

Databázové systémy

Dokumentace projektu

Ondřej Kepřt xkepřt03@stud.fit.vutbr.cz
Vladimír Drengubiak xdreng01@stud.fit.vutbr.cz

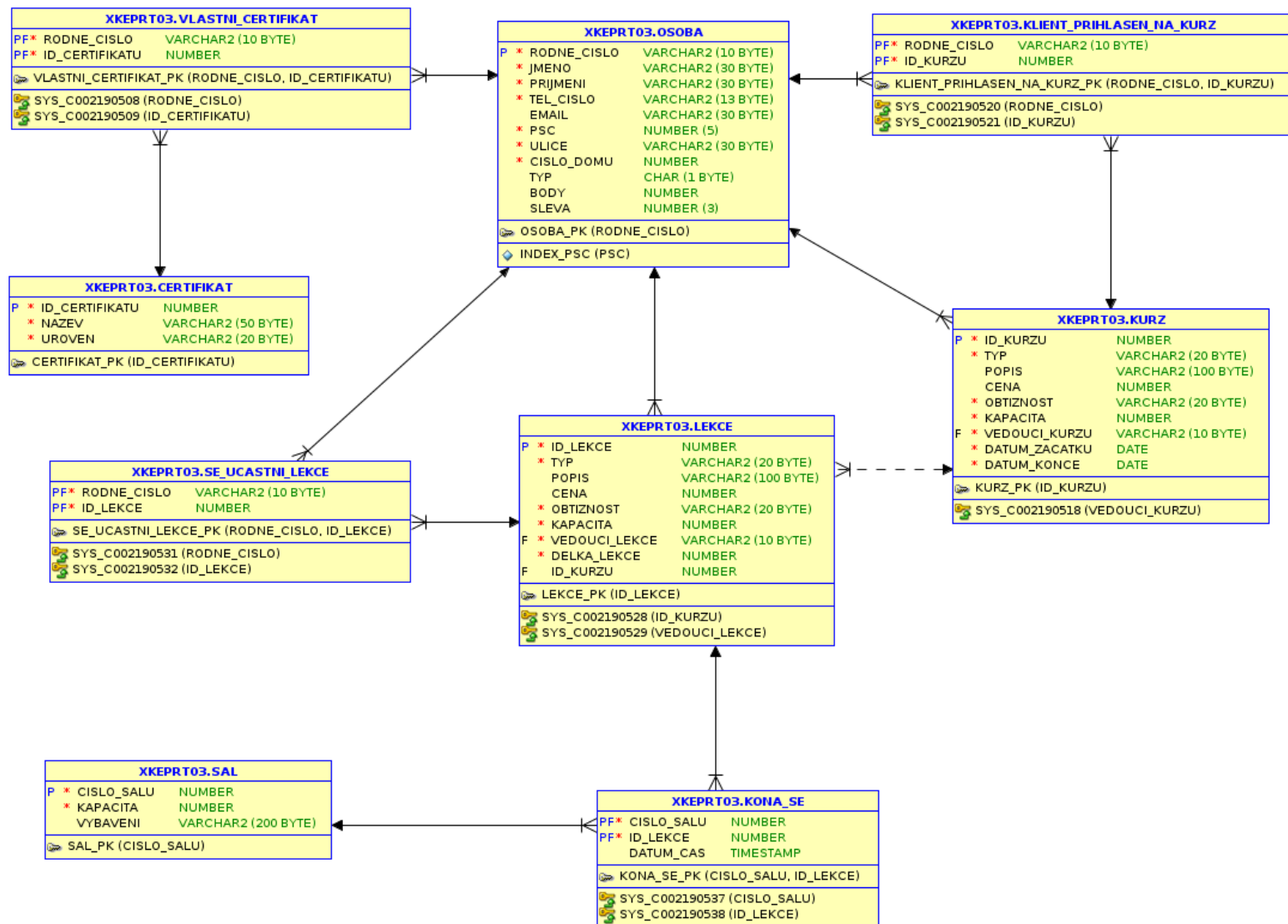
2. května 2021

1 Zadání

Rozhodli jsme se pokračovat v zadání projektu z předmětu IUS: 58. Fitness centrum

Navrhněte jednoduchý IS fitness centra, které organizuje různé kurzy skupinových lekcí (zumba, TRX, kruhový trénink, atd.). Ve fitness centru pracují instruktoři, kteří vedou jednotlivé skupinové lekce, a lidé na recepci, kteří se musí kromě vítání přichozích klientů a mixování proteinových koktejlů zapojit do práce s IS fitness centra prostřednictvím vytváření členských karet pro jednotlivé klienty, kteří se rozhodli pravidelně trápit svá těla ve fitness centru a chtějí využít členské výhody. Aby karta nebyla využívána jinými klienty než jejím vlastníkem, musí být v IS uloženy základní informace o klientech, jejich rodná čísla a adresy. Zákazník si může vypsát kurzy, které navštěvuje a informace o jednotlivých lekcích. Navíc si může zobrazit rozvrh vypisovaných kurzů a zjistit počet volných míst na jednotlivých lekcích a jejich cenu. Zákazník se může registrovat buď na jednu lekci nebo na celý kurz. Kurzy mají svou délku trvání, obtížnost a popis. Skupinové lekce probíhají v různých sálech fitness centra, které mají konkrétní název, umístění a maximální kapacitu. Lekce jsou vedené jedním instruktorem, mají maximální kapacitu účastníků a odehrávají se v daném sále v určitý čas a den v týdnu. Předpokládejte, že jeden instruktor může být vyškolen pro vedení různých kurzů, toto modelujte. Kromě pravidelných skupinových lekcí nabízí fitness centrum i individuální lekce, na kterých se instruktor věnuje pouze jednomu klientovi. Tyto lekce jsou podobného charakteru jako ty skupinové, jen je konkrétnímu klientovi věnováno více pozornosti. Instruktor má možnost vložit do systému nové typy kurzů a konkrétní lekce (a to jak skupinové, tak i individuální) a měnit čas a sál, ve kterém se lekce konají. Systém musí být na požádání schopen vypsát rozvrh pro jednotlivé místnosti.

Obrázek 1: Schéma databáze generované pomocí SQL Developeru



2 Generalizácia

V našej databáze je generalizácia reprezentovaná typom v entitnej množine osoba. Typ predstavuje klienta symbolizovaného za pomoci znaku 'K' alebo inštruktora, ktorý je symbolizovaný znakom 'I'.

3 Triggery

Zadanie požadovalo implementovať trigger na generovanie hodnôt primárneho kľúča. Trigger `ID_lekce_insert` generuje hodnoty primárneho kľúča tabuľky lekcie.

Nasledujúci trigger mal za úlohu kontrolovať vlastníctvo certifikátu osobou typu inštruktor a nápodobne bolo treba kontrolovať, že kurz alebo lekcia je vedená osobou typu inštruktor. Pokiaľ osoba nieje inštruktor, trigger zahlásí výnimku.

Bol implementovaný trigger na kontrolu rodného čísla `kontrola_rc`, podľa štandardných kritérií Českej Spávy Sociálneho Zabezpečení. Pokiaľ je vložené neplatné rodné číslo, trigger zahlásí výnimku.

4 Procedúry

Vytvorili sme procedúru `prihlasit_na_kurz` aby osoba, ktorá sa prihlási na kurz bola automaticky prihlásená aj na lekcie prislúhajúcemu kurzu. Procedúra využíva kurzor, ktorý vyhľadá všetky lekcie daného kurzu. Najprv je osoba prihlásená na kurz, potom sa prechádza kurzor a prihlasuje sa na jednotlivé lekcie. Pokiaľ je osoba prihlásená na lekcii pokračuje sa ďalej.

Taktiež bolo treba procedúru `odhlasit_z_kurzu`, ktorá zabezpečí aby osoba odhlasujúca z kurzu bola odhlásená aj zo všetkých lekcii daného kurzu.

Procedúra `zmena_vedouciho_kurzu` zaistí zmenu vedúceho kurzu. Pokiaľ procedúra prijme parameter 'Y' tak zmena vedúceho nastane taktiež v lekciach, ktoré viedol starý vedúci kurzu.

5 Explain plan

Za pomoci príkazu `explain plan` sme boli schopný ukázať plán dotazu `select`, ktorý vyhľadá osoby z mesta zadaného pomocou PSČ a zároveň navštevujú daný kurz. Pri prvom použití bola demonštrovaná verzia bez indexu čo ma malo za následok prechádzanie celej tabuľky osoba.

Obrázek 2: Explain plan bez indexu

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|----------------------|----------------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | 2 | 80 | 3 (0) | 00:00:01 |
| 1 | SORT GROUP BY NOSORT | | 2 | 80 | 3 (0) | 00:00:01 |
| 2 | NESTED LOOPS | | 2 | 80 | 3 (0) | 00:00:01 |
| * 3 | TABLE ACCESS FULL | OSOBA | 5 | 100 | 3 (0) | 00:00:01 |
| * 4 | INDEX UNIQUE SCAN | SYS_C002170647 | 1 | 20 | 0 (0) | 00:00:01 |

Nasledovalo druhé použitie `explain plan` kde spracovanie toho istého dotazu bolo obohatené o index nad stĺpcom PSČ v tabuľke osoba. `Select` už neprechádzal tabuľku osoba po riadkoch, ale využil priechod cez index nad PSČ. Namiesto prechádzania tabuľky sa využilo prechádzanie indexov, čo sa značne prejavilo na cene operácii.

Obrázek 3: Explain plan s indexom

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|-----------------------------|----------------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | 2 | 80 | 2 (0) | 00:00:01 |
| 1 | SORT GROUP BY NOSORT | | 2 | 80 | 2 (0) | 00:00:01 |
| 2 | NESTED LOOPS | | 2 | 80 | 2 (0) | 00:00:01 |
| 3 | TABLE ACCESS BY INDEX ROWID | OSOBA | 5 | 100 | 2 (0) | 00:00:01 |
| * 4 | INDEX RANGE SCAN | INDEX_PSC | 5 | | 1 (0) | 00:00:01 |
| * 5 | INDEX UNIQUE SCAN | SYS_C002170647 | 1 | 20 | 0 (0) | 00:00:01 |

Dodatočná optimalizácia by mohla byť realizovaná, že by sa najprv pomocou poddotazu najprv našli všetci členovia daného kurzu a potom z nich vybrali iba osoby s daným PSČ.

6 Materializovaný pohľad

Materializovaný pohľad vytvorený z tabuliek definovaných jedným užívateľom, pri čom pohľad patrí druhému užívateľovi. Pohľad využíva predom definovaného logu nad tabulkou kurz. Tento log nám umožňuje pohľad zmeniť okamžite po zmene tabuľky z ktorej vychádza ihneď po committe zmeny. Build immediate nám naplní pohľad dátami ihneď po vytvorení.

7 Práva

Na pridelenie práv členovi tímu boli použité príkazy grant all on a grant execute on. Grant all on udeľuje práva na prácu s tabulkami a grant execute on udeľuje práva na prácu s procedúrami.