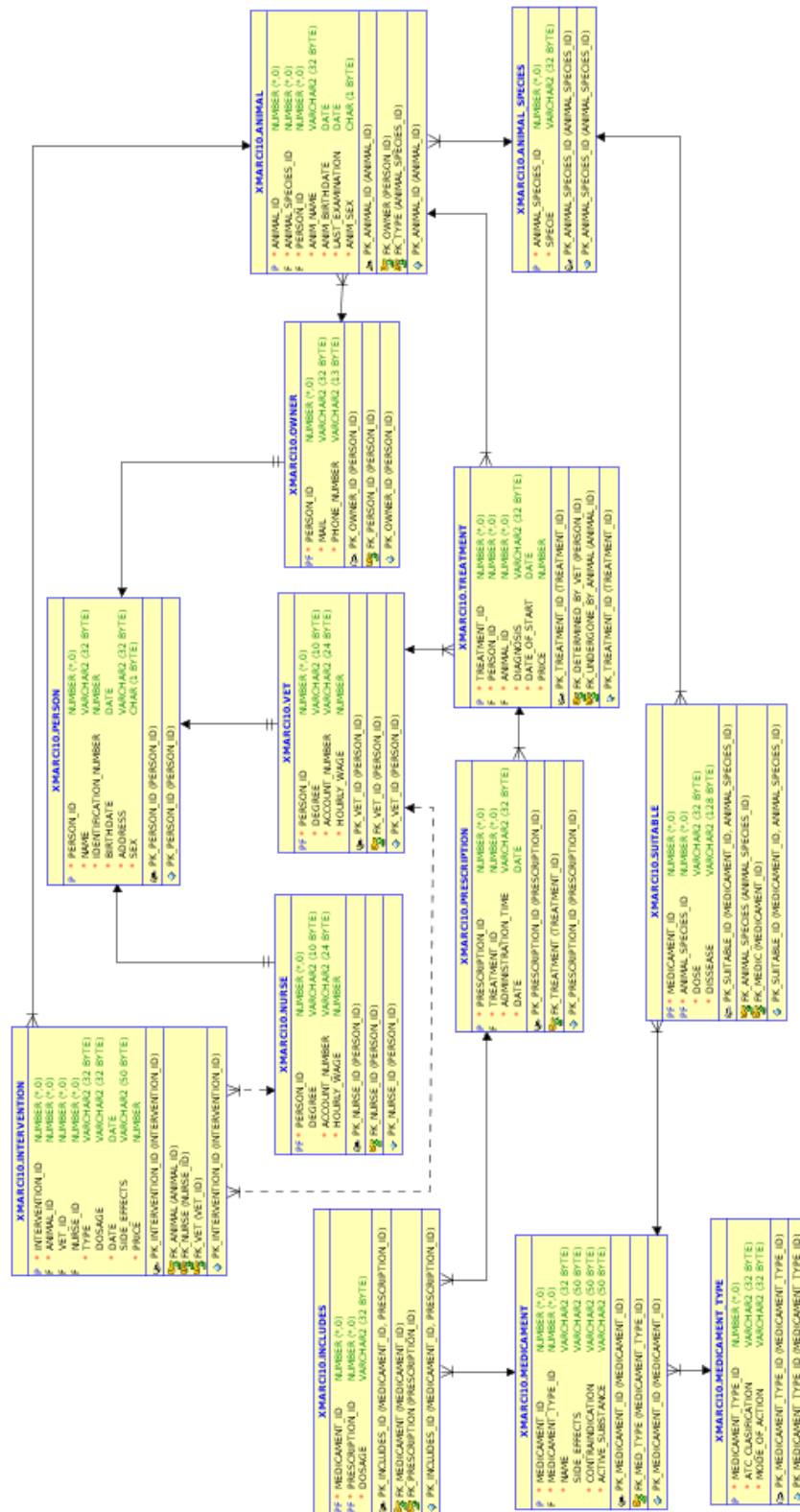
Obsah

1	ER Diagram	2
2	Schéma relačnej databázy	3
3	Generalizácia / Špecializácia	4
4	SQL skript	4
5	Triggery	4
6	Procedúry	4
7	Pridelenie práv	5
8	Explain plan a index	6
9	Materializovaný pohľad	7

1 ER Diagram



2 Schéma relačnej databázy



3 Generalizácia / Špecializácia

Nami vytvorený datový model databázy obsahuje jeden vzťah generalizácie / špecializácie. Entitná množina *PERSON* má tri špecializácie: *VET*, *NURSE* a *OWNER*. Vybrali sme možnosť vytvorenia tabuľky pre nadtyp (*PERSON*) a tabuľky podtypov s primárnym kľúčom nadtypu aj kvôli možnosti, aby aj zamestanci kliniky mohli byť v databáze vedený ako majitelia zvierat bez duplicity údajov.

4 SQL skript

Na začiatku skript zmaže všetky databázové objekty kvôli prípadným konfliktom pri opätovnom spustení, vytvorí tabuľky, nastaví primárne, cudzie kľúče, naplní tabuľky dátami a vykoná viacero ukázkových dotazov **SELECT** podľa zadania. V ďalšej časti vytvorí niekoľko databázových triggerov a zavolá vytvorené procedúry pre demonštráciu ich činnosti. Pomocou **EXPLAIN PLAN** sa uskutočňuje výpis plánu vykonania databázového dotazu a pre optimalizáciu spracovania dotazu sa používa index. Nakoniec sa pridelia práva aj druhému užívateľovi a na jeho strane sa vytvorí materializovaný pohľad.

5 Triggery

Z celkovo troch implementovaných triggerov je práve jeden trigger pre automatické generovanie hotnôt primárneho kľúča tabuľky *ANIMAL*. Ak pri vkladaní dát o novom zvierati do tabuľky chýba hodnota primárneho kľúča, teda *ANIMAL_ID*, dosadené bude automaticky nasledujúce číslo zo sekvencie.

Kedže databáza uchováva údaje aj o číslach účtov, ďalší z triggerov kontroluje správnosť čísla účtu vo forme IBAN. Správny IBAN musí obsahovať konkaténáciu dvoch písmen identifikujúcich krajinu a 22 číslíc, reprezentujúcich číslo deliteľné 11 (použitá funkcia **MOD**). V našom prípade je obmedzený na České a Slovenské predpony.

Posledným z nami implementovaných triggerov slúži na kontrolu rodného čísla. Trigger okrem počtu číslic rodného čísla kontroluje aj jeho deliteľnosť číslom 11, dátum (deň, mesiac, rok) a prírastok mesiaca v prípade ženského pohlavia a vyčerpania kombinácií pre daný deň.

6 Procedúry

Vrámcí projektu sme implementovali aj tri procedúry. Tie obsahujú okrem iného aj ošetrovanie výnimok, kurzory a premenné s dátovým typom odkazujúcim sa na riadok (**ROWTYPE**).

Procedúra *indisposition_of_animal_species* (chorľavosť druhov zvierat) podľa id druhu zvierat zistí a vypíše percentuálny pomer všetkých chorôb a chorôb, na ktoré už bol liečený tento druh.

Tabuľka druhov zvierat a chorôb ktorými trpel:

	ANIMAL_SPECIES_ID	SPECIE	DIAGNOSIS
1	1	Pes	Bolest klbov
2	2	Macka	Hnacka
3	2	Macka	Zapal ucha
4	3	Zajac	Blchy
5	3	Zajac	Hnacka
6	3	Zajac	Zapal ustnej dutiny

Použitie príkazu: `exec indisposition_of_animal_species(2);`

Script Output

Tento druh uz trpel 40% vsetkych doposial evidovanych chorob

Procedúra `owners_expenses_for_treatments` vyžaduje jeden parameter (`OWNER_ID`) a slúži na zistenie informácie, koľko peňazí daný majiteľ celkovo zaplatil za uskutočnené liečby všetkých svojich zvierat.

Tabuľka majiteľov a cien ktore zaplatili za jednotlivé liečby:

	PERSON_ID	NAME	PRICE
1	1	Jakub Sopka	9,8
2	1	Jakub Sopka	74,9
3	1	Jakub Sopka	9,9
4	3	Marian Bazinga	9,5
5	5	Rudolf Stramak	17,45
6	7	Peter Velonos	25
7	8	Katka Sitniakova	74,5

Použitie príkazu: `exec owners_expenses_for_treatments(1);`

Script Output

Majiteľ už minul 94,6 eur na liečby svojich zvierat.

Procedúra `average_price_of_intervention` poskytuje informáciu, aká bola priemerná cena zákroku vykonaného zamestnancom, ktorého id bolo predané parametrom tejto procedúry.

Tabuľka zamestnancov a nimi vykonaných zákrokov s cenami zákrokov.

	VET_ID	NURSE_ID	TYPE	PRICE
1	6	(null)	EKG	5,5
2	(null)	4	Odber krvi a spracovanie vzorky	2,6
3	6	(null)	EKG	6,5
4	(null)	2	Anestezia inhalacna	15,9
5	(null)	4	Odber krvi a spracovanie vzorky	2,6
6	1	4	Punkcia močového mechura	60,2

Použitie príkazu: `exec average_price_of_intervention(4);`

Script Output

Zamestnanec s id 4 priemerne zinkasoval 21,8 eur za zákroky na ktorých sa podieľal.

7 Pridelenie práv

Súčasťou projektu bola aj definícia prístupových práv k databázovým objektom pre druhého člena tímu. Pridelené práva sú na bežnej úrovni sestry, tá má plný prístup k tabuľkám týkajúcim sa majiteľov, zvierat, zákrokov a podobne, avšak na tabuľku liečby má obmedzené práva iba na *SELECT*.

8 Explain plan a index

Klauzula **EXPLAIN PLAN** sa používa pre zobrazenie plánu, ako bude dotaz spracovávaný databázou. Použili sme dotaz *SELECT* zobrazujúci meno vlastníka a počet zvierat ktoré vlastní. Pre tento dotaz vyzerá EXPLAIN PLAN nasledovne:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		6	264	4 (25)	00:00:01
1	HASH GROUP BY		6	264	4 (25)	00:00:01
2	NESTED LOOPS		6	264	3 (0)	00:00:01
3	NESTED LOOPS		6	264	3 (0)	00:00:01
4	TABLE ACCESS FULL	ANIMAL	6	78	3 (0)	00:00:01
* 5	INDEX UNIQUE SCAN	PK_PERSON_ID	1		0 (0)	00:00:01

PLAN_TABLE_OUTPUT

6	TABLE ACCESS BY INDEX ROWID	PERSON	1	31	0 (0)	00:00:01
---	-----------------------------	--------	---	----	-------	----------

Analýza získaného plánu:

Na nultom riadku sa nachádza operácia *SELECT STATEMENT*, ktorá reprezentuje náš *SELECT* dotaz. *HASH GROUP BY* znamená zoskupovanie položiek pomocou kľúča z hashovacej funkcie. Operácie *NESTED LOOPS* značia prehl'adávanie spájaných tabuliek vo vnorených cykloch. Nasleduje prechod celou tabuľkou - *TABLE ACCESS FULL* a akcia *INDEX UNIQUE SCAN*, čiže prístup k tabuľkám cez B-strom, za účelom získania unikátneho riadku podľa primárneho kľúča *PERSON_ID*.

Po vytvorení indexu na stĺpec *PERSON_ID* v tabuľke *ANIMAL*, t.j stĺpec spoločný pre obe tabuľky používaný pri spájaní pomocou *NATURAL JOIN* vyzerá EXPLAIN PLAN nasledovne:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		6	264	2 (50)	00:00:01
1	HASH GROUP BY		6	264	2 (50)	00:00:01
2	NESTED LOOPS		6	264	1 (0)	00:00:01
3	NESTED LOOPS		6	264	1 (0)	00:00:01
4	INDEX FULL SCAN	INDEXEXPLAIN	6	78	1 (0)	00:00:01
* 5	INDEX UNIQUE SCAN	PK_PERSON_ID	1		0 (0)	00:00:01

PLAN_TABLE_OUTPUT

6	TABLE ACCESS BY INDEX ROWID	PERSON	1	31	0 (0)	00:00:01
---	-----------------------------	--------	---	----	-------	----------

Ako môžeme vidieť, databáza používa vytvorený index *INDEXEXPLAIN* pre prístup k hodnotám z indexovaného stĺpca. *INDEX FULL SCAN* znamená, že prechádza všetky položky z indexu. Vytvorením indexu sa znížila cena (cost) o polovicu.

9 Materializovaný pohľad

Druhý člen tímu vytvára materializovaný pohľad používajúci tabuľky prvého člena tímu a pridelí mu prístupové práva. Ide o materializovaný pohľad na tabuľku obsahujúcu zvieratá a informácie o podstupených zákrokoch na klinike. Prvý člen tímu vytvára LOGY nad týmito tabuľkami ANIMAL a INTERVENTION v ktorých sa budú zaznamenávať prípadné zmeny v tabuľkách.

Pri nasledovnom vkladaní do tabuľky sa zaznamená táto informácia do logu a keďže používame *REFRESH FAST ON COMMIT*, pri aktualizácii pohľadu sa nebude vykonávať znova dotaz SELECT a načítavať všetky dáta z tabuľky ale použijú sa informácie z logu a aplikujú sa na dáta pohľadu. Na to aby bol pohľad použitý pre optimalizáciu rovnakého dotazu je potrebné nastaviť mu vlastnosť *ENABLE QUERY REWRITE*.

Použitím **EXPLAIN PLAN** sme získali plán operácií, ako bude dotaz vykonávaný databázou.

Id	Operation	Name
0	SELECT STATEMENT	
1	NESTED LOOPS	
2	TABLE ACCESS BY INDEX ROWID	ANIMAL
3	INDEX UNIQUE SCAN	PK_ANIMAL_ID
4	TABLE ACCESS FULL	INTERVENTION

Pomocou príkazu `ALTER SESSION SET query_rewrite_enabled = TRUE;` sme povolili spomínanú vlastnosť *QUERY REWRITE* materializovaného pohľadu a opäť spustili EXPLAIN PLAN nad rovnakým dotazom. Ako vidieť na obrázku, databáza použila tento pohľad pri obsluhu dotazu.

Id	Operation	Name
0	SELECT STATEMENT	
1	MAT_VIEW REWRITE ACCESS FULL	ANIMAL_INTERVENTION

Materializovaný pohľad a tabuľka INTERVENTION **pred commitom**:

ANIM_NAME	TYPE	DATE	ANIMAL_ROW_ID	INTERVENTION_ROW_ID
Mandu	Anestezia inhalacna	23.01.18	AAAUwAAWAAABxGAAD	AAAUwxAAWAAASOAAAD
Mandu	Odber krvi a spracovanie vzorky	02.08.17	AAAUwAAWAAABxGAAD	AAAUwxAAWAAASOAAE
Mandu	Punkcia mocoveho mechura	02.08.17	AAAUwAAWAAABxGAAD	AAAUwxAAWAAASOAAF

INTERVENTION_ID	ANIMAL_ID	VET_ID	NURSE_ID	TYPE	DOSAGE	DATE	SIDE_EFFECTS
1	1	6		EKG		19.02.18	
2	3		4	Odber krvi a spracovanie vzorky	50ml	23.01.18	Nevoľnosť
3	2	6		EKG		23.01.18	
4	4		2	Anestezia inhalacna	25ml	23.01.18	
5	4		4	Odber krvi a spracovanie vzorky	25ml	02.08.17	Nevoľnosť
6	4	1	4	Punkcia mocoveho mechura		02.08.17	

6 rows selected.

1 row inserted.

Commit complete.

Vložení nových údajov pomocou:

```
INSERT INTO xmarci10.INTERVENTION (ANIMAL_ID, VET_ID, "TYPE", "DATE", PRICE)
VALUES (4, 6, 'EKG', TO_DATE('2018-02-21', 'yyyy-mm-dd'), 5.50);
```

do tabuľky *INTERVENTION*, sa modifikuje LOG nad touto tabuľkou a pri následnom commite sa aktualizujú dáta pohľadu. Túto zmenu zobrazuje nasledujúci výpis z materializovaného pohľadu a tabuľky.

ANIM_NAME	TYPE	DATE	ANIMAL_ROW_ID	INTERVENTION_ROW_ID	
Mandu	Anestezia inhalacna	23.01.18	AAAUwvAAWAAABxGAAD	AAAUwxAAWAAASOAAAD	
Mandu	Odber krvi a spracovanie vzorky	02.08.17	AAAUwvAAWAAABxGAAD	AAAUwxAAWAAASOAAE	
Mandu	Punkcia mocoveho mechura	02.08.17	AAAUwvAAWAAABxGAAD	AAAUwxAAWAAASOAAF	
Mandu	EKG	21.02.18	AAAUwvAAWAAABxGAAD	AAAUwxAAWAAASNAAA	

INTERVENTION_ID	ANIMAL_ID	VET_ID	NURSE_ID	TYPE	DOSAGE	DATE	SIDE_EFFECTS
7	4	6		EKG		21.02.18	
1	1	6		EKG		19.02.18	
2	3		4	Odber krvi a spracovanie vzorky	50ml	23.01.18	Nevoľnosť
3	2	6		EKG		23.01.18	
4	4		2	Anestezia inhalacna	25ml	23.01.18	
5	4		4	Odber krvi a spracovanie vzorky	25ml	02.08.17	Nevoľnosť
6	4	1	4	Punkcia mocoveho mechura		02.08.17	