

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Finální schéma databáze Pekárna

Daniel Fajmon - xfajmo05

Jakub Dunčko - xdunck01

29. dubna 2023

Obsah

| | | |
|----------|---|----------|
| 1 | Zadání | 2 |
| 2 | Popis systému | 2 |
| 3 | Pojmy | 2 |
| 4 | Entity Relationship Diagram | 3 |
| 5 | Use Case Diagram | 4 |
| 6 | Generalizace | 5 |
| 7 | Implementace | 5 |
| 7.1 | Drop table | 5 |
| 7.2 | Create table, insert, check | 5 |
| 7.3 | Select | 5 |
| 7.4 | Trigger | 5 |
| 7.5 | Explain plan | 6 |
| 7.5.1 | Provedení explain plan bez indexace | 6 |
| 7.5.2 | Provedení explain plan s indexací | 7 |
| 7.6 | Procedures | 7 |
| 7.7 | Materialized view | 7 |
| 7.8 | With case select | 8 |
| 7.9 | Privileges | 8 |
| 8 | Závěr | 8 |

1 Zadání

Navrhněte jednoduchý informační systém pekárny, který bude spravovat informace o nabízených druzích pečiva jak z hlediska výroby (suroviny, náklady atd.), tak odbytu (objednávky). Pekárna zajišťuje rozvoz vlastními auty nebo si odběratelé odvoz zajišťují sami. Systém musí umožnit vedení pekárny plánovat produkci v závislosti na objednávkách, poskytovat informace pro rozvoz apod.

2 Popis systému

Systém umožňuje zákazníkům snadné vytváření objednávek na pečivo z aktuální nabídky a získání informací o složení včetně alergenů. Zákazníci mohou volit způsob dopravy a sledovat stav své objednávky.

Ředitel výroby má přehled o počtu objednávek, plánuje objednávání surovin a pečiva a přiděluje firemní auta šoférům. Dále má možnost zobrazit náklady za palivo za aktuální den. Pekaři a šoféři mohou upravovat stav objednávek (zpracovává se, připravena k vyzvednutí, doručena).

Systém také ukládá informace o zaměstnancích a zákaznících a umožňuje jejich úpravu nebo smazání.

3 Pojmy

Jako **generalizaci** jsme použili entity pekaře a šoféra, kteří dědí vlastnosti zaměstnance.

Číslo účtu si u zákazníka pamatujeme z důvodu dalších možných objednávek a tím ulehčili práci se systémem a urychlením zpracování.

Nabídka zahrnuje všechny druhy pečiva, které je možné upéct ze surovin na skladě nebo možné doobjednat podle zákaznickových potřeb.

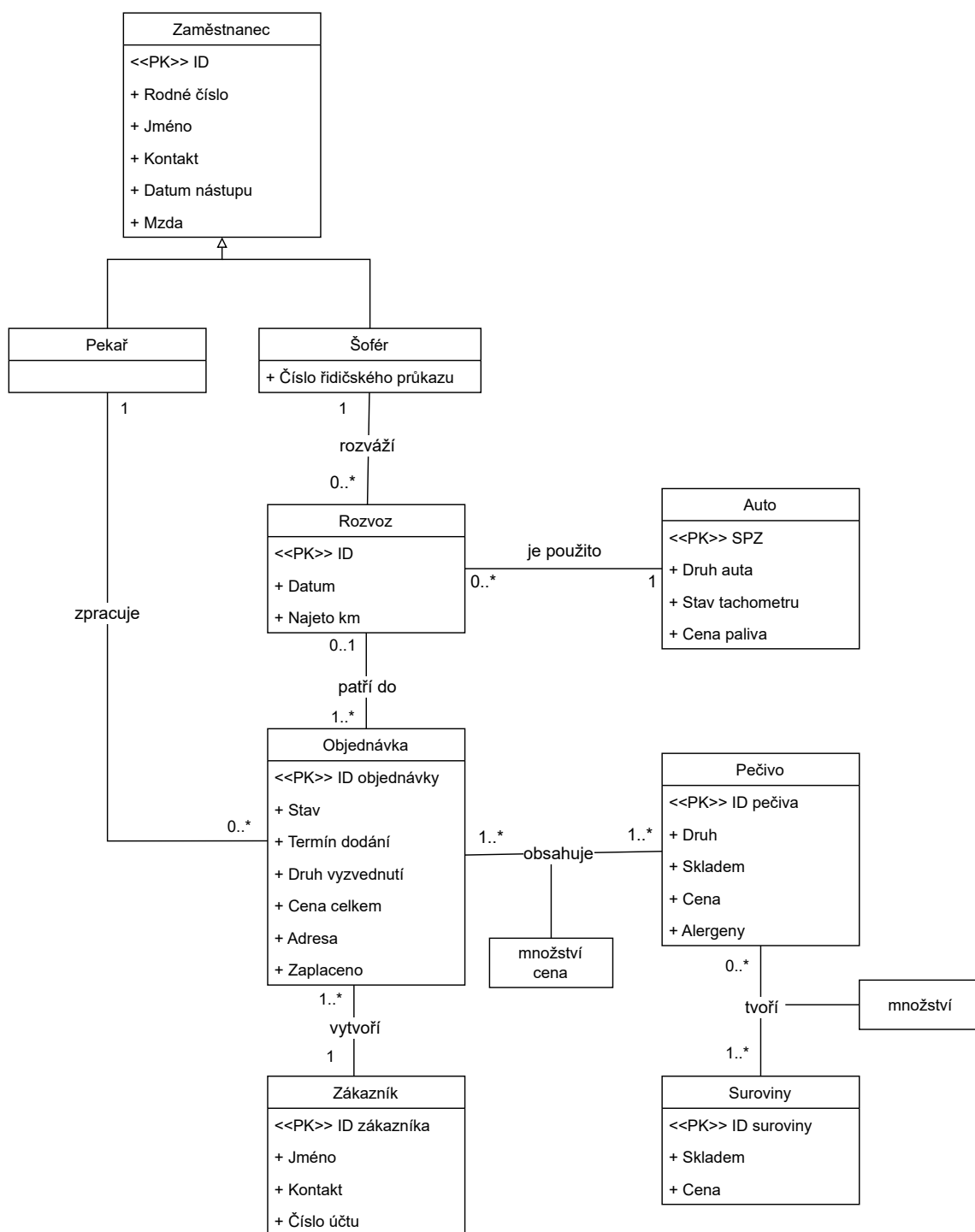
Náklady za cestu jsou spotřebované pohonné hmoty za konkrétní rozvoz.

Spravovat osobní informace je případ, kdy zaměstnanec může měnit svoje osobní informace, ale není schopen měnit informace ostatních zaměstnanců.

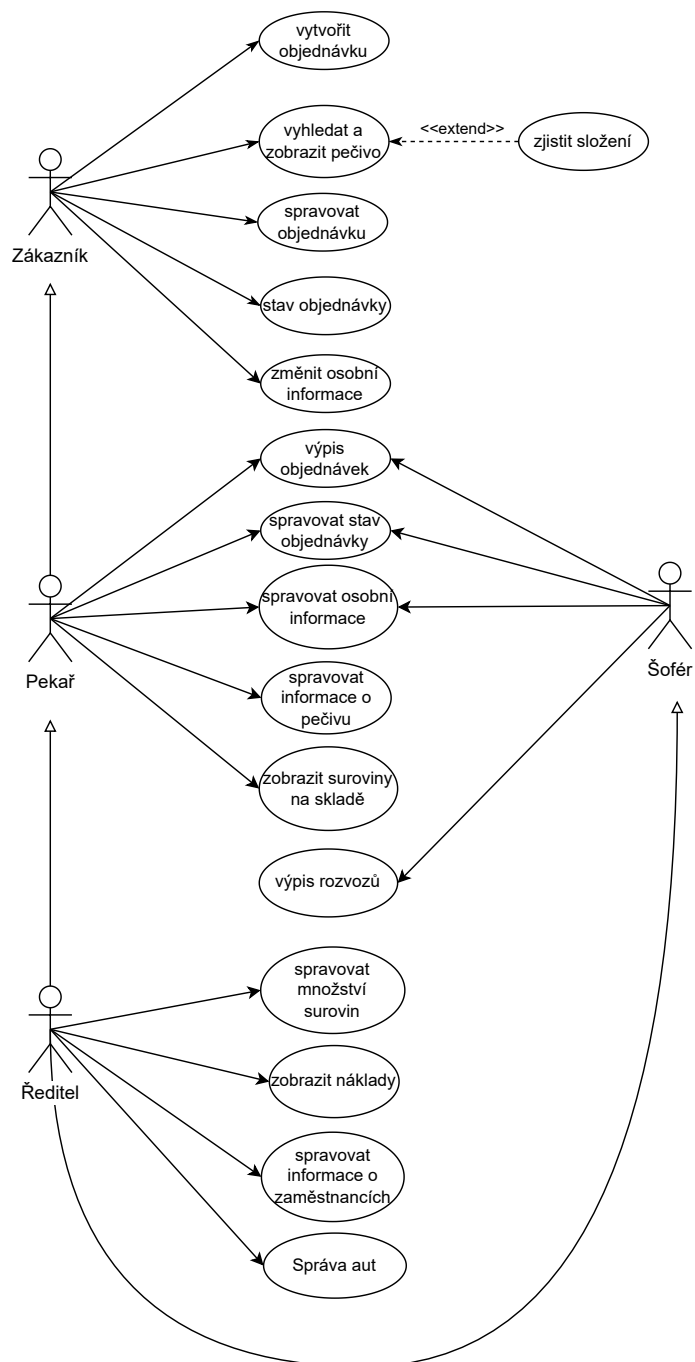
Alergeny zpracováváme podle tabulky hlavních potravinových alergenů a pečivu se přidávají pouze čísla alergenů.

Spravovat informace o zaměstnancích - pouze ředitel je schopen měnit datum nástupu zaměstnance a jeho výši mzdy.

4 Entity Relationship Diagram



5 Use Case Diagram



6 Generalizace

Jako generalizaci jsme využili pekaře a šoféra, kteří dědí z tabulky zaměstnanec. V databázi jsme to implementovali tabulku zaměstnanec, ze které dědí ID.

7 Implementace

7.1 Drop table

Smazání duplicitních tabulek v případě, že tabulky se stejným názvem již existují. Bez tohoto by nebylo možné skript pouštět opakovaně.

7.2 Create table, insert, check

Vytvoření tabulek s atributy a kontrolami nad daty. Nastavení sdílených klíčů. Následně vložení dat.

7.3 Select

Vytvoření tabulek, které mohou uživatelé databáze použít.

- 1 Informace o zákazníkovi a objednávce
- 2 Informace o řidiči
- 3 Informace pro řidiče kam a komu doručit objednávku
- 4 Počet objednávek podle druhu vyzvednutí
- 5 Kolik pečiva je třeba vypéct na objednávku
- 6 Výpis objednávek, které lze vyzvednout
- 7 Výpis kolik je potřeba materiálu na všechny aktuální objednávky

7.4 Trigger

Za úkol bylo vytvořit 2 triggery. My jsme vytvořily:

- 1 Zrušení objednávky podle datumu
- 2 Zrušení objednávky podle množství pečiva

Náš první trigger kontroluje, že objednávka má termín objednání menší jak termín dodání. V opačném případě dojde k erroru a vypíše se "Neplatný termín dodání. Dodání musí být v budoucnosti.". Druhý trigger kontroluje, že objednávka obsahuje počet pečiva větší jak 0, jinak dojde k chybové hlášce "Počet pečiva na objednávce musí být větší jak 0.".

7.5 Explain plan

EXPLAIN PLAN je příkaz, který poskytuje informace o plánu dotazu, tedy jakým způsobem bude dotaz vykonán v databázi. Tyto informace jsou užitečné pro optimalizaci dotazů a výkonu databáze, a mohou pomoci najít potenciální problémy v dotazu a navrhnout vhodná řešení a možnosti k optimalizaci.

7.5.1 Provedení explain plan bez indexace

```
PLAN_TABLE_OUTPUT
Plan hash value: 134555561
```

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|-------------------|----------------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | 9 | 2745 | 7 (15) | 00:00:01 |
| 1 | SORT GROUP BY | | 9 | 2745 | 7 (15) | 00:00:01 |
| * 2 | HASH JOIN | | 9 | 2745 | 6 (0) | 00:00:01 |
| 3 | TABLE ACCESS FULL | CUSTOMER | 7 | 1953 | 3 (0) | 00:00:01 |
| * 4 | TABLE ACCESS FULL | CUSTOMER_ORDER | 9 | 234 | 3 (0) | 00:00:01 |

0 SELECT - celkový select dat

1 HASH GROUP BY - vytvoření hashovací tabulky spojených pomocí řádků

2 HASH JOIN - spojení tabulek CUSTOMER a CUSTOMER_ORDER pomocí hashovacího algoritmu

3 TABLE ACCESS FULL - plné skenování tabulky CUSTOMER, která vrací 7 řádků

4 TABLE ACCESS FULL - plné skenování tabulky CUSTOMER_ORDER, která vrací 9 řádků

Celkový odhad provedení je dotazu je 1 sekunda.

7.5.2 Provedení explain plan s indexací

```
PLAN_TABLE_OUTPUT
Plan hash value: 2612057074
```

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|-------------------------------------|------------------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | 9 | 2745 | 6 (17) | 00:00:01 |
| 1 | SORT GROUP BY | | 9 | 2745 | 6 (17) | 00:00:01 |
| * 2 | HASH JOIN | | 9 | 2745 | 5 (0) | 00:00:01 |
| 3 | TABLE ACCESS FULL | CUSTOMER | 7 | 1953 | 3 (0) | 00:00:01 |
| 4 | TABLE ACCESS BY INDEX ROWID BATCHED | CUSTOMER_ORDER | 9 | 234 | 2 (0) | 00:00:01 |
| * 5 | INDEX RANGE SCAN | INDEX_ORDER_PAID | 9 | | 1 (0) | 00:00:01 |

0 SELECT - celkový select dat

1 SORT GROUP BY - seřazení výsledků podle kritérií

2 HASH JOIN - spojení tabulek CUSTOMER a CUSTOMER_ORDER pomocí hashovacího algoritmu

3 TABLE ACCESS FULL - plné skenování tabulky CUSTOMER, která vrací 7 řádků

4 TABLE ACCESS BY INDEX ROWID BATCHED - přistoupení k tabulce přes index a načtou se pouze ty řádky, které jsou relevantní pro dotaz

5 INDEX RANGE SCAN - získávají se data, která splňují podmínky dotazu pomocí indexu index_order_paid

Explain plan ukazuje že s indexací přibude jedna operace a navýší se využití CPU, ale sníží se celková cena dotazu.

7.6 Procedures

1 ORDER_STATUS_UPDATE - získá status konkrétní objednávky a posune ji do dalšího stavu

2 ORDER_STATUS_LIST - vypíše všechny objednávky v zadaném stavu

7.7 Materialized view

Materializovaný pohled byl vytvořen pro zobrazení informací o autech a jejich řidičích. Využili jsme:

1 CACHE - materializovaný pohled bude ukládán v cache a nebudou se opakovaně provádět dotazy na vytvoření pohledu

2 BUILD IMMEDIATE - materializovaný pohled bude vytvořen okamžitě poté, co je definice pohledu vytvořena a uložena

3 ENABLE QUERY REWRITE - umožňuje přepisovat dotazy tak, aby využívaly materializovaný pohled místo tabulek, z nichž byly vytvořeny

7.8 With case select

Vytvořili jsme dotaz, který se podívá na všechny objednávky a určí podle ceny jakou prioritu má objednávka. Priorita se určuje podle průměrné ceny všech objednávek.

7.9 Privileges

Práva byla udělena vedoucímu pekárny (xdunck01), který by měl být schopen měnit všechny údaje.

8 Závěr

Závěrem je kompletní relační databáze testována na školním serveru oracle gort.fit.vutbr.cz na verzi Oracle Database 18c Enterprise Edition Release 18.0.0.0.0 - Production