

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

Databázové systémy

2016/2017

Zadání - Svět magie

Kouzelnický svět vytváří informační systém pro evidenci kouzel a kouzelníků. Magie v kouzelnickém světě je členěna podle elementů (např. voda, oheň, vzduch,.), které mají různé specializace (obrana, útok) a různé, ale pevně dané barvy magie (např. ohnivá magie je pomerančově oranžová). Každé kouzlo má jeden hlavní element , přičemž každý kouzelník má pozitivní synergii s určitými elementy. U kouzelníků rovněž evidujeme velikost many, jeho dosaženou úroveň v kouzlení (předpokládáme klasickou stupnici. E, D, C, B, A, S, SS.). U jednotlivých kouzel pak jejich úroveň složitosti seslání a sílu. Kouzla mohou být útočná či obranná, přičemž obranná mohou mít navíc i vedlejší elementy. Kouzla mohou být samovolně sesílána, pokud je daný kouzelník umí, nebo lze využít kouzelnických knih, tzv. grimoárů.

Grimoáry v sobě seskupují různá kouzla. Uchováváme veškerou historii jejich vlastnictví. Grimoáry mohou obsahovat kouzla různých elementů, nicméně jeden z elementů je pro ně primární. Využívání grimoárů způsobuje ztrátu jejich nabitě magie a je nutno je znovu dobít, ale pouze na dedikovaných místech, kde prosakuje magie (míra prosakování magie daného místa je evidována) určitých typů element (předpokládejte, že na daném místě prosakuje právě jeden typ). Toto nabití však nemusí být provedeno vlastníkem, ale i jiným kouzelníkem a zároveň musí být provedeno první den v novém měsíci od půlnoci do tří hodin do rána.

Alternativním způsobem seslání magie je pak s využitím svítek, který obsahuje právě jedno útočné kouzlo a po jeho použití se rozpadne.

ER Diagram

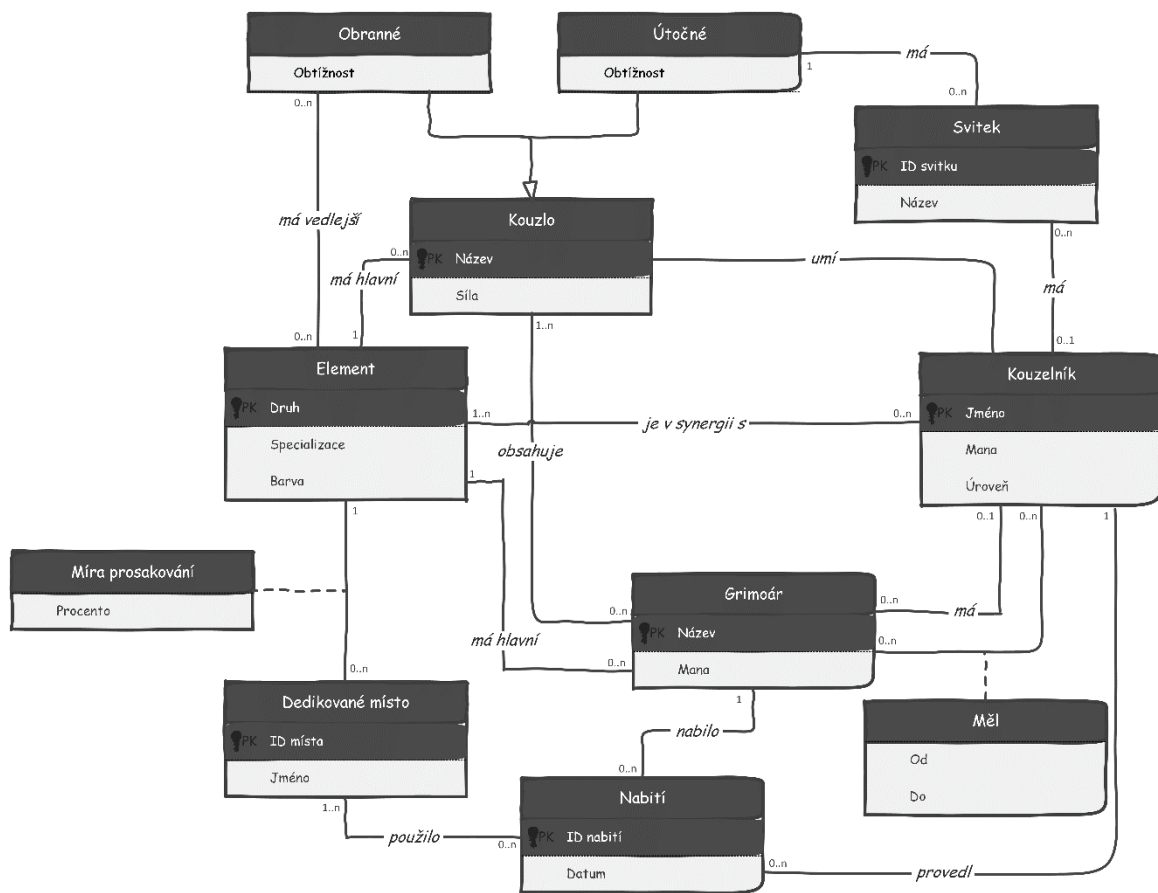
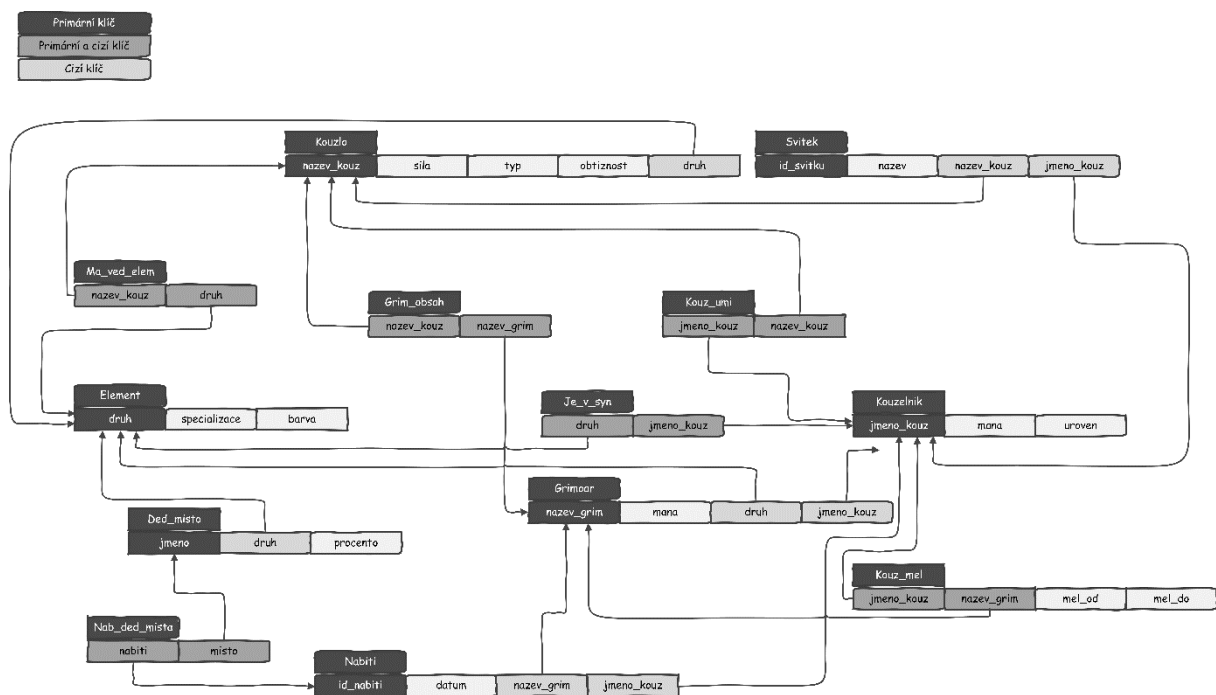
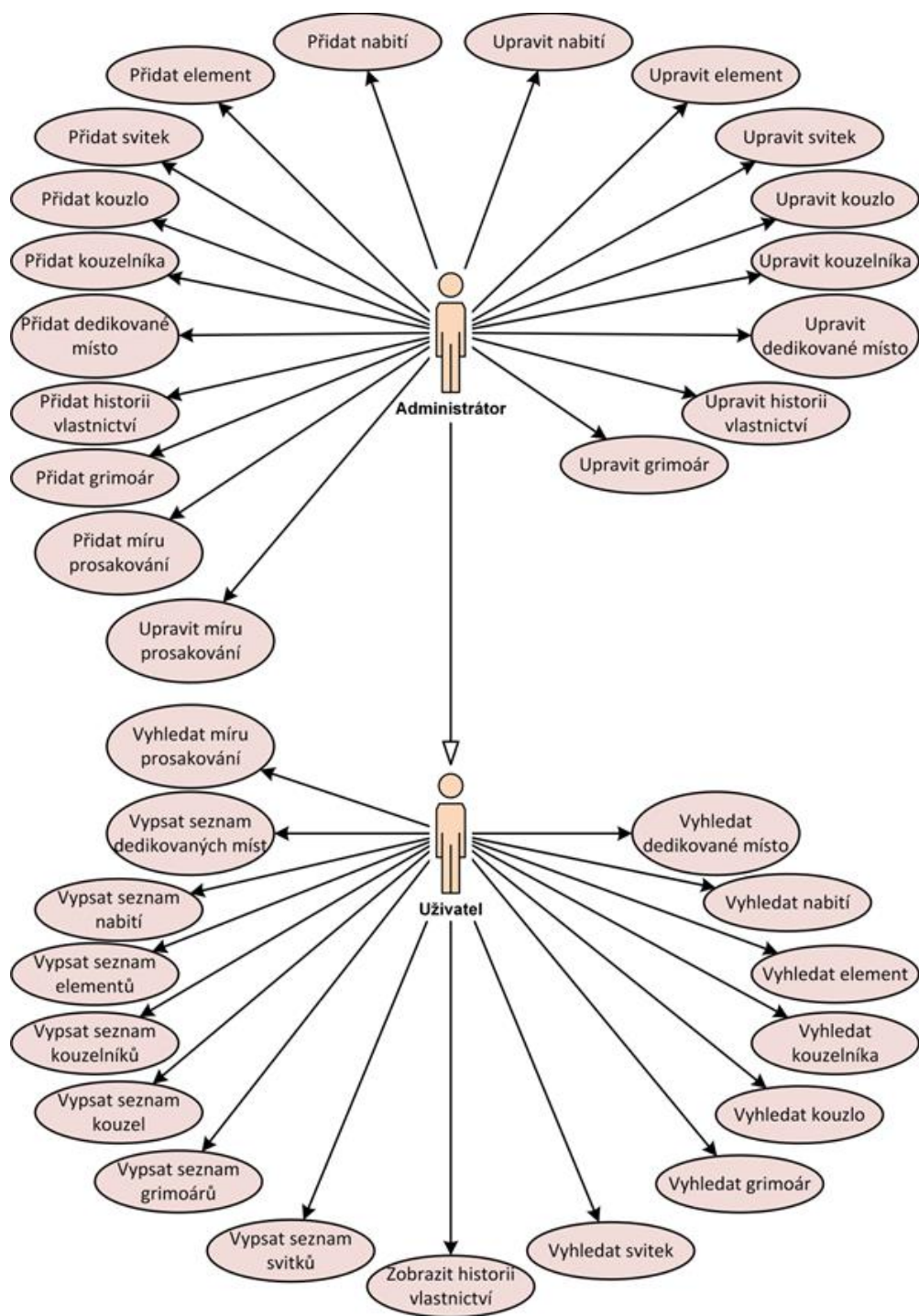


Diagram relační databáze



Model případů užití



Generalizace a Specializace

Generalizace a specializace byla využita u tabulky „Kouzlo“, kde můžou být dva druhy kouzel - útočné a obranné. Tyto nové tabulky obsahují stejné typy atributů a liší se pouze v jednotlivých vazbách, které jsou ošetřeny dvěma triggerly. Z tohoto důvodu byl zvolen převod do jedné tabulky, která navíc obsahuje položku typ určující druh kouzla.

Triggerly

Jak je zmíněno výše, skript obsahuje dva triggerly, které kontrolují vazby na specializace kouzla. První kontroluje, zdali svitek obsahuje útočné kouzlo a druhý trigger kontroluje, zdali jsou vedlejší elementy přiřazeny pouze obranným kouzlům. Další dva triggerly automaticky přidělují identifikační čísla svitkům a nabitím. Ze zbývajících triggerů jeden kontroluje čas, kdy bylo provedeno nabití a poslední řeší množství many kouzelníků. To je odvozeno od jejich dosažené úrovně v kouzlení, kde úroveň E může mít maximálně 2 000 many, úroveň D 5 000, C 10 000, B 20 000, A 30 000, S 50 000 a nejvyšší úroveň SS může mít maximálně 100 000 many.

Procedury

První procedura vypíše procentuální zastoupení elementu mezi kouzly a zároveň vypíše i jeho průměrnou míru prosakování ze všech dedikovaných míst. K tomu používá kurzor, kde postupně prochází jednotlivé řádky tabulky vzniklé spojením tabulek „Kouzla“ a „Elementu“ a ukládá si celkový počet kouzel a kouzel s daným elementem. Chyby jsou ošetřené výjimkami. Jedna řeší vstupní parametr, pokud by nebyl nalezen mezi elementy a další řeší ostatní chyby.

Druhá procedura vypíše procentuální vyjádření vzácnosti daného kouzla, kde zjišťuje, do jaké míry je dané kouzlo běžně dostupné. Minimum je 0%. To je nejběžnější kouzlo, které umí každý kouzelník, je v každém grimoáru a v každém svitku. Na druhou stranu maximum je 100% a to je nejvzácnější kouzlo, kde žádný kouzelník ho neumí a žádný grimoár ani svitek ho neobsahuje.

Explain plan a index

Zvolený SELECT dotaz obsahuje spojení tří tabulek, agregační funkci COUNT a klauzuli GROUP BY. Ten byl volán dvakrát, poprvé bez použití indexu a podruhé s vytvořením a využitím indexu. V obou případech byla využita funkce Explain plan, která měla dva následující výstupy:

Bez použití indexu

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		6	966	10 (20)	00:00:01
1	SORT ORDER BY		6	966	10 (20)	00:00:01
2	HASH GROUP BY		6	966	10 (20)	00:00:01
3	HASH JOIN OUTER		6	966	8 (0)	00:00:01
4	HASH JOIN OUTER		6	636	5 (0)	00:00:01
5	TABLE ACCESS FULL	ELEMENT	4	144	3 (0)	00:00:01
6	INDEX FAST FULL SCAN	PK_JE_V_SYN	5	350	2 (0)	00:00:01
7	TABLE ACCESS FULL	KOUZELNIK	4	220	3 (0)	00:00:01

S využitím indexu

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		6	966	9 (23)	00:00:01
1	SORT ORDER BY		6	966	9 (23)	00:00:01
2	HASH GROUP BY		6	966	9 (23)	00:00:01
3	HASH JOIN OUTER		6	966	7 (0)	00:00:01
4	HASH JOIN OUTER		6	636	4 (0)	00:00:01
5	TABLE ACCESS BY INDEX ROWID BATCHED	ELEMENT	4	144	2 (0)	00:00:01
6	INDEX FULL SCAN	MUJINDEX	4		1 (0)	00:00:01
7	INDEX FAST FULL SCAN	PK_JE_V_SYN	5	350	2 (0)	00:00:01
8	TABLE ACCESS FULL	KOUZELNIK	4	220	3 (0)	00:00:01

Z výše uvedených tabulek lze vyčíst, že se snížila cena daných operací, což jsou přístupy na disk, ale na druhou stranu vzrostlo využití CPU. Samotný průběh zpracování probíhá následovně. Ze všeho nejdřív je zde v obou případech SELECT STATEMENT, což je volání dotazu SELECT. Následuje sjednocení řádků podle hashovacího klíče a následně se spojí tabulky. Tabulka ELEMENT je zpřístupněna pomocí kompletního skenu tabulky, tedy se procházela celá. S použitím indexu se do tabulky ELEMENT přistupuje pomocí konkrétního řádku, kde se využije právě tento index a INDEX FAST FULL SCAN určí jeden jedinečný řádek podle primárního klíče.

Přístupová práva

Vzhledem k tomu, že tento projekt vypracovával jeden člověk, tak práva byla udělena uživateli „rychly“. Tento uživatel simuluje řadového kouzelníka, který může využívat databázi a její procedury, ale měl by mít omezený přístup, aby například nemohl v databázi upravovat ostatní uživatele.

Materializovaný pohled

Nejprve byla potřeba vytvořit materializované logy, které uchovávají změny hlavní tabulky. Díky tomu se při změnách může využít „fast refresh on commit“, který je rychlejší oproti „complete refresh“, který spouští celý dotaz materializované pohledu od začátku. Konkrétní materializovaný pohled byl vytvořen nad tabulkou „Kouzelník“. Skript nejprve vypíše vše, co daný pohled obsahuje, pak do něj vloží další data a opět vypíše jeho obsah.