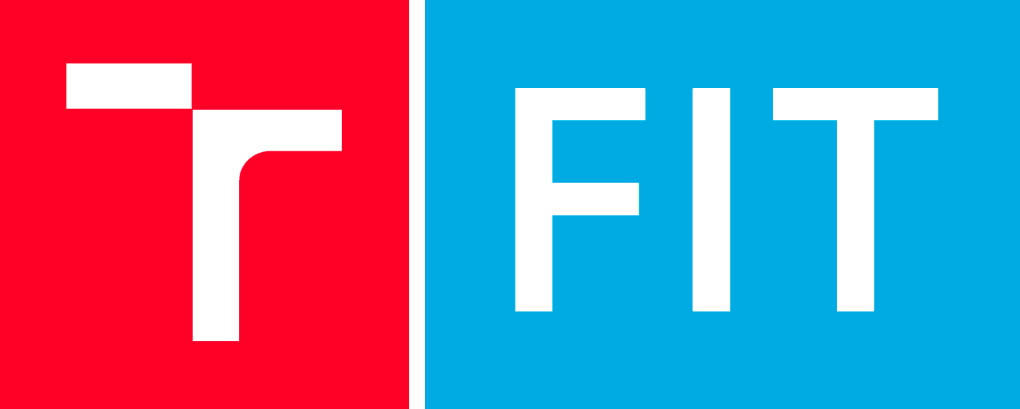
**Vysoké učenie technické v Brne**

Fakulta informačných technológií



Dokumentácia popisujúca finálne schéma databáze

Databázové systémy 2017/2018

Konceptuálni model

Zadanie č.30 - Nemocnica

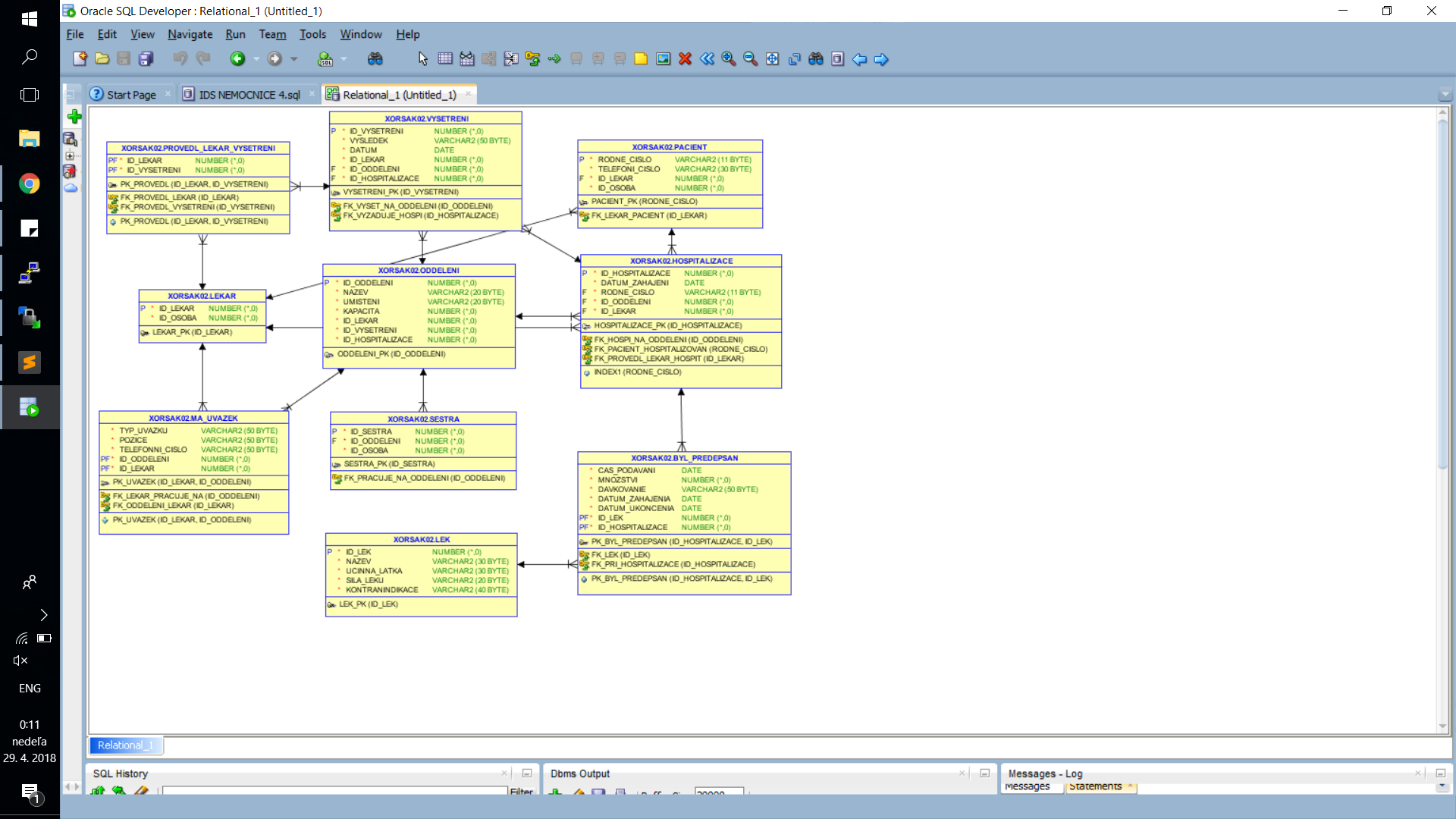
Orsák Maroš (xorsak02)

Vesovic Nevena (xvesov00)  
Brno 23.4.2018

Zadanie nemocnica

Navrhněte 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 može byť pacient, setrička alebo aj lekár. Ciže nemože sa stať napríklad, že sestrička bude zároveň aj lekár.

Implementácia

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

Triggery

Medzi prvýmivecami čo sme implementovali boli databázové triggery. Prvý slúžil na automatické inkrementovanie primárneho klúča osoby zo sekvencie. Napríklad pokiaľ bude pri vkladaní záznamu do dané tabulky hodnota primárneho klúča nedefinovaná, tj. NULL. Druhý trigger slúži na overenie správneho formátu rodného čísla.

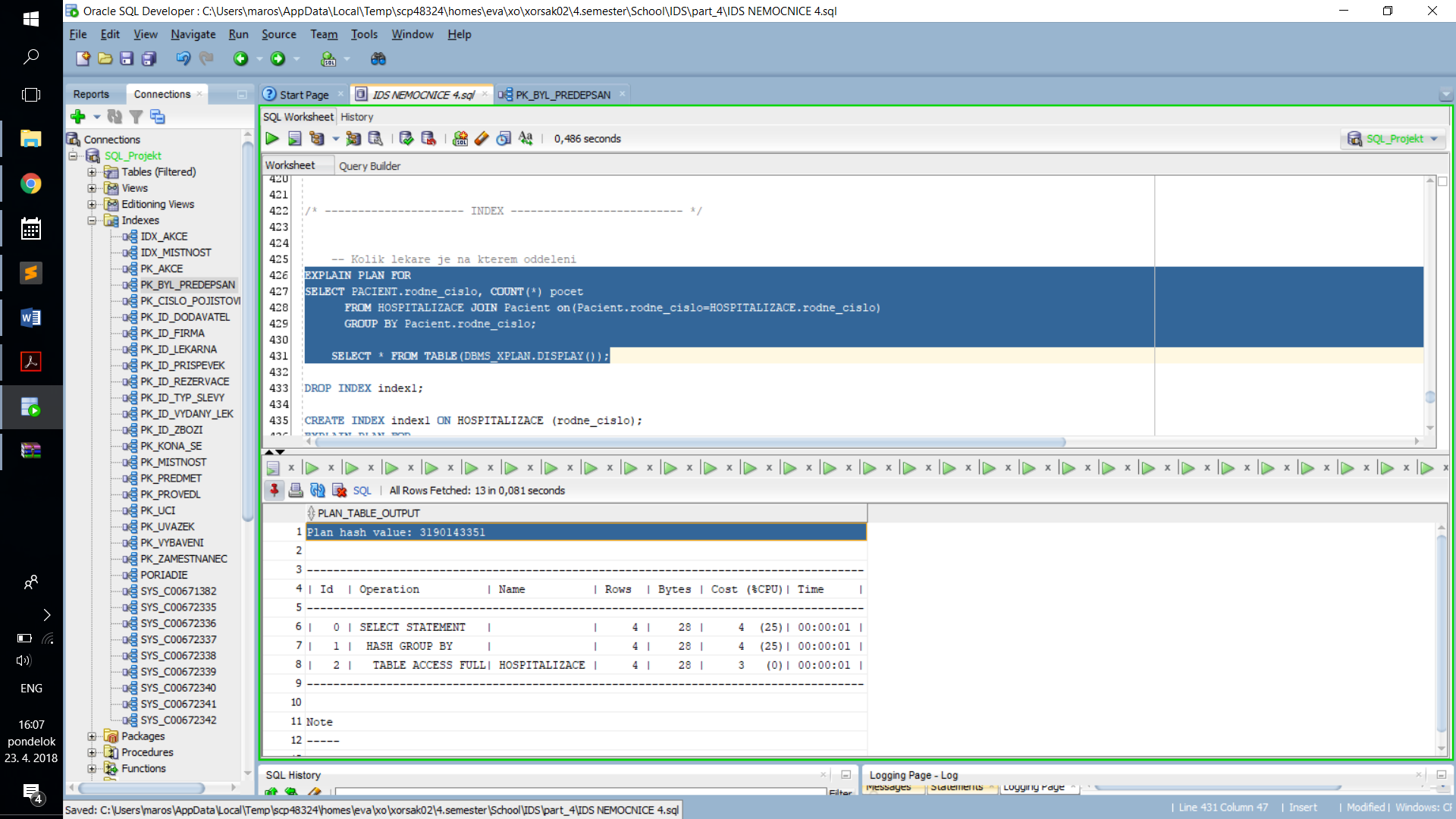
Procedúry

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 ošetrenie výnimky pri delenie nulou. Procedúry slúžia na určenie percentuálneho zastúpenia doktorov a sestričiek v systéme.

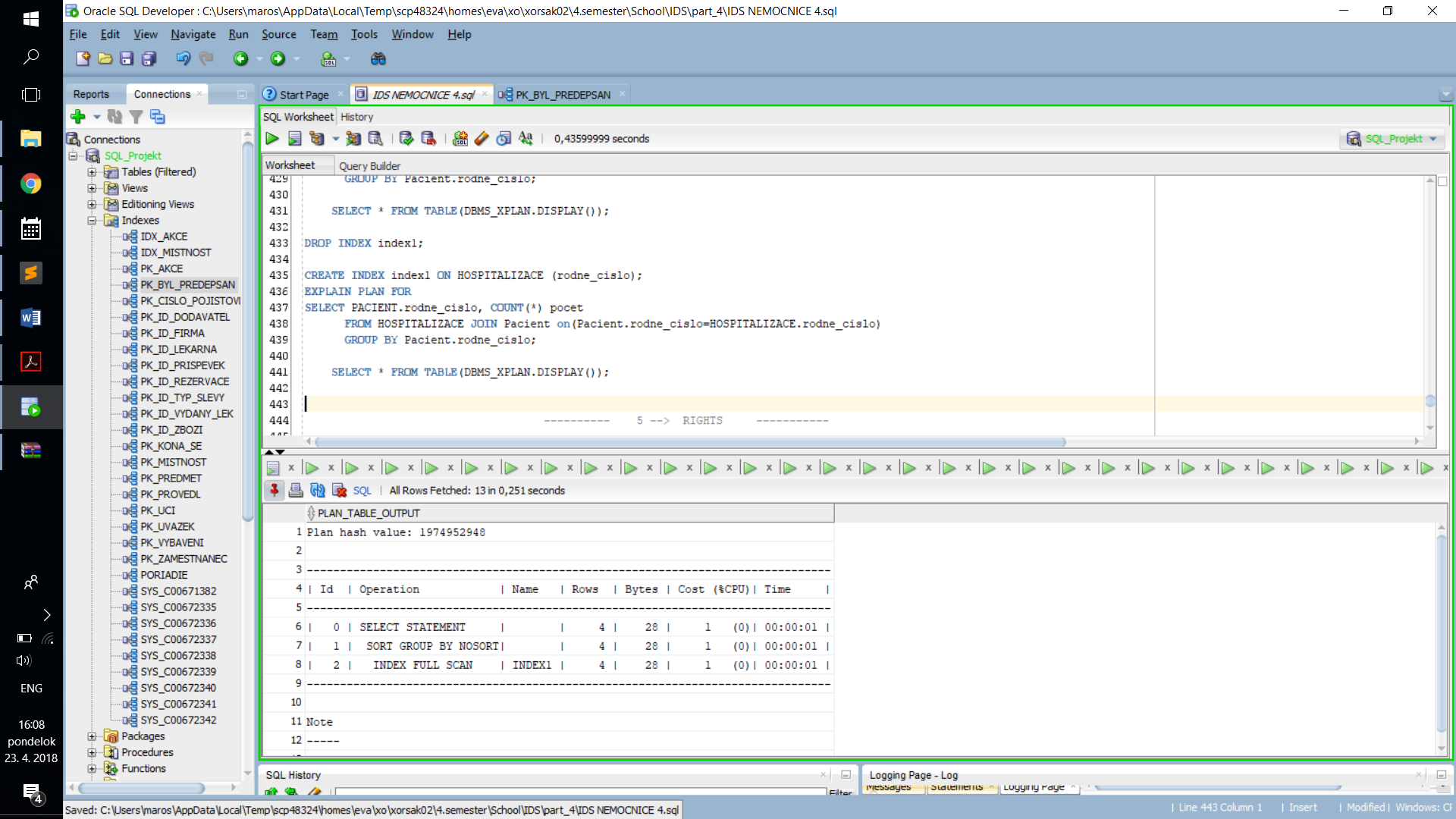
Index a Explain plan

Pomocou Explain planu sme mali za úlohu demonštrovať funkciu indexu. Najskôr sme spustili Explain plan na jednoduchom selekte, ktorý vypíše kolko lékarov je na ktorom oddeleni. Využíva sa pritom spojenie dvoch tabuliek. Potom sme vytvorili index v tabulke 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:



Náročnosť selektu s použitím indexu:



Materializovaný pohľad

Vytvorili jsme materializovany pohled pre druhého člena týmu, ktorý používa tabuľku VYSETRENI definovanú druhým členom týmu. Aby sme pri zmenách hlavnej tabuľky mohli využívať fast refresh on commit, miesto complete refresh najprv sme si vytvorili materializovane logy, kde sa uchovavaju zmeni hlavnej tabulky. Potom jsme vytvorili materializovaní pohľad a overili jeho funkčnost na príklade, kde počítame počet vyšetreni. Prv sme vypisali čo obsahuje materializovaný pohľad, do hlavnej tabulky pridali data, povrdili sme zmenu a znovu dali vypísať všetko čo materializovaní pohľad obsahoval.

Prístupové práva

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

Zaver

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