

VYSOKÉ UČENÍ TECHNICKÉ V BRNE  
FAKULTA INFORMAČNÝCH TECHNOLOGIÍ

Databázové systémy 2020/2021  
Projektová dokumentácia  
Zadanie č.59: Social site

Matej Jurík (xjurik12)  
Marek Miček (xmicek08)

Brno  
3.5.2021

## ZADANIE :

Představte si, že jste Mark Zuckergerg v roce 2004 a chystáte se vytvořit sociální síť, která bude známá po celém světě. Jelikož chcete o svých uživateliích vědět maximum informací, budete o nich chtít uložit veškeré jejich základní informace včetně škol, které navštěvovali, bydliště, zaměstnání, kontaktu, rodiny a vztahů, atd. Jedná se o sociální síť, a tak svých uživatelům dovolíte, aby vzájemně vytvářeli (leckdy) imaginární přátelství mezi sebou. Abyste o svých uživatelích věděli všechno nejen vy, ale i ostatní uživatelé, vytvoříte zeď, na kterou budou jednotliví uživatelé publikovat příspěvky, které budou mít obsah, datum, místo a čas publikování a můžou v nich být označeni i jiní uživatelé. Aby si uživatelé mohli sdílet nejen své duchapřítomné příspěvky, ale také fotky svých domácích mazlíčků a naleštěných dvoukolových miláčků, dovolíte jim vytvářet i alba fotek, které budou mít svůj název, nastavení soukromí a popis. Na jednotlivých fotkách mohou být označeni různí uživatelé a bude u nich uveden čas, datum a místo pořízení a jedna z fotek bude vždy titulní fotka alba. Navíc může být fotka pořízena v rámci nějaké akce. Uživatelé si mohou prostřednictvím konverzací s jistým názvem, do níž můžou být zapojeni dva a více uživatelů, vyměňovat zprávy, které budou mít svůj obsah, datum, čas a místo zaslání. Aby vaší uživatelé neseseděli přeci jen stále za obrazovkami svých notebooků, dovolíte jim vytvářet akce, které se konají na určitém místě, v určitý čas a den. Účastníci akce by měli znát, o jakou akci se jedná a pokud se jim akce zalíbí, tak se mohou akce, ať už jen virtuálně, či skutečně zúčastnit.

Cieľom tejto dokumentácie je popísať implementáciu pokročilejších objektov nad schémou databáze v poslednej časti projektu do predmetu Databázové systémy. Implementácia pozostávala z vytvorenia nasledujúcich objektov :

## 1) TRIGGERY

Prvý trigger `Auto_Inc_ID` sme využívali pre automatické generovanie primárneho kľúča pre tabuľku `Akcia` v prípade, že sa do tejto tabuľky vkladala hodnota `NULL`. Najskôr sme si implementovali sekvenciu `AUTO_INC` začínajúcu číslom jeden, pričom táto hodnota sa inkrementuje vždy, keď za primárny kľúč do tabuľky `Akcia` dosadíme hodnotu z tejto sekvencie.

Druhý trigger `Check_Born_Num` slúži na overenie správnosti vkladaneho rodného čísla do tabuľky `Uzivatel`, pričom uvažujeme rodné čísla pre rok narodenia 1954 a mladší, keďže práve od tohto roku platí dnešný formát rodného čísla. Rodné číslo musí mať presne desať znakov, musí byť deliteľné jedenástimi a pre ženy platí, že k mesiacu narodenia majú pripočítané číslo päťdesiat. Práve z týchto obmedzení sme vychádzali pri implementácii tohto triggeru. Najskôr sme overili, či v rodnom čísle nie sú iné znaky ako čísla. Následne sme overili dĺžku, deliteľnosť jedenástimi a správny rozsah mesiaca narodenia (0-12). Pri overení validity dňa narodenia sme najskôr skontrolovali rozsah (1-31) a podľa mesiaca narodenia aj hornú hranicu dňa narodenia. To znamená, že človek narodený v apríli, júni, septembri alebo novembri nemôže mať deň narodenia nad číslom 30 a človek narodený vo februári nad číslom 29.

## 2) PRIVILÉGIA

V tomto prípade bola implementácia veľmi jednoduchá. Používateľovi `xjurik12` boli pridelené prístupové práva na všetky tabuľky v našej databáze a na materializovaný pohľad `JOBS`.

### 3) MATERIALIZOVANÝ POHĽAD

Materializovaný pohľad `Jobs` zobrazuje všetkých užívateľov a ich zamestnanie. Keďže na rozdiel od klasického pohľadu, materializovaný nie je aktualizovaný pri prípadnej zmene bázevej tabuľky pokiaľ to explicitne nevyžiadame pomocou nejakého triggeru alebo funkcie, tak práve túto vlastnosť sme demonštrovali našou implementáciou. Najskôr sme na náš pohľad `Jobs` aplikovali dotaz `SELECT`, následne sme aktualizovali tabuľku `Zamestnanie`, kde sme jednému užívateľovi zmenili zamestnanie a znova aplikovali dotaz `SELECT` na pohľad `Jobs`. Ako sme predpokladali, pohľad `Jobs` sa neaktualizoval.

### 4) PROCEDÚRY

Implementovali sme celkovo dve procedúry. Prvá je nazvaná `POCET_STUDUJUCICH_UZIVATELOV` a jej hlavnou úlohou je výpis počtu študujúcich užívateľov oproti celkovému počtu užívateľov. Ako detail vypisuje aj percento študujúcich užívateľov. V rámci jej implementácie je využité zachytávanie výnimky pri delení nulou, ktorá by mohla nastať pri nulovom celkovom počte užívateľov. Ďalšia, viac komplexná procedúra, s názvom `POCET_AKCII_NA_MIESTE` už využíva aj kurzor na postupné prehľadávanie tabuľky `Akcia`. V cykle tabuľkou prechádzame a hľadáme miesto akcie špecifikované parametrom tejto procedúry `CIELOVE_MIESTO`. Pri každej zhode pripočítame celkový počet zhôd do premennej `CIELOVY_POCET`, ktorú nakoniec vypíšeme ako nájdený počet akcií na špecifikovanom mieste. V tejto procedúre je ošetrená výnimka nenájdenia dát v rámci kurzora.

## 5) EXPLAIN PLAN

Detailne popisuje príkaz SELECT, v ktorom spájame 2 tabuľky, aby sme našli zoznam užívateľov, ktorých admin má priezvisko Kmeť. Index vytvárame pre Administrátorov v rámci tabuľky Užívateľ. Týmto sa pri náročnejšom vyhľadávaní optimalizuje čas vykonávania dotazu SELECT, pretože nie je potrebné prechádzať cez všetky riadky tabuľky Užívateľ.