IDS – projekt 1
ER diagram kočičího informačního systému

Autoři:

- David Hudák, xhudak03
- Štefan Mračna, xmracn00

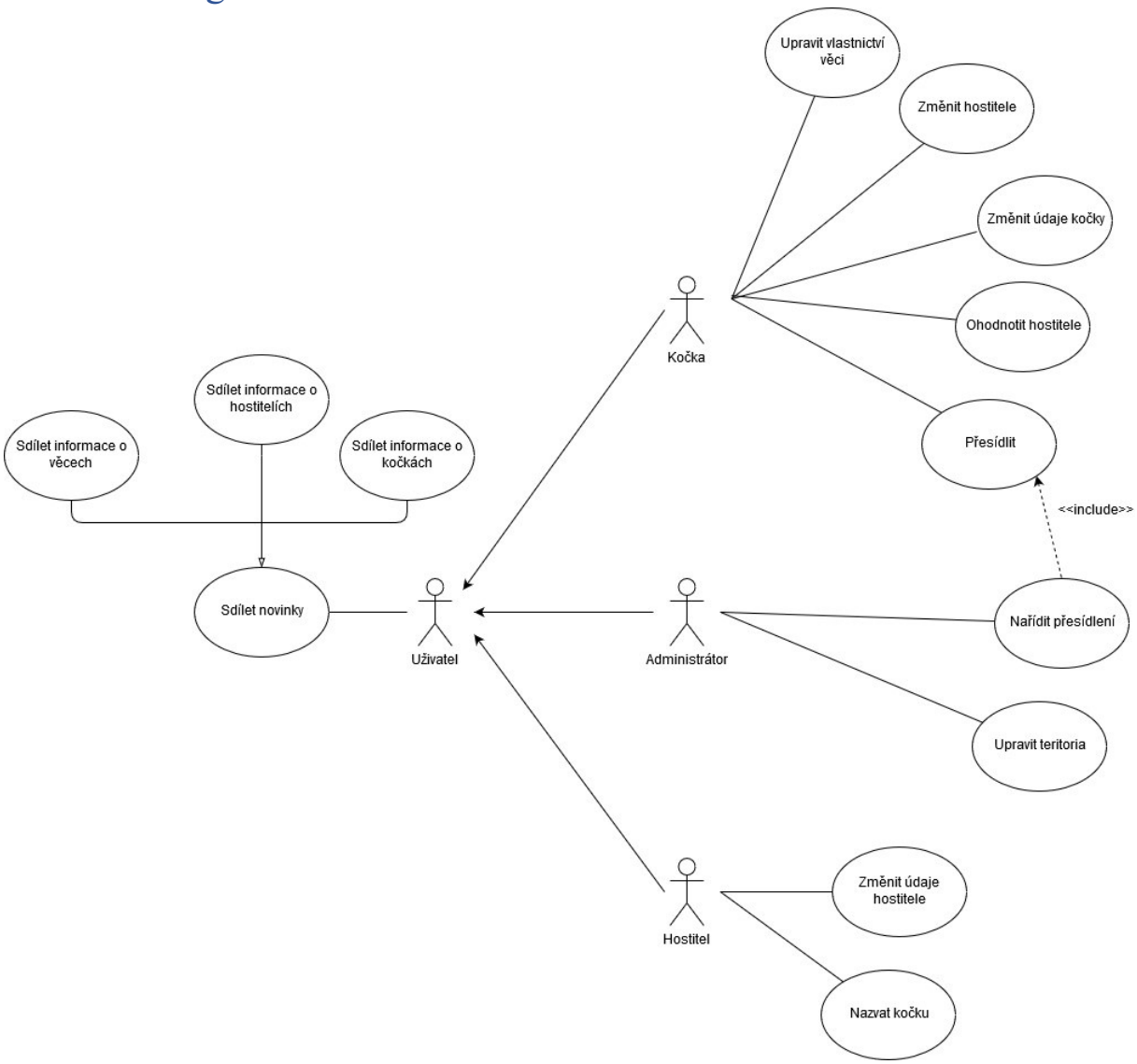
Zadání (doslovně převzato z předmětu IUS)

Kočky chtějí zefektivnit jejich dominanci lidského světa a proto Vám zadaly vytvořit KIS (Kitty Information System). Tento systém uchovává informace o jednotlivých rasách koček, jejich specifické rysy, jako možné barvy očí, původ, maximální délku tesáků, apod. a u konkrétních koček pak jejich hlavní jméno, vzorek kůže, barvu srsti a pod. Každá kočka má právě devět životů, nicméně v systému vedeme pouze ty, které již proběhly a aktuálně probíhají, a vedeme u nich informaci o délce života, místo narození a případně (u minulých životů) o místě (v rámci, kterého teritoria) a způsobu smrti. Kočky jsou samozřejmě majetnické a chtějí si vést všechny teritoria (máme teritoria různých typů, jako např. jídelna, klub – teritoria by se měla dále dělit na vnější a vnitřní, přičemž o vniřních teritoriích si vedeme záznam o počtu místností a u vnějších teritoriích si vedeme záznam o rozloze), ve kterých se kdy pohybovaly a které věci si přivlastnily a v kterém intervalu je vlastnily (kočky se lehce znudí a své věci prostě zahodí). Systém rovněž vede informace o jejich hostitelích, kteří jim slouží. Veďte u nich jejich základní informace (jméno, věk, pohlaví, místo bydlení…), které rasy koček preferují a rovněž jméno, kterým kočku nazývali (např. Pan Tlapoň, Bublina, Gaston, .). Některé vlastnictví koček však mohou být propůjčována svým hostitelům. Současně veďte informaci (pokud je přítomna) o teritoriu v rámci kterého se vlastnictví nachází, typ vlastnictví (hračka, cokoliv…) a jeho kvantitu. Jednotlivá teritoria však mají omezenou kapacitu na kočky a v případě překočení (doslova) se kočky přesídlí. Systém umožňuje kočkám zasílat pravidelné novinky o životech ostatních koček a nových dostupných hostitelích, ke kterým by se mohly přesídlit a věcech, které by mohly zabrat.

Dodatek k zadání

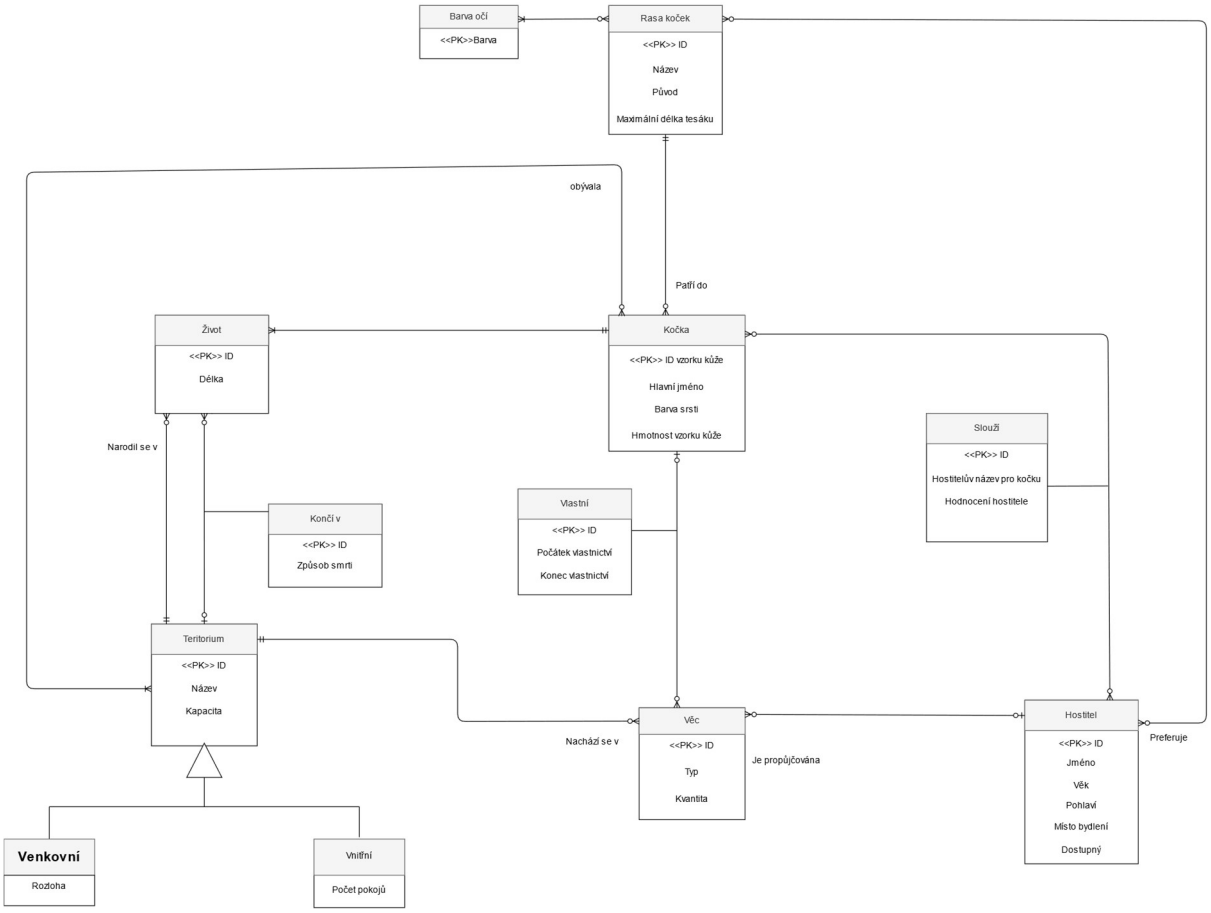
V rámci zadání předmětu IDS nastala povinnost mít alespoň jednu generalizaci/specializaci. Proto k zadání dodáváme, že kočka může mít buď venkovní teritorium, nebo vnitřní teritorium.

Use-case diagram



V use-case diagramu můžeme vidět, že systém má 4 (prakticky 3) typy uživatelů – kočku, administrátora a hostitele. Každý z nich se pak dá považovat také za uživatele. Uživatel obecně může sdílet novinky různých druhů. Hostitel má pak vyhrazené právo nazvat kočku a měnit své údaje. Kočka má na rozdíl od hostitele mnohem širší paletu možností, co může v systému dělat, tj. od hodnocení svého hostitele, po změnu teritoria. Administrátor pak má možnost nařídit změnu teritoria a případně je upravovat.

ER diagram



Jak bylo zmíněno v dodatku k zadání, oproti původnímu (IUS ER diagramu) byla přidána možnost typu teritoria. Jinak zde vidíme, že teritorium kočka osidluje narozením v něm a dále si může přivlastnit i další teritoria (osídlení teritoria pak může končit smrtí, o tom je nutné zanechat v systému informaci). Dále kočka může vlastnit nějakou věc (pouze však po dobu, kterou chce, přesto bychom nadále měli vést údaj o vlastnictví i po zániku). Hostitel pak slouží kočce, a to konkrétně takové, jejíž rasu preferuje. Za to si v systému může přiřadit k dané kočce jméno. Jako mýlka by se mohlo zdát, proč barva očí je vázána k rase a nikoliv kočce. Jedná se o požadavek na to, aby jednotlivé rasy měly uvedeny své typické barvy očí.

Procedury

V SQL skriptu se vyskytují celkově dvě procedury. První procedura zjišťuje, zdali věci vlastněné kočkami jsou postačující. Tedy provede se výpočet doby vlastnictví podělený jejich počtem. Pokud takový výpočet vychází jako větší než jedna (kočka má jednu věc na den a více), pak je situace v pořádku a vypíše se zpráva, že kočka má dostatečnou kvantitu věci. Pokud nastává opačná situace, věci je málo. V této proceduře je ještě ošetřena situace, kdy se v databázi nachází údaj o situaci, že kvantita věci je nula (to znamená, že kočka nejspíš věc zničila, nebo ji nikdy nevlastinla). Druhá procedura vypočítává průměrné stáří hostitele.

Triggery

Ve skriptu se nachází dva triggery. První z nich se spouští před přidáním položky do tabulky a pro každý řádek kontroluje, zdali přidávaná rasa kočky nemá příliš velké zuby. Experimentálním měřením bylo zjištěno, že naprostým maximem je 300 jednotek, tudíž je jakákoliv přesahující jednotka přesahující 300 oříznuta na 300, jelikož při měření takové hodnoty nejspíše došlo k chybě. Druhý trigger slouží k výpočtu indexu hostitele na základě vstupní sekvence.

Přístupová práva

Z praktických důvodů bylo poskytnuto přistupovat do databáze v kompletním rozsahu pro druhého člena (navíc byl udělen přístup do materializovaného pohledu).

Materializovaný pohled

Materializovaný pohled určený pro druhého uživatele obsahuje pouze pohled na tabulky kočky a rasy koček, z nichž bere pouze údaj o hlavním jménu kočky a název rasy. Jedná se o dvě nejzásadnější informace, které se o kočce dají říct, z toho důvodu byly vybrány zrovna tyto.

Explain plan

V následujícím výpisu je ukázka neoptimalizovaného příkazu select, který již byl vytvořen v předchozí části:

```
PLAN_TABLE_OUTPUT
Plan hash value: 2909830679

-------------------------------------------------------------------------------------
| Id  | Operation            | Name       | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT     |            |     8 |   936 |    10  (10)| 00:00:01 |
|   1 |  HASH GROUP BY       |            |     8 |   936 |    10  (10)| 00:00:01 |
|*  2 |   HASH JOIN          |            |     8 |   936 |     9   (0)| 00:00:01 |
|*  3 |    HASH JOIN         |            |     8 |   416 |     6   (0)| 00:00:01 |
|   4 |     TABLE ACCESS FULL| RASA_KOČEK |     5 |   130 |     3   (0)| 00:00:01 |
|   5 |     TABLE ACCESS FULL| KOČKA      |     8 |   208 |     3   (0)| 00:00:01 |
|   6 |    TABLE ACCESS FULL | TERITORIUM |     6 |   390 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("K"."KOČKA_TERITORIUM"="T"."ID")
   3 - access("K"."KOČKA_RASA"="R"."ID")

Note
-----
   - dynamic statistics used: dynamic sampling (level=2)
   - this is an adaptive plan
```

Tento explain plan pak byl prováděn na příkazu select, který získává informace ze tří různých tabulek a využívá pouze k tomu, aby získal informaci o názvu teritoria a součtu maximálních délek tesáků ras, které v tomto teritoriu žijí. Příkaz viz zde:

```
SELECT T.název, sum(MAXIMÁLNÍ_DÉLKA_TESÁKU) max_delka_tesaku
 FROM KOČKA K, RASA_KOČEK R, TERITORIUM T
 WHERE K.kočka_rasa = R.id and K.kočka_teritorium = T.id
 GROUP BY T.název;
```

Zcela zásadní informací je, že z tabulek potřebujeme opravdu minimum informací (hmotnost kožichu, různé barvy, jména koček, délky života koček atd. jsou zbytečné) a tudíž se vyplatí vytvořit pro všechny tři tabulky indexy. Tyto indexy pak obsahují pouze to nejzákladnější, tedy index a název z teritoria, čísla cizích klíčů z tabulky kočky a index s maximální délkou tesáků z rasy kočky. Výslednou optimalizaci můžeme vidět na následujícím výpisu:

```
PLAN_TABLE_OUTPUT
Plan hash value: 3491045115

-------------------------------------------------------------------------------------------
| Id  | Operation          | Name            | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |                 |     8 |   936 |     4  (25)| 00:00:01 |
|   1 |  HASH GROUP BY     |                 |     8 |   936 |     4  (25)| 00:00:01 |
|*  2 |   HASH JOIN        |                 |     8 |   936 |     3   (0)| 00:00:01 |
|*  3 |    HASH JOIN       |                 |     8 |   416 |     2   (0)| 00:00:01 |
|   4 |     INDEX FULL SCAN| RASOINDEX       |     5 |   130 |     1   (0)| 00:00:01 |
|   5 |     INDEX FULL SCAN| KOČKOINDEX      |     8 |   208 |     1   (0)| 00:00:01 |
|   6 |    INDEX FULL SCAN | TERITORIUMINXED |     6 |   390 |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("K"."KOČKA_TERITORIUM"="T"."ID")
   3 - access("K"."KOČKA_RASA"="R"."ID")

Note
-----
   - dynamic statistics used: dynamic sampling (level=2)
   - this is an adaptive plan
```

Z uvedené optimalizace můžeme vidět, že bylo uvedené jednotky ceny procesoru byly sníženy průměrně na třetinu, což je poměrně velká úspora. Je to především díky tomu, že se nemusely procházet celé tabulky, ale jenom jejich indexované podčásti. Důležitou nevýhodou, kterou je třeba neopomenout, zůstává nemalá paměťová náročnost, která sice není ukázána na tomto výpisu, avšak je zcela očekávatelná vzhledem ke vzniku znovu prakticky celých tabulek pro optimalizaci. Proto je nejspíše vhodné dané optimalizace zvážit a z efektivního hlediska je nejspíše nejvhodnější indexovat samotnou kočku, jelikož obsahuje nejvíce údajů a současně z ní jsou žádány pouze cizí klíče (propojení tabulky rasy a teritoria).