

# IDS projekt

## Pekárna

Michal Klčo, Lukáš Galbička  
xklcom00@stud.fit.vutbr.cz, xgalbi01@stud.fit.vutbr.cz

### Popis

Skript vytvára a upravuje schému databázy pre informačný systém pekárne. Táto databáza uchováva informácie o zaregistrovaných zákazníkoch, ich objednávkach, pečive, zamestnancoch, materialoch, objednávkach materiálov a pod.. Skript taktiež vytvára rôzne ošetrenia proti porušeniu integrity a optimalizačné vylepšenia.

### Trigger

Prvý trigger `Customer_Remover` implementovaný v skripte zabezpečuje zachovanie integrity dát pri vymazávaní dát z tabuľky `Customer`. V prípade, že by sa vymazal záznam bez tohto ošetrenia, záznamy v tabuľke `Order` by mali neplatné cudzie kľúče (odkazovali by na neexistujúci záznam). Trigger teda vymaže aj záznamy z tabuľky `Order`, ale aj z tabuľky `OrderedBread`, keďže obsahujú cudzí kľúč odkazujúci sa na záznamy z tabuľky `Order`.

Ďalší trigger `auto_inc` spôsobuje implicitné nastavenie primárneho kľúča v prípade, že sa vloží nový záznam do tabuľky `Bread` a nezáadá sa hodnota primárneho kľúča explicitne (alebo zadá `NULL`). Použité hodnoty sa získavajú zo sekvencie `seq` príkazom `NEXTVAL`.

### Procedure

Procedúra `IsEnough` s jedným vstupným parametrom (meno materiálu) vypisuje stav a správu na základe množstva materiálu v tabuľke `Materials`. Ak je množstvo daného materiálu na sklade pod určitú úroveň, procedúra vypíše správu o nedostatku. Procedúra zachytáva výnimku `NO_DATA_FOUND` v prípade, že ako vstupný argument bol zadaný názov materiálu, ktorý sa nenachádza v tabuľke `Materials`.

Taktiež procedúra `Salary` je implementovaná s jedným vstupným parametrom, ktorý predstavuje minimálnu hranicu platu zamestnanca, ktorý má byť vypísaný. V deklaračnej časti je vytvorený kurzor príkazom `SELECT`, ktorý vyberá všetky záznamy z tabuľky `Employee`. Pomocou podmieneného príkazu `sum` potom vypísané len požadované záznamy.

## 1 Explain plan a index

Vytvorili sme požiadavku pre výpis všetkých objednávkov pečiva a celkovej ceny (s použitím `SUM`). Tento príkaz `SELECT` pracuje nad tromi tabuľkami. Výpis z `EXPLAIN PLAN` nás informuje o tom, že metódou spojenia pri vykonávaní tohto príkazu je Hash Joins, ktorá je často použitá so spojením `INNER JOIN` (v našom prípade s klauzulou `WHERE`). Optimalizátor zvolil priamy prístup k tabuľkám Full Table Scan - obyčajné prechádzanie tabuľkami. Vytvorením dvoch indexov nad dvoma stĺpcami, z ktorých je braná hodnota pre výpočet celkovej ceny objednávky sme docieli to, že optimalizér pristupuje k tabuľkám pomocou Index Fast Full Scans, Index Unique Scans a Table Access by Rowid. Výraznejšie zrýchlenie by však bolo pozorovateľné až pri väčšom počte záznamov v tabuľkách.

## **Materialized view a prístupové práva**

Po pridelení prístupových práv užívateľovi xgalbi01, tento užívateľ vytvorí materializovaný pohľad z údajov z tabuliek užívateľa xklcom00. Pomocou príkazu `SELECT` nad týmto pohľadom má tak užívateľ efektívny prístup k záznamom z iných tabuliek.