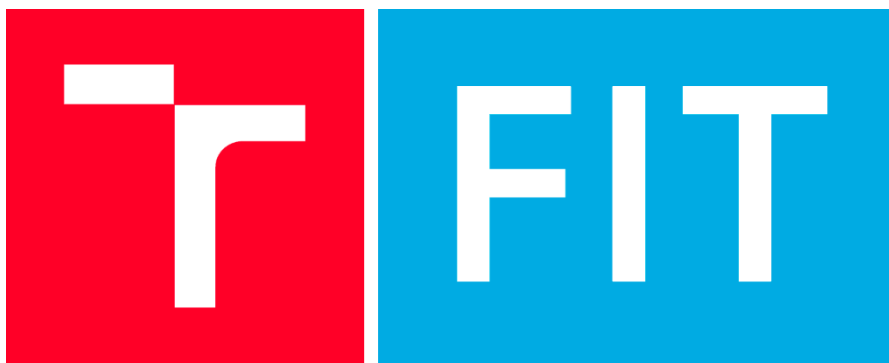


Vysoké učenie technické v Brne

Fakulta informačných technológií



Dokumentácia popisujúca finálne schéma databáze

Databázové systémy 2017/2018

Konceptuálny model

Zadanie č.30 - Nemocnica

Orsák Maroš (xorsak02)

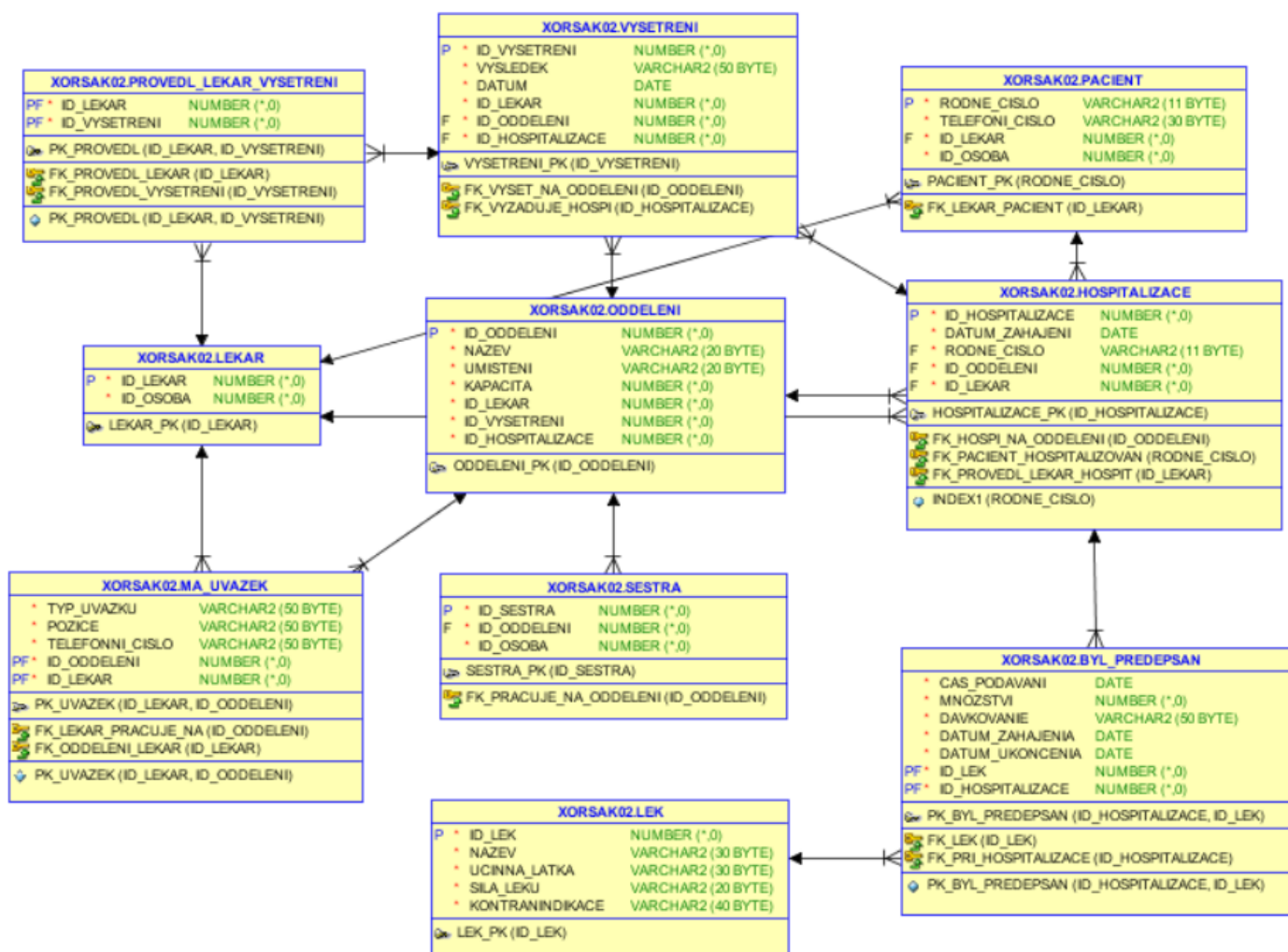
Vesovic Nevena (xvesov00)

Brno 23.4.2018

Zadanie nemocnica

Navrhnete IS malé nemocnice, který by poskytoval základní údaje o lékařích, sestrách či pacientech, kteří jsou a byli hospitalizováni v nemocnici. IS uchovává informace o všech těchto hospitalizacích, přičemž pacient může být v jeden čas hospitalizován pouze na jednom oddělení nemocnice. Při každé hospitalizaci je mu určen jeho ošetřující lékař. Lékaři mohou pracovat na více odděleních zároveň. Na každém oddělení má lékař určitý úvazek, telefon atd., zatímco sestry pracují pouze na jednom oddělení. V rámci pobytu v nemocnici může pacient podstoupit různá vyšetření, která byla provedena na určitém oddělení ve stanoveném čase a provedl je určitý lékař, který také zapisuje výsledky vyšetření do IS. Dále mu mohou být podávány různé léky, každé podávání léku má určité detaily (kdy se podává, kolikrát apod.). V systému jsou uloženy i všeobecné informace o lécích (název, účinná látka, síla léku, kontraindikace atd.), aby si lékař mohl zkontrolovat správnost naordinovaného dávkování

Schéma Relacnej Dabázy



Generalizácia a špecializácia

Vo svojej databáze využívame tabuľku OSOBA, a to v prípade že osoba môže byť pacient, sestrička alebo aj lekár. Čiže nemože sa stať napríklad, že sestrička bude zároveň aj lekár.

Implementácia

Skript ma na začiatku dropnúť všetky tabuľky a taktiež sekvencie pre prípad, že sa daný skript pustil viac ako raz. Následne nato sú vytvorené všetky tabuľky, ktoré reprezentujú objekty v našej vytvorenej databáze. Ďalej sa spúšťajú trigre, ktoré kontrolujú index a taktiež správny formát rodného čísla. Potom v našom skripte vytvárame primárne a cudzie kľúče k určeným tabuľkám. Akonáhle určíme primárne a cudzie kľúče tak vkladáme do databázy data, ktoré sa potom vyberajú a selectujú. Na koniec sprístupníme práva pre všetky tabuľky a materializovaný pohľad obom členom tímu.

Triggery

Medzi prvými vecami čo sme implementovali boli databázové trigery. Prvý slúžil na automatické inkrementovanie primárneho kľúča osoby zo sekvencie. Napríklad pokiaľ bude pri vkladaní záznamu do danej tabuľky hodnota primárneho kľúča nedefinovaná, tj. NULL. Druhý trigger slúži na overenie správneho formátu rodného čísla.

Procedúry

Druhá časť bola implementácia dvoch ne-triviálnych procedúr. Na začiatku si vytvoríme kurzor pomocou ktorého budeme schopný pracovať s viacerými riadkami v danej databáze. V procedúrach využívame ošetrovanie výnimky pri delení nulou. Procedúry slúžia na určenie percentuálneho zastúpenia doktorov a sestričiek v systéme.

Index a Explain plan

Pomocou Explain planu sme mali za úlohu demonštrovať funkciu indexu. Najskôr sme spustili Explain plan na jednoduchom selekte, ktorý vypíše kolko lekárov je na ktorom oddelení. Využíva sa pritom spojenie dvoch tabuliek. Potom sme vytvorili index v tabuľke Hospitalizace na stĺpci rodne_cislo, pomocou ktorého vzniklo efektívnejšie spojenie tabuľky pacientov a hospitalizace. Použitie indexu sme museli vynútiť, keďže na tak malej databáze by bolo vytvorenie indexu príliš náročná operácia. Po spustení sme dosiahli lepšie výsledky v zabratí procesorového času.

Náročnosť selektu bez použitia indexu:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		4	28	4 (25)	00:00:01
1	HASH GROUP BY		4	28	4 (25)	00:00:01
2	TABLE ACCESS FULL	HOSPITALIZACE	4	28	3 (0)	00:00:01

Náročnosť selektu s použitím indexu:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		4	28	1 (0)	00:00:01
1	SORT GROUP BY NOSORT		4	28	1 (0)	00:00:01
2	INDEX FULL SCAN	INDEX1	4	28	1 (0)	00:00:01

Materializovaný pohľad

Vytvorili sme materializovaný pohľad pre druhého člena tímu, ktorý používa tabuľku VYSETRENI definovanú druhým členom tímu. Aby sme pri zmenách hlavnej tabuľky mohli využívať fast refresh on commit, miesto complete refresh najprv sme si vytvorili materializované logy, kde sa uchovávajú zmeny hlavnej tabuľky. Potom sme vytvorili materializovaný pohľad a overili jeho funkčnosť na príklade, kde počítame počet vyšetrení. Prv sme vypísali čo obsahuje materializovaný pohľad, do hlavnej tabuľky pridali data, povrdili sme zmenu a znovu dali vypísať všetko čo materializovaný pohľad obsahoval.

Prístupové práva

Prístupové práva boli dané kolegovi tímu a teda mal privilégia meniť data v databáze.

Záver

Skript bol otestovaný na školskom serveri Oracle. Bol vytvorený v smart IDE od poprednej firmy JetBrains s názvom DataGrip. Využili sme mnoho zdrojov ako pomoc na StackOverflow, Wiki a Google. Za zmienku stoja aj demostrančné cvičenia, ktoré hrali veľkú rolu k pochopeniu problematiky a taktiež informácie z IDS studijnej opory.