



Model informačního systému
Databázové systémy
2014/2015

Ordinace praktického lékaře

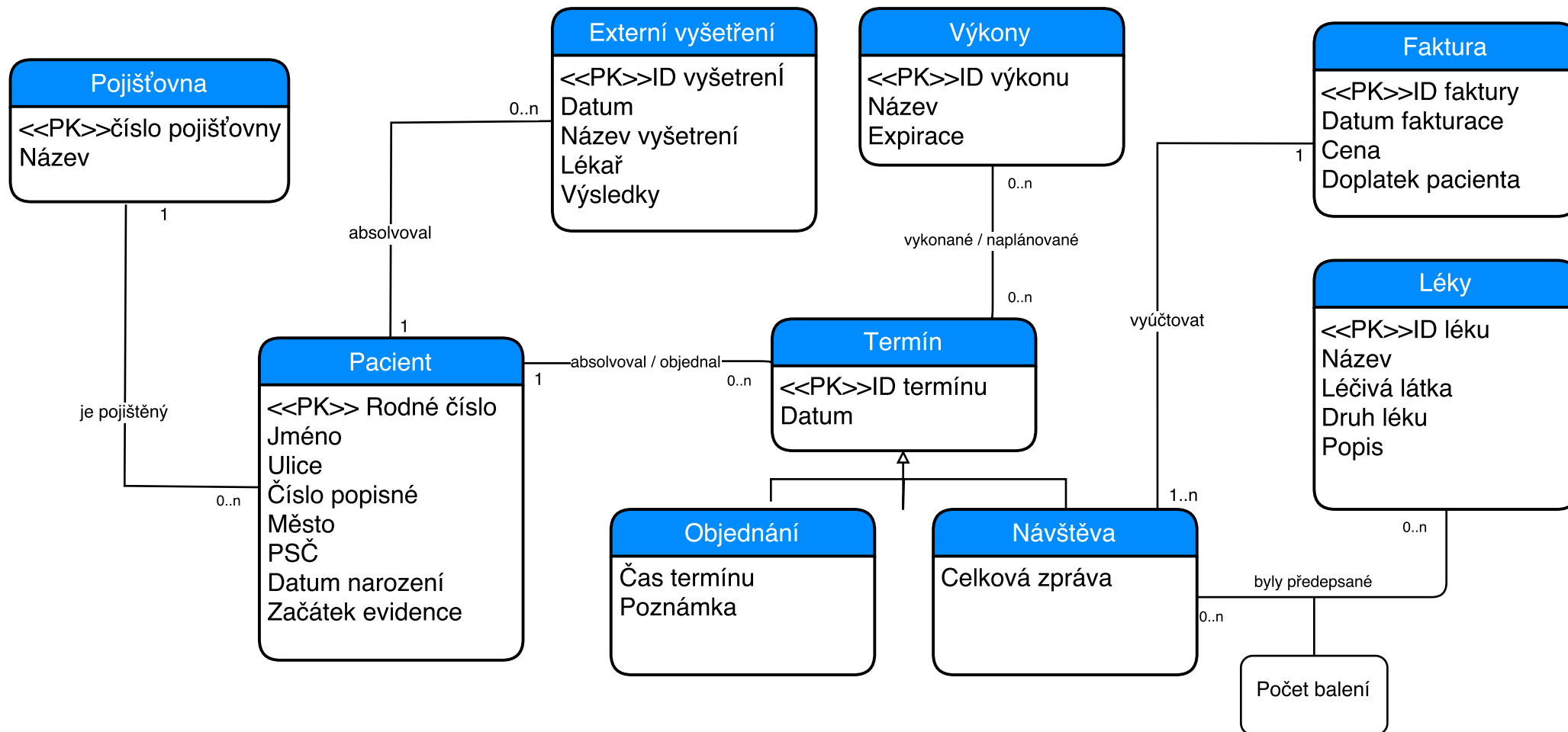
9. března 2016

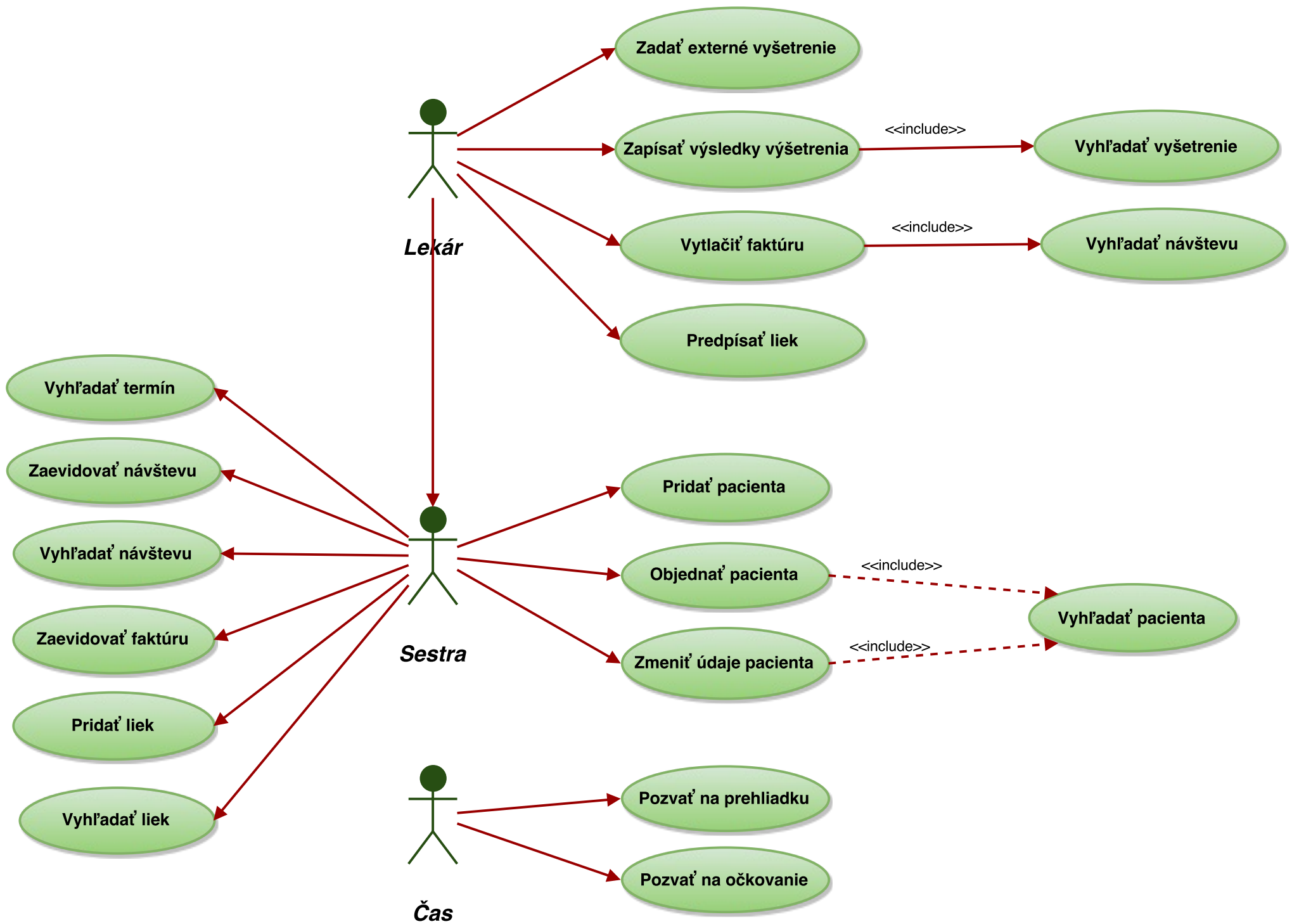
Václav Kondula, xkondu00@stud.fit.vutbr.cz
Martin Krajňák, xkrajn02@stud.fit.vutbr.cz

Fakulta Informačních technologií
Vysoké učení technické v Brně

Neformální specifikace

V ordinaci praktického lékaře jsou ošetřováni objednaní pacienti i akutní případy cizích pacientů. Uvažujte ordinaci jednoho lékaře a jedné sestry. Systém umožní sestře objednávat pacienty na daný termín a případně i specifikovat jaké výkony jsou v plánu během tohoto termínu provést. Dále eviduje návštěvy pacienta (ať již byly předem naplánovány anebo šlo o akutní případ). U každé návštěvy je třeba také evidovat jaké výkony byly provedeny a jaké léky byly předepsány. Systém bude zvat pacienty na pravidelné prohlídky a očkování, které mají různou dobu účinnosti. Pacient je vždy pojištěncem jediné pojišťovny, předpokládejte, že nás nezajímají změny pojišťovny, které pacient v průběhu času provedl. Systém musí evidovat faktury pro pojišťovnu a umožnit lékaři je vytisknout. V průběhu jedné návštěvy mohl lékař zažádat o (jedno či více) vyšetření, na které musel pacient dojít na jiné pracoviště.





Implementace

Generalizace / Specializace

Generalizovanou entitou v naší databázi je tabulka *Termin*. Ta totiž může představovat termín objednání nebo termín uskutečněné návštěvy. Pacient může přijít bez objednání a taktéž se může objednat a nepříjít. Dle zadání je také nutné evidovat uskutečněné i naplánované výkony, tudíž obě specializace se pojí se stejnými tabulkami. Rozhodli jsme se proto implementovat *Objednání* i *Návštěvu* do jedné tabulky termín a rozlišovat je na základě booleanovského přepínače *Vykonane*.

Triggery

První trigger *RODNE_CISLO* se vykoná při vkládání nebo úpravě záznamu v tabulce *pacient* a slouží k ověření validity rodného čísla. Nejprve oddělí prvních 6 znaků a převede jej na celočíselný typ integer. Pokud se jedná o ženu, odečte 5000 a následně převede toto číslo na datum pomocí funkce *TO_DATE()* a formátu 'YYMMDD'. Pokud by toto datum nebylo validní, funkce vyhodí výjimku. Následně ověřuje, jestli je celé rodné číslo dělitelné jedenácti.

Druhý trigger *ID_LEKU_INC* vytváří v tabulce léků hodnoty pro primární klíče. Pokud uživatel explicitně zadá ID léku, trigger neprovede žádnou akci, pokud však je hodnota ID *NULL*, pak jej nahradí následující hodnotou ze sekvence pro ID léků.

Materializovaný pohled

Materializovaný pohled *pojistovnaNahled* slouží k přístupu "Všeobecné zdravotní pojišťovny" k fakturám vystavených na jejich klienty. Tento pohled tedy zobrazuje rodné číslo, jméno a příjmení pacienta, ID faktury, datum, cenu a doplatek pro všechny faktury vystavené na klienty VZP. Pohled má nastavené příznaky *REFRESH ON COMMIT*, který sestaví pohled pokaždé, kdy se změní záznamy v master tabulkách a je proveden commit. Dále *BUILD IMMEDIATE*, aby se pohled sestavil okamžitě po jeho vytvoření a nakonec trojici *LOGGING*, *CACHE* a *ENABLE QUERY REWRITE*, které slouží ke zrychlení vyhledávání v materializovaném pohledu. Zaměstnance VZP v našem případě simuluje uživatel *xkrajn02*, který má práva k práci s tímto pohledem.

Procedury

Procedura *rocniZuctovani* slouží na spočítání cen, výkonů a doplateků ze jeden celý kalendářní rok. Parametry procedury jsou číslo pojišťovny, pro kterou chceme zúčtování zobrazit a rok, který představuje zúčtovací období. Podle čísla pojišťovny je vyhledán název pojišťovny a uložený do proměnné *jmeno_pojistovny*. Tato úloha využívá spojení několika tabulek, přičemž samotné spojení je uloženo do kurzoru. Následně jsou záznamy procházeny po řádcích a sčítají se požadované údaje - cena a doplatek. V proceduře je ošetřena i výjimka pro případ, že číslo zadané pojišťovny neexistuje. Výsledkem je výpis požadovaných údajů.

Úlohou procedury *analyzaDruhuLeku* vytvoření statistických údajů pro druh léku, který je zadaný parametrem. Do kurzoru jsou uloženy záznamy o všech lécích, které byly předepsané. Procedura zobrazí všechny výkony, ke kterým byli léky daného druhu předepsané, a zároveň vypočítá procentuální zastoupení daného druhu léku v rámci všech léků, které byly předepsané.

Explain plan a indexace

Pro demonstraci *explain plan* a zrychlení dotazů za pomoci indexace sloupců jsme si připravili pohled, který zobrazuje název pojišťovny a počet jejích klientů, kteří mají trvalé bydliště v Praze a jsou pacienti v naší ordinaci. Z toho důvodu jsme se rozhodli indexovat sloupec odpovídající městu trvalého bydliště pacientů.

V následujících tabulkách můžeme vidět, že přidáním indexu se snížila cena téměř všech operací a *table access full* byl nahrazen *table access by index rowid batched*. Tím došlo k výraznému zrychlení, protože přístup do tabulky přes indexy, kde nás zajímá jen malá část ze všech záznamů je daleko rychlejší než procházení celé tabulky. Přestože změny budou daleko patrnější při plných tabulkách, již nyní můžeme vidět snížení ceny ze 17 na 13 bodů.

Před přidáním indexu:

Id	Operation	Name	Rows	Bytes	Cost
0	SELECT STATEMENT		2	108	4
1	HASH GROUP BY		2	108	4
2	NESTED LOOPS		2	108	3
3	NESTED LOOPS		2	108	3
4	TABLE ACCESS FULL	PACIENT	2	44	3
5	INDEX UNIQUE SCAN	PK_POJISTOVNA	1		0
6	TABLE ACCESS BY INDEX ROWID	POJISTOVNA	1	32	0

Po přidání indexu:

Id	Operation	Name	Rows	Bytes	Cost
0	SELECT STATEMENT		2	108	3
1	HASH GROUP BY		2	108	3
2	NESTED LOOPS		2	108	2
3	NESTED LOOPS		2	108	2
4	TABLE ACCESS BY INDEX ROWID BATCHED	PACIENT	2	44	2
5	INDEX RANGE SCAN	INDEX_MESTA	2		1
6	INDEX UNIQUE SCAN	PK_POJISTOVNA	1		0
7	TABLE ACCESS BY INDEX ROWID	POJISTOVNA	1	32	0