

UNIVERZITA PARDUBICE  
Fakulta elektrotechniky a informatiky

System pro podporu řízení provozu obalové linky  
mlékárenské společnosti

David Bělina

Bakalářská práce

2013

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2012/2013

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **David Bělina**  
Osobní číslo: **I10010**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Systém pro podporu řízení provozu obalové linky mlékárenské společnosti**  
Zadávající katedra: **Katedra informačních technologií**

### Z á s a d y   p r o   v y p r a c o v á n í :

Teoretická část se bude věnovat problematice vývoje a použití relačních databázových systémů v rámci webových technologií.

Praktická část se bude věnovat návrhu a implementaci systému pro podporu řízení provozu obalové linky mlékárenské společnosti.

Pro aplikaci budou klíčové především následující vlastnosti:

- a) evidence skladových zásob,
- b) sledování a nastavování provozu jednotlivých balicích linek (vytížení, poruchy),
- c) evidence a sledování kvality jednotlivých obalových sérií,
- d) řešení reklamací od zákazníků,
- e) tvorba periodických reportů.

Použité technologie: HTML, JavaScript, PHP, MySQL nebo Oracle XE.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Nowicki, S., Lecky-Thomson, E.: **PHP 6 Programujeme profesionálně.** Computer Press, 2010. ISBN 978-80-251-3127-5.
2. Castro, E.: **HTML, XHTML a CSS Názorný průvodce tvorbou WWW stránek.** Computer Press, 2007. ISBN 978-80-251-1531-2.
3. Lacko, L.: **1001 tipů a triků pro SQL.** Computer Press, 2011. ISBN 978-80-251-3010-0.
4. Oggel, B., Kofler, M.: **PHP 5 a MySQL 5 Průvodce webového programátora.** Computer Press, 2007. ISBN 978-80-251-1813-9.

Vedoucí bakalářské práce:

**RNDr. David Žák, Ph.D.**

Katedra informačních technologií

Konzultant bakalářské práce:

**Ing. Tomáš Váňa**

Katedra informačních technologií

Datum zadání bakalářské práce:

**21. prosince 2012**

Termín odevzdání bakalářské práce:

**10. května 2013**



prof. Ing. Simeon Karamazov, Dr.  
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.  
vedoucí katedry

V Pardubicích dne 29. března 2013

## **Prohlášení autora**

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Nesouhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 13. 8. 2013

David Bělina

## **Poděkování**

Na tomto místě bych rád poděkoval vedoucímu své bakalářské práce panu RNDr. Davidu Žákovi, Ph.D. a především Ing. Tomáši Váňovi za pomoc a cenné rady, které mi poskytli v souvislosti s touto prací.

## **Anotace**

Teoretická část bakalářské práce se bude věnovat problematice vývoje a použití relačních databázových systémů v rámci webových technologií. V praktické části bude vytvořen návrh a implementace systému pro podporu řízení provozu obalové linky mlékárenské společnosti.

## **Klíčová slova**

PHP, MySQL, databázový systém, webové rozhraní, výroba, relační datový model.

## **Title**

System to support traffic management packaging lines dairy company

## **Annotation**

The theoretical part will deal with development and use of relational database systems in web technologies. In the practical part will be created of the design and implementation of a system to support traffic packaging lines dairy company.

## **Keywords**

PHP, MySQL , database system, web interface, production, relation data model.

# Obsah

<b>Seznam zkratk</b> .....	<b>8</b>
<b>Seznam obrázků</b> .....	<b>9</b>
<b>Seznam tabulek</b> .....	<b>10</b>
<b>Úvod</b> .....	<b>11</b>
<b>1 Databázový systém</b> .....	<b>12</b>
1.1 Historie.....	12
1.2 Relační databázový systém .....	15
1.2.1 Systém řízení báze dat.....	15
1.2.2 Nejrozšířenější produkty .....	16
1.3 Databázové jazyky .....	18
1.3.1 SQL.....	18
1.3.2 PL/SQL .....	19
<b>2 Relační datový model</b> .....	<b>21</b>
2.1 Coddova pravidla .....	21
2.2 Princip relačního modelu .....	22
2.3 Základní relační operace .....	23
2.4 Vztahy mezi relacemi (tabulkami).....	24
2.5 Klíče .....	25
2.6 Normalizace .....	26
<b>3 Předmět činnosti implementovaného systému</b> .....	<b>28</b>
3.1 Proces výroby.....	28
3.1.1 Proces míchání .....	28
3.1.2 Balení produktu.....	28
3.2 Skladové zásoby.....	29
3.2.1 Vrchní folie .....	29
3.2.2 Spodní folie .....	30
3.2.3 Polotovary .....	30
3.2.4 Protlaký .....	30
<b>4 Implementace aplikace</b> .....	<b>32</b>
4.1 Přehled požadavků a základních funkcí v aplikaci .....	32
4.2 Použité technologie a prostředky pro vývoj.....	32
4.3 Uživatelské role.....	33

4.3.1	Skladník .....	33
4.3.2	Strojník ve výrobě .....	33
4.3.3	Mistr ve výrobě .....	33
4.3.4	Vedoucí výroby .....	33
4.3.5	Výrobní ředitel .....	34
4.3.6	Administrátor .....	34
4.4	Evidence klíčových záznamů při výrobě .....	35
4.4.1	Evidence palet .....	35
4.4.2	Evidence polotovarů .....	35
4.4.3	Evidence protlaků .....	36
4.4.4	Evidence spodní folie .....	36
4.4.5	Evidence vrchní folie .....	36
4.4.6	Evidence prostojů .....	37
4.4.7	Evidence doprodeje .....	37
4.5	Databáze .....	38
4.6	Adresářová struktura webové aplikace .....	45
4.7	Webové rozhraní .....	46
	<b>Závěr .....</b>	<b>51</b>
	<b>Literatura .....</b>	<b>52</b>
	<b>Příloha A – E-R Diagram návrhu databáze .....</b>	<b>53</b>



## Seznam zkratek

SŘBD	System řízení báze dat
IS	Informační systém
DBS	Data Base System
RDBS	Relation Data Base System
SQL	Structured Query Language
HTML	HyperText Markup Language
PHP	Hypertext Preprocessor
GPL	General public licence
DDL	Data Definition Language
DML	Data Manipulation Language
DCL	Data Control Language
TCC	Transaction Control Commands
PL/SQL	Procedural Language/Structured Query Language
NF	Normální forma
CSS	Cascading Style Sheets

## Seznam obrázků

Obrázek 1.1 - E. F. Codd.....	13
Obrázek 1.2 - Relační databázový systém .....	15
Obrázek 1.3 - Adminer logo.....	16
Obrázek 1.5 - Oracle logo .....	17
Obrázek 1.4 - PhpMyAdmin logo.....	17
Obrázek 1.6 - PostgreSQL logo .....	18
Obrázek 1.7 - Programový blok .....	19
Obrázek 2.1 - Relace .....	22
Obrázek 2.2 – Projekce .....	23
Obrázek 2.3 – Restrikce .....	24
Obrázek 2.4 - Vztah mezi relacemi 1:N.....	25
Obrázek 2.5 - Vztahy a klíče.....	26
Obrázek 3.1 - Proces míchání .....	28
Obrázek 3.2 - Obalová-balící linka .....	29
Obrázek 3.3 - Evidenční štítek vrchní folie .....	30
Obrázek 3.4 - Spodní folie .....	30
Obrázek 3.5 - Připojené kontejnery ve výrobě.....	31
Obrázek 4.1 - Vazba jednotlivých částí systému .....	33
Obrázek 4.2 - Use case diagram pro uživatelské role .....	34
Obrázek 4.3 - Realizace skladových zásob E-R Diagram.....	39
Obrázek 4.4 - Uživatelé E-R Diagram .....	39
Obrázek 4.5 - Reklamace E-R Diagram.....	40
Obrázek 4.6 - Evidence výroby palet E-R Diagram .....	41
Obrázek 4.7 - Ukázka DDL scriptu pro tvorbu tabulky Palety_vyroba.....	44
Obrázek 4.8 - Adresářová struktura projektu ve vývojovém prostředí Netbeans .....	45
Obrázek 4.9 - Vzhled webového rozhraní.....	46
Obrázek 4.10 - Přidávání nových uživatelů .....	47
Obrázek 4.11 - Negativní vyskakovací okno .....	47
Obrázek 4.12 - Pozitivní vyskakovací okno .....	47
Obrázek 4.13 - Přidávání nové palety- výřez pohledu z webového rozhraní .....	48
Obrázek 4.14 - Přehled plánu výroby - výřez pohledu z webového rozhraní.....	48
Obrázek 4.15 - Výpis dat do tabulky - PHP kód.....	49
Obrázek 4.16 - Přihlášení do systému - PHP kód .....	50
Obrázek 4.17 - Konverze data - PHP kód .....	50

## Seznam tabulek

Tabulka 4.1 - Formulář evidence palet .....	35
Tabulka 4.2 - Formulář polotovarů .....	36
Tabulka 4.3 - Formulář protlaků .....	36
Tabulka 4.4 - Formulář obalového materiálu.....	37
Tabulka 4.5 - Formulář doprodejů .....	38

## Úvod

V dnešní době je ve světě mnoho společností, které na trh přináší své vlastní výrobky. Tyto výrobky většinou vznikají na základě výrobního procesu. V případě kontrolních auditů, prováděných ve společnosti, je nezbytné uchovávat informace o tom, jak daný výrobek vznikl. K těmto účelům se mohou využívat tištěné formuláře nebo speciálně navržený řídicí-evidenční systém. Cílem mé bakalářské práce bude jeden takový systém navrhnout.

Ještě před samotným návrhem se budu v teoretické části věnovat relačním databázovým systémům. Nastíním průběh historických okamžiků, které se podílely na vývoji těchto systémů. Poté se zaměřím přímo na relační databázové systémy, co je jejich součástí a jednotlivé softwarové produkty z oblasti relačních databázových systémů. V poslední fázi teoretické části popíši relační model dat, principy a techniky využívající se při jeho návrhu.

Cílem praktické části je navrhnout systém pro řízení provozu obalové linky mlékárenské společnosti. Tento systém by měl nahradit systém klasických tištěných formulářů, používajících se v jedné nejmenované společnosti, zabývající se výrobou mlékárenských produktů. Klíčové pro systém bude zejména zaznamenávání údajů o vyrobených paletách. Díky evidování palet bude možné dohledat, z jakého materiálu byly výrobky na paletě vyrobeny. Výhodou je zjednodušení práce v případě vzniku problémů, kdy daný výrobek nebude vyhovovat požadavkům a bude se zjišťovat příčina problému. Cílem bude především zjednodušit práci uživatelům a vytvořit co nejjednodušší, intuitivní rozhraní. Systém bude dále schopen zobrazovat informace o skladových zásobách a poskytovat různé periodické reporty o výrobě. Evidování reklamací od odběratelů by v tomto systému také nemělo chybět. Dále bude zaznamenávat vzniklé poruchy a specifikovat jejich druh.

Webové uživatelské rozhraní bude validní a dodržovat specifikace W3C, které by měly zaručit správné zobrazení v internetových prohlížečích.

K praktické části mi pomohou informace a formuláře, z nejmenovaných mlékáren, kde mi byly poskytnuty podklady pro tvorbu a návrh systému.

# 1 Databázový systém

Databázový systém neboli DBS slouží pro uchovávání informací, vyhledávání a poskytování údajů. Můžeme je dělit podle použitého datového modelu na před-relační (hierarchické a síťové), relační nebo postrelační (objektově orientované, deduktivní, atd.). V dnešní době jsou nejrozšířenější relační databázové systémy, případně objektově-relační.

## 1.1 Historie

Databázové systémy za sebou mají dlouhý vývoj, jejich architektura a databázový model se měnil. Předchůdcem byly papírové kartotéky, které byly uspořádány dle různých kritérií. Veškeré jejich změny závisely přímo na člověku.

Postupně se začalo zavádět zpracování dat na stroji. První velké strojové zpracování dat bylo provedeno ve Spojených státech amerických v roce 1890 a to sčítání lidu. Tato databáze byla vytvořena Hermanem Hollerithem. Paměťovým médiem byl děrný štítek a na elektromechanických strojích probíhalo zpracování dosažených informací. Těto metody zpracování dat se využívalo další půl století.

V roce 1935 bylo ve Spojených státech uzákoněno vedení informací o 26 milionech zaměstnancích jako Social Security Act. Firma IBM tak vytvořila první digitální počítač pro komerční využití, tzv. UNIVAC I., založený na státem podporovaném projektu Electronic Discrete Variable Automatic Computer (EDVAC) z University of Pennsylvania. Mozkem těchto počítačů byl stále děrný štítek.

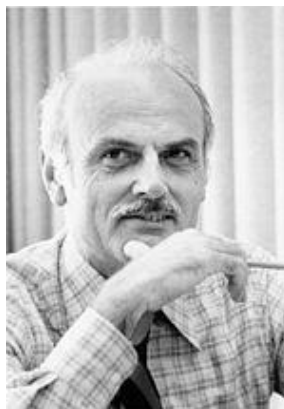
V padesátých letech 20. století se s vývojem počítačů rozvíjí i databáze. Původní univerzální strojový kód procesorů byl nahrazen požadavkem na vyšší jazyk pro zpracování dat. Byl proto formulován požadavek na univerzální databázový jazyk konferencí z roku 1959. V dalším roce byla publikována první verze jazyka COBOL (common business-oriented language) na konferenci CODASYL. Vzniknuté seskupení CODASYL (Data System Languages) bylo ustanoveno ministerstvem obrany USA pro standardizaci software aplikací. Tato verze jazyka COBOL se stala nejrozšířenější pro hromadné zpracování dat na příštích několik let. Pokrokem byl přechod od magnetických pásek umožňující sériový přístup k datům k magnetickým diskům. Na konferenci CODASYL v roce 1965 byl vytvořen výbor Database Task Group (DBTG) Charlesem Bachmanem a dalšími výzkumníky. Jeho úkolem bylo vytvoření koncepce databázových systémů. Vznikaly tak první síťové systémy řízení bází dat (SŘBD) na sálových počítačích. Charles Bachman, průkopník databází, ve firmě General Electric prodával produkt Integrated Data Store (IDS). Posléze vznikla na bázi CODASYL řada produktů od firem Eckert-Mauchly Computer Corporation (UNIVAC), Honeywell Incorporated, Siemens AG a pro mikropočítače od Digital Equipment Corporation (DEC) a Prime Computer Corporation.

Tyto základní SŘBD vycházely ze dvou přístupů:

- vzájemně propojované soubory dat (síťové a hierarchické databáze)
- fyzicky nezávislé soubory (relační databáze)

Ve zprávě The DBTG April 1971 Report se objevují pojmy jako databáze, jazyk pro definici schématu, subschéma a jiné. V této zprávě byla výborem popsána celá architektura síťového databázového systému.

Jedním z prvních SŘBD byl IMS vyvinutý v roce 1968 firmou IBM pro program letu na Měsíc Program Apollo. Firma IBM má základ ve spojení firmy Hermana Holleritha s další firmou v roce 1911, kdy vznikla firma International Business Machines. První relační databáze startují článkem E. F. Codd z roku 1970 „A Relational Model of Data for Large Shared Data Banks“, které pohlíží na data jako na tabulky. Codd se snažil přijít s možností i pro netechnické uživatele při ukládání a manipulaci s daty za použití srozumitelných příkazů z běžné angličtiny. Původní koncepce už předpokládala ukládání dat do tabulek. Tento návrh byl založen na nezávislosti dat na použitém hardware, na způsobu jejich fyzického uložení, na přístupu k datům pomocí neprocedurálního jazyka. Dle relační teorie lze pomocí základních operací uskutečnit veškeré operace s daty a ostatní operace jsou již kombinacemi jejich. Roku 1985 Codd publikoval 12 pravidel pro databázové systémy založené na relačním modelu dat.



**Obrázek 1.1 - E. F. Codd**

Průběžně se vyvinuly dva projekty relačních databází, System-R v rámci IBM a od roku 1972 Ingres na University of California at Berkeley s podporou armády a National Science Foundation. Vývoj u IBM prošel dvěma fázemi, v letech 1974-1975 šlo hlavně o praktické ověření návrhu relačního modelu a v letech 1978-1979 se implementoval plně funkční víceuživatelský systém. Původní jazyk System-R byl SEQUEL (Structured English Query

Language), posléze vznikla verze SEQUEL 2, která byla přejmenována na SQL. Dotazovací jazyk SQL byl uznán za standard.

Druhý projekt byl veden Michaellem Stonebrakerem a Eugenem Wongem na University of California at Berkeley. Šlo jim o vytvoření systému pro geografická data Berkeleyjské ekonomické skupiny, INGRES (Interactive Graphics and Retrieval System). Ingres a System-R byly vyvíjeny pod různými operačními systémy a hardware. Ingres používal vlastní jazyk QUEL podobný SQL. Michael Stonebraker posléze založil Ingres Corporation pro komercializaci kódu Ingresu. Další ze členů týmu Robert Epstein se stal spoluzakladatelem Sybase. Komerční verze Ingresu byla dostupná v roce 1980. Další ze členů Ingres týmu Paul Hawthorn odešel a podílel se na založení Illustra Information Technologies Incorporated, známé jako Informix. Kód z Berkeley se stal zdrojem pro vývoj téměř všech dnešních SQL databází. Mezi týmy z Berkeley a IBM probíhala spolupráce a studenti z univerzity se mohli dostat na stáž do firmy.

První SQL databází na trhu se stal v roce 1980 Oracle pro počítače VAX. Firma Oracle byla založena Larry Ellisonem, inspiroval se Systemem-R od IBM. Firma IBM přišla na trh s produktem DB2 v roce 1983 pro počítače MVS a stála u zrodu daného odvětví, nebyli ale schopni se odpoutat od starého produktu a uvést na trh produkt nový. Na výzkumu v oblasti relačních databází se podílely i další univerzity.

Řadu let bránil rozšíření relačního datového modelu nedostatečný výkon tehdejších počítačů. V osmdesátých letech přichází zlatý věk relačních databází. ISO a ANSI publikují v roce 1987 počáteční standard SQL, 1989 ISO publikuje dodatek Integrity Enhancement Feature a standard dostává název SQL89. V roce 1992 dochází k první úpravě tohoto ISO standardu, který je pak znám jako SQL2 nebo SQL92.

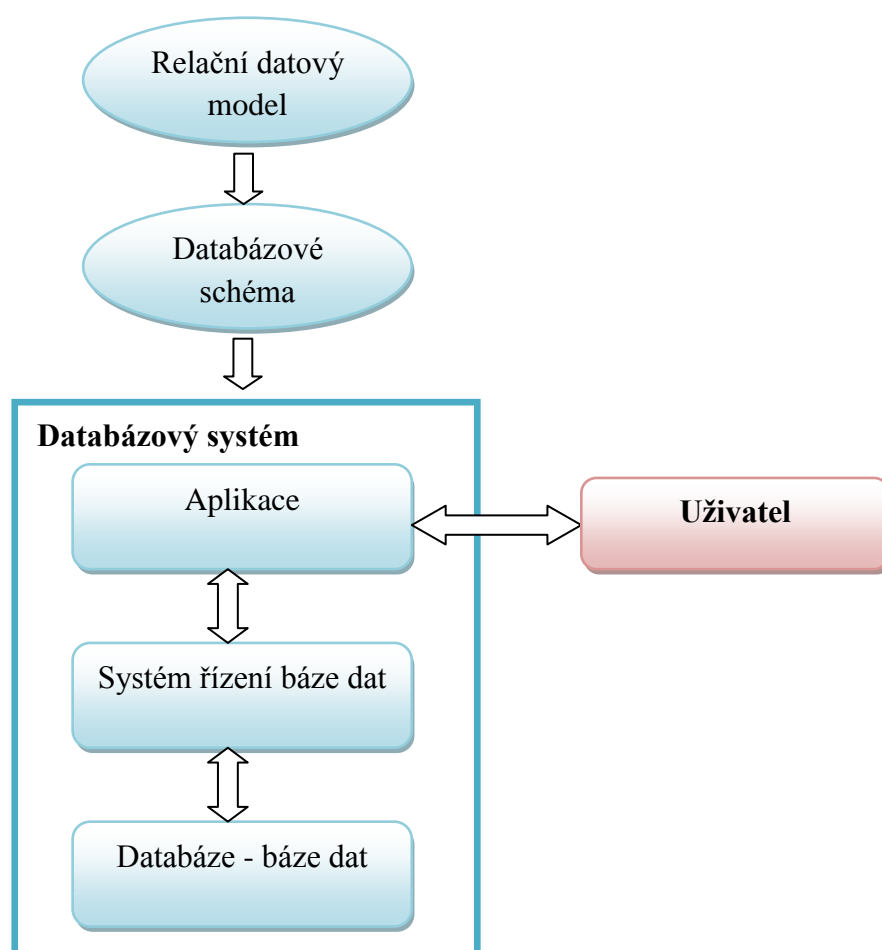
Původní projekt Ingres se přeměnil roku 1985 do projektu Postgres, jehož snahou bylo vytvořit relačně objektovou databázi. Projekt Postgres trval až do roku 1994 a byl veden Michaellem Stonebrakerem. Po ukončení se projektu zhostili dva studenti univerzity v Berkeley Jolly Chen a Andrew Yu, kteří ho přejmenovali na Postgres95, v open source podobě je v následujícím roce přejmenován na PostgreSQL a je vyvíjen dodnes. University of California at Berkeley se i nadále soustředila na vývoj původního Postgresu v projektu Mariposa. Tento SQL server se experimentálně soustředil hlavně na replikace v sítích s 10 000 uzly.

SQL databáze nejsou odvětvím, kde by denně byla vyvinuta nová verze a její nové vlastnosti. Vývoj podobných projektů trvá 10–15 let. Samotné dnes existující standardy představují dostatečnou inspiraci v tom, co „dál“.

Teorie relačních databází je základním kamenem většiny dnešních databázových systémů. Dnes se již výhradně používají relační databáze, přestože se ještě ve světě využívají hierarchické i síťové. Při zmínce o databázové technologii se obvykle míní technologie relační.

## 1.2 Relační databázový systém

Relační databázový systém (RDBMS) je systém založený na relačním datovém modelu. Relační databáze jsou nejrozšířenějším typem databázových systémů a často nahrazují starší hierarchické a síťové databáze. Mezi nejvýznamnější společnosti v oblasti relačních databází patří Oracle, IBM, Microsoft a mezi jejich nejznámější zástupce produkty MySQL, Oracle Database, DB2 a MS SQL. Nedílnou součástí relačního databázového systému je systém řízení báze dat (SŘBD).



Obrázek 1.2 - Relační databázový systém

### 1.2.1 Systém řízení báze dat

Systém řízení báze dat bývá zkracováno na SŘBD nebo se lze setkat i s ekvivalentním názvem DBMS (database management system). Jedná se o softwarové vybavení zajišťující práci z bází dat, které tvoří rozhraní mezi aplikací a databází. Díky této vrstvě uživatel nemusí vědět nic o skutečné fyzické podobě a způsobu uložení dat.



## Základní funkce SŘBD

- Manipulace s daty,
- zajištění integrity a bezpečnosti dat,
- definice dat v databázi,
- souběžný přístup (transakční zpracování) a zotavení po chybách,
- zajištění co nejvyšší výkonnosti.

### 1.2.2 Nejrozšířenější produkty

#### MySQL

MySQL je zkratka z anglického názvu My Structured Query Language. Je to databázový systém původně vytvořený švédskou firmou MySQL AB, v dnešní době ve vlastnictví společností Sun Microsystems jakožto dceřinná společnost Oracle Corporation. MySQL nabízí jak bezplatnou licenci GPL<sup>1</sup>, tak i placenou komerční licenci. Databáze MySQL je jedna z prvních hojně rozšířených systémů. Využití tohoto systému můžeme například při práci s PHP, JAVA, C, C++, Perl, atd. Jeho největší předností je výborná podpora ze strany PHP, proto většina hostingových společností nabízí jeho kombinaci. Nejnovější verze MySQL je 5.6.

Pro správu MySQL se využívají nástroje PhpMyAdmin a Adminer.

Adminer je odlehčená alternativa PhpMyAdmin, nabízející jednoduchou správu databáze. Vzhledem ke své jednoduchosti je Adminer v průměru zhruba 2,5x rychlejší než jeho konkurent PhpMyAdmin. Adminer je napsán českým vývojářem a je šířen jako jediný zdrojový skript pod licencí GNU a Apache.



Obrázek 1.3 - Adminer logo

---

<sup>1</sup> GPL - Licence pro svobodný software.

PhpMyAdmin umožňuje pokročilou správu databáze pomocí webového rozhraní. Jedná se o nejpoužívanější nástroj pro správu MySQL a je vyvíjen pod licencí GPL.



Obrázek 1.4 - PhpMyAdmin logo

## Oracle Database

Jak již název napovídá, jde o produkt od společnosti Oracle Corporation. Ve své kategorii je jedničkou a pokrývá téměř 50% trhu. Oracle Database se využívá především ve středních a velkých firmách. Jedná se o moderní multiplatformní systém s vysokým výkonem a pokročilými možnostmi při zpracování dat. Systém samozřejmě podporuje standardní relační dotazovací jazyk SQL a dále pak jeho procedurální nadstavbu PL/SQL. Obsahuje podporu pro objektové databáze a také databáze, které jsou uloženy v hierarchickém modelu dat. Za číselným označením verze je i písmeno, značící, na kterou oblast je systém zaměřen. Například písmeno g značí, že systém obsahuje sadu nástrojů pro snadné nasazení na gridových sítích. Aktuální verzí je Oracle Database 12.



Obrázek 1.5 - Oracle logo

Pro správu Oracle databáze je možné využít bezplatný program s grafickým prostředím pro vytváření databáze, SQL Developer. Ten umožňuje jednoduché prohlížení databázových objektů, spouštění SQL příkazů a skriptů. Dále nabízí práci s PL/SQL příkazy, které se dají editovat a ladit. SQL Developer zjednodušuje práci s databází a zvyšuje produktivitu.

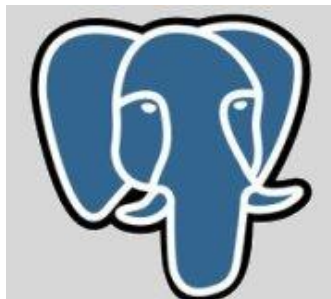
## PostgreSQL

Relační databázový systém, který je vydáván jako open source software<sup>2</sup> a na jeho vývoji se podílí globální skupina vývojářů a firem, se nazývá PostgreSQL. PostgreSQL je

---

<sup>2</sup> Software s otevřeným zdrojovým kódem.

primárně vyvíjen pro unixové operační systémy, ale existují i balíčky pro Windows. Jeho předností je právě rozšiřitelnost a implementace funkcí za pomoci zabudovaného jazyka PL/pgSQL, kterou je obdoba PL/SQL známá od Oraclu. Systém je možné dále rozšiřovat o nové datové typy, operátory a agregační funkce. Momentálně se nachází ve verzi PostgreSQL 9.2.4.



Obrázek 1.6 - PostgreSQL logo

### 1.3 Databázové jazyky

Většina relačních DBMS využívá dotazovací jazyk. Jedná se o jazyk sloužící jako prostředek pro komunikaci mezi uživateli (aplikací) a SŘBD. Podle způsobu zpracování je rozdělujeme na dvě základní skupiny: procedurální a neprocedurální. Procedurální znamená posloupnost operací, jak chceme daného výsledku dosáhnout. U neprocedurálního způsobu se manipuluje s daty pomocí dotazování.

Zástupcem neprocedurálního jazyka je jazyk SQL, QBE a procedurálního PL/SQL.

#### 1.3.1 SQL

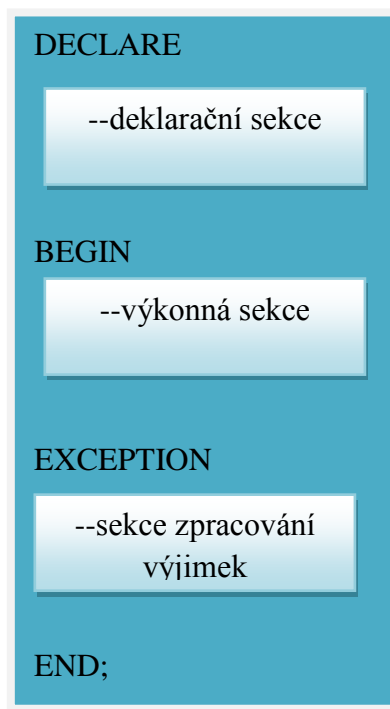
Standardní příkazy jazyka SQL se dělí do několika skupin: příkazy pro definici dat, pro manipulaci s daty, pro řízení dat a ostatní příkazy. V současnosti se používá verze SQL92 a označuje se jednoduše jako jazyk SQL.

- **Příkazy pro definici dat**
  - Zkráceně DDL (Data Definition Language).
  - Slouží pro vytváření struktury databáze.
  - Vytvořené skupiny lze také upravovat, doplňovat nebo mazat.
  - Patří sem příkazy CREATE, ALTER a DROP.
  
- **Příkazy pro manipulaci s daty**
  - Zkráceně DML (Data Manipulation Language).
  - Pro získávání dat z databáze a jejich úpravu.
  - Patří sem příkazy SELECT, UPDATE, INSERT a DELETE.

- **Příkazy pro řízení dat**
  - Zkráceně DCL (Data Control Language).
  - Pro nastavování přístupových práv, řízení provozu a údržby databáze.
  - Patří sem příkazy GRANT, REVOKE, ALTER USER a DROP USER.
- **Příkazy pro řízení transakcí**
  - Zkráceně TCC (Transaction Control Commands).
  - Patří sem příkazy COMMIT, ROLLBACK, SAVEPOINT a SET TRANSACTION.

### 1.3.2 PL/SQL

Firma Oracle vyvinula procedurální nadstavbu jazyka SQL, kterým je PL/SQL. Ten přidává konstrukce procedurálního programování jako například cykly, podmínky, procedury nebo funkce. Základní stavební kámen v PL/SQL je blok. Bloky mohou být vnořeny jeden do druhého a tvoří program v PL/SQL. Každý vytvořený blok spouští jednu logickou akci v programu.



Obrázek 1.7 - Programový blok

Deklarační sekce slouží pro deklaraci proměnných, cursorů nebo konstant. Mohou se zde vyskytovat i uživatelem deklarované výjimky.

Ve výkonné sekci se nachází funkční logika a je to aplikační jádro programového bloku. Pomocí SQL příkazů v této sekci můžeme manipulovat s daty v databázi. PL/SQL příkazy manipulují s daty v programovém bloku.

Sekce zpracování výjimek obsahuje ošetření chybových stavů, které mohou nastat ve výkonné sekci.

## 2 Relační datový model

Tato část se zaměří na relační datový model, bude se zabývat pravidly pro jeho tvorbu a základními vlastnostmi, které jsou důležité při návrhu.

### 2.1 Coddova pravidla

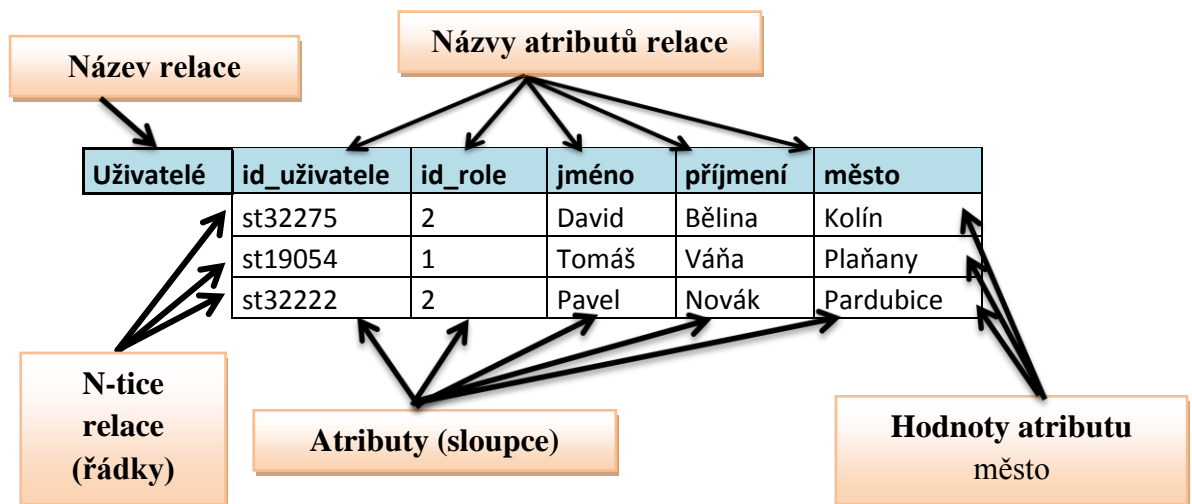
Autorem relačního modelu je Edgar Frank Codd a níže uvedených dvanáct Coddových pravidel jsou tzv. měřítkem, zda je daný databázový model skutečně "relačním". Tyto pravidla se dají ještě rozdělit do pěti skupin na základní pravidla, pravidla týkající se integrity, modifikace, struktury dat a poslední skupina o nezávislosti dat.

1. *„Informační pravidlo: všechna data musejí být reprezentována jako hodnoty v tabulkách.*
2. *Pravidlo zajišťující přístup: každá hodnota musí být dosažitelná pomocí názvu tabulky, názvu sloupce a klíče.*
3. *Zpracování neznámých hodnot: neznámé (NULL) hodnoty jsou podporovány pro vyjádření neznámé informace, a to nezávisle na datovém typu.*
4. *Popis relačního katalogu: Popis celé databáze je na logické úrovni reprezentován stejně jako běžná data, tedy také jako tabulka.*
5. *Pravidlo pro jazyk: musí existovat alespoň jeden jazyk, který podporuje DDL, DML, integritní omezení, práci s transakcemi a autorizaci.*
6. *Pravidlo pohledů: všechny pohledy, které jsou teoreticky aktualizovatelné, jsou také systémem aktualizovatelné.*
7. *Pravidlo operací: schopnost zpracování relace jako operandu je zachována nejen při čtení dat, ale i u vkládání, aktualizace a odstranění dat.*
8. *Pravidlo fyzické nezávislosti dat: výsledky operací jsou nezávislé na fyzické datové struktuře.*
9. *Pravidlo logické nezávislosti dat: výsledky operací jsou nezávislé na změně logické struktury.*
10. *Pravidlo nezávislosti dat na integritních omezeních: integritní omezení musí být definovatelná prostředky relační databáze a musí být uchovatelná v katalogu a ne v aplikačním programu.*
11. *Pravidlo nezávislosti dat na distribuci: výsledky operací jsou nezávislé na konkrétním rozmístění dat v distribuované databázi.*

12. Pravidlo nenarušitelnosti SŘBD: žádný uživatel (ani aplikace) nesmí obcházet ani narušovat rozhraní SŘBD.<sup>3</sup>

## 2.2 Princip relačního modelu

Relační model definuje způsob reprezentace dat, jejich ochranu (integritu dat) a operace, které nad daty můžeme provádět. Zakládá se na základních matematických principech, odvozených z teorie množin a na predikátové logice prvního řádu. Jeho základním konstruktem je relace.



Obrázek 2.1 - Relace

- **Atribut** - Konkrétní informace o subjektu (např. identifikační číslo, jméno, příjmení a město).
- **N-tice relace** – Představuje řádky v relaci (tabulce).
- **Relace** - Jedná se o množina prvků (tabulky).
  - Neexistují duplicitní n-tice.
  - Všechny atributy relace jsou atomické a dále nedělitelné.
  - Libovolné pořadí atributů i n-tic.

<sup>3</sup> KAPOUN, Jan. Průkopníci informačního věku (30.) - Edgar Codd. In: *Business World* [online]. Praha: IDG Czech, 2012 [cit. 2013-07-25]. Dostupné z: <http://businessworld.cz/analyzy/prukopnici-informacniho-veku-30-edgar-codd-10089>.

- **Integritní omezení** – Logické podmínky, které musí být na datech splněny.
  - Referenční integrita.
  - Specifikace klíče relace.

## 2.3 Základní relační operace

Základní relační operace se dají ještě dále rozdělit na dvě skupiny, na klasické množinové operace a speciální relační operace.

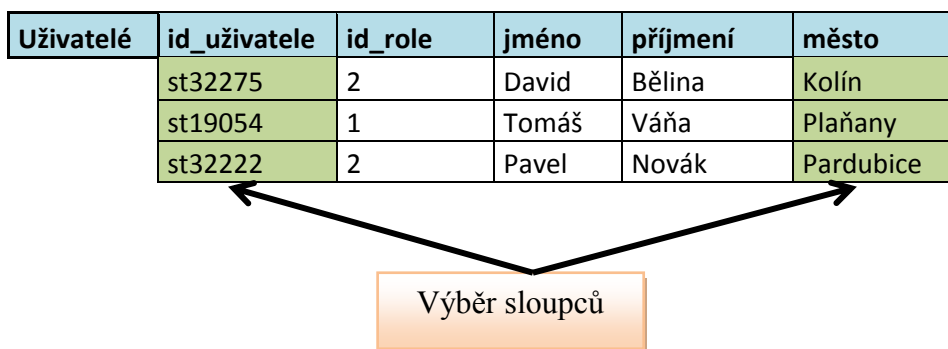
### Klasické množinové operace

Pro klasické množinové operace (kromě kartézského součinu) se předpokládá, že obě tabulky vstupující do operace jako operandy jsou stejného stupně (stejný počet sloupců), dále položky z obou tabulek jsou definovány ve stejném pořadí a jsou stejného typu.

- **Sjednocení** - Vytvoří novou tabulku, obsahující všechny řádky obou výchozích tabulek.
- **Průnik** - Vytvoří tabulku, která obsahuje pouze totožné řádky z obou tabulek.
- **Množinový rozdíl** - Vytvoří tabulku, která bude obsahovat všechny řádky první vstupní tabulky kromě těch, které se vyskytují i v druhé tabulce.
- **Symetrický rozdíl** - Vytvoří tabulku, která bude obsahovat všechny řádky obou tabulek kromě těch, které jsou stejné.
- **Kartézský součin** - Vytvoří tabulku spojením řádků obou tabulek systémem, každý s každým.

### Speciální relační operace

- **Projekce** – Jedná se o výběr sloupců tabulky – vertikální množina.



Obrázek 2.2 – Projekce



- **Restrikce (selekce)** - Výběr řádků tabulky na základě stanovené podmínky – horizontální podmnožina.

Uživatelé	id_uzivatele	id_role	jméno	příjmení	Město
	st32275	2	David	Bělina	Kolín
	st19054	1	Tomáš	Váňa	Plaňany
	st32222	2	Pavel	Novák	Pardubice



Obrázek 2.3 – Restrikce

- **Spojení** - Jde o propojení tabulek, mající alespoň jednu stejnou položku.

## 2.4 Vztahy mezi relacemi (tabulkami)

Slouží pro spojení spolu souvisejících dat, které jsou umístěny v různých relacích (tabulkách). Každý vztah se charakterizuje pomocí třech základních vlastností stupněm, kardinalitou a parcialitou vztahu. Stupeň vztahu může být unární nebo binární a pomocí kardinality se vyjadřuje skutečnost, kolik n-tic jedné relace může vstoupit do vztahu s kolika n-ticemi druhé relace. Parcialitou neboli členstvím ve vztahu se určuje, zda je účast ve vztahu povinná nebo volitelná.

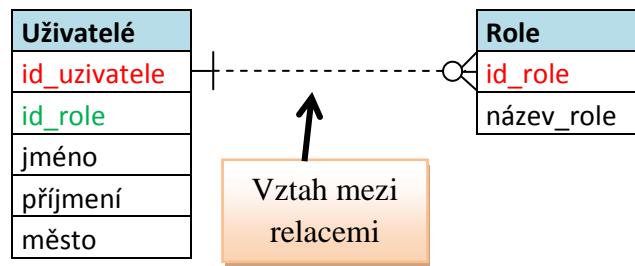
### Stupně vztahu:

- **Unární**
  - Vazba sama na sebe.
- **Binární**
  - Mezi dvěma relacemi.

### Kardinalita vztahu:

- **Vztah 1:1**
  - Situace, kdy jedna n-tice z jedné relace odpovídá právě jedné n-tici z relace druhé.

- **Vztah 1:N**
  - Situace, kdy jedna n-tice z jedné relace může odpovídat více n-ticím z relace druhé.
- **Vztah M:N**
  - Situace, kdy více n-tic z jedné relace může odpovídat více n-ticím z relace druhé.



Obrázek 2.4 - Vztah mezi relacemi 1:N

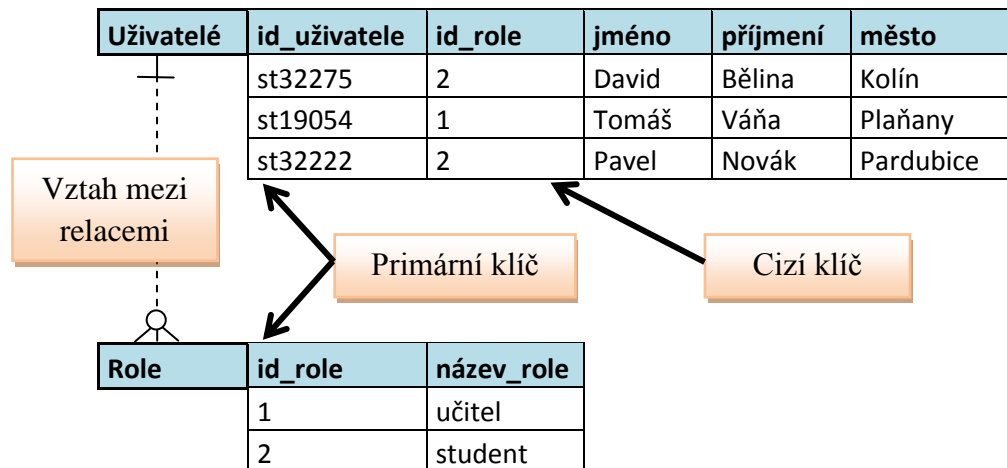
Na Obrázku 2.4 je vidět binární stupeň vztahu mezi dvěma relacemi. Obrázek ukazuje kardinalitu vztahu reprezentovanou 1:N, protože jedna role může být přiřazena více uživatelům.

## 2.5 Klíče

- **Kandidátní klíč**
  - Slouží k jednoznačnému určení n-tice relace (řádku).
  - Každá tabulka musí mít minimálně jeden kandidátní klíč.
  - Může se stát primárním klíčem.
- **Primární klíč**
  - Jeden z kandidátních klíčů, ostatní se nazývají alternativní klíče.
  - Základním prostředek k adresaci n-tic.
  - Musí definovat hodnotu, nesmí být NULL.

- Cizí klíč

- Každá hodnota cizího klíče je buď plně zadána nebo nezadána, tedy NULL.
- Určuje, které n-tice z různých relací spolu souvisí.



Obrázek 2.5 - Vztahy a klíče

Na Obrázku 2.5 je vidět vztah mezi relacemi, dále pak primární klíč relace Uživatelé s názvem atributu id\_uživatele a cizí klíč id\_role, který vznikl ze vztahu s relací Role na základě jejího primárního klíče id\_role.

## 2.6 Normalizace

Normalizace je posloupnost kroků, při kterých se provádí postupná dekompozice tabulek do vhodnějšího tvaru, aby se odstranila redundance (opakování informací) a byly zachovány závislosti a bezztrátovost při zpětném spojení.

Definice: „Normalizace je technika používaná pro vytvoření sady tabulek s minimální redundancí, která podporuje datové požadavky organizace“.<sup>4</sup>

### Přehled normálních forem:

- 0. normální forma (0.NF)

- Relace je v nulté normální formě právě tehdy, existuje-li alespoň jeden atribut, který obsahuje více než jednu hodnotu.

<sup>4</sup> ŽÁK, David.: Databázové systémy I – Přednáška č. 10. Pardubice: Univerzita Pardubice. 2012. [Citace: 24. 7. 2013].

- **1. normální forma (1.NF)**
  - Relace je v první normální formě, když všechny její hodnoty jsou atomické.
- **2 normální forma (2.NF)**
  - Relace je v druhé normální formě, když je v první normální formě a každý neklíčový atribut je plně funkčně závislý na primárním klíči.
- **3. normální forma (3.NF)**
  - Relace je v třetí normální formě, když je v druhé normální formě a všechny neklíčové atributy jsou vzájemně nezávislé.
- **Boyce-Coddova normální forma (BCNF)**
  - Někdy se nazývá jako 3,5 NF.
  - Relace je v Boyce-Coddově normální formě, pokud mezi kandidátními klíči není žádná funkční závislost.
- **4. normální forma (4.NF)**
  - Relace je ve čtvrté normální formě, je-li ve třetí normální formě, a pokud atributy v ní obsažené popisují pouze jeden fakt nebo jednu souvislost.
- **5. normální forma (5.NF)**
  - Relace je v páté normální formě, pokud je ve čtvrté normální formě a nelze do ní přidat nový atribut nebo skupinu atributů tak, aby se vlivem skrytých závislostí rozpadla na několik dílčích tabulek.

### 3 Předmět činnosti implementovaného systému

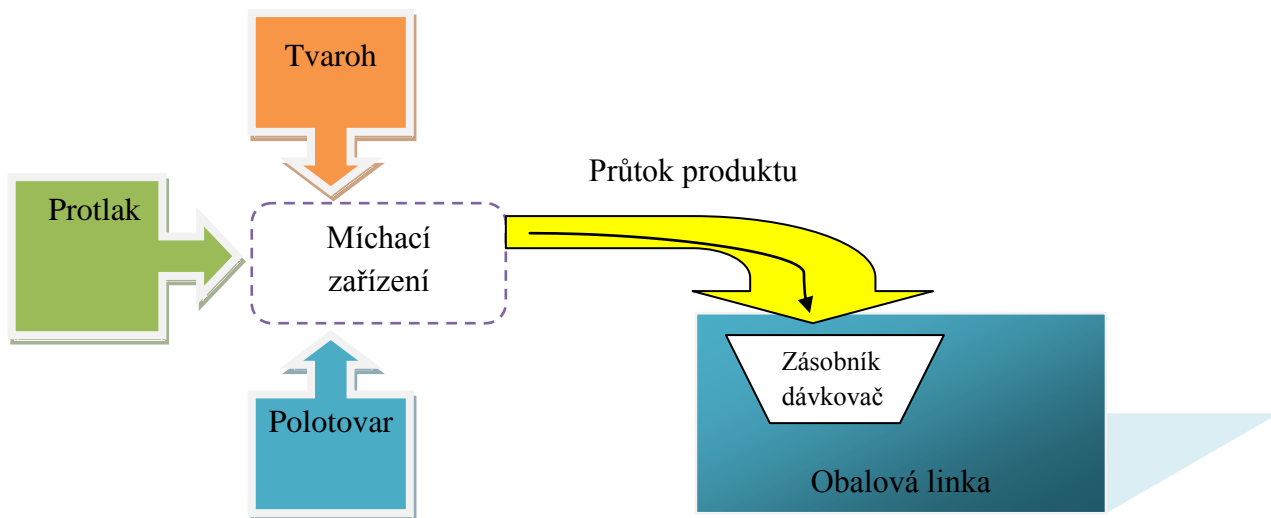
V této kapitole se budu zabývat činnostmi, které přiblíží danou problematiku výroby a jsou klíčové pro realizaci daného řídicího systému.

#### 3.1 Proces výroby

Nyní se zmíním o procesu výroby, na jehož základě dojde k zabalení finálního produktu. Budu se soustředit na popsání výrobního procesu pomocí obalové linky neboli baličky a míchacího zařízení.

##### 3.1.1 Proces míchání

Základním produktem mlékárenské společnosti je výroba z tvarohu. Je potřeba zajistit jeho přísun do obalové linky. Dále se podle určitého druhu výrobku přimíchávají ovocné složky (protlaky) a pro zvýšení tučnosti i smetana (polotovar). Tyto složky jsou odčerpávány z kontejnerů. Pro správný přísun finálního produktu do baličky se využívá míchací zařízení, kde se po připojení kontejnerů všech potřebných složek nastaví poměr jejich smíchání. Míchací zařízení je propojené s obalovou linkou a pomocí cesty se produkt doplňuje do zásobníku dávkovače Obrázek 3.1.

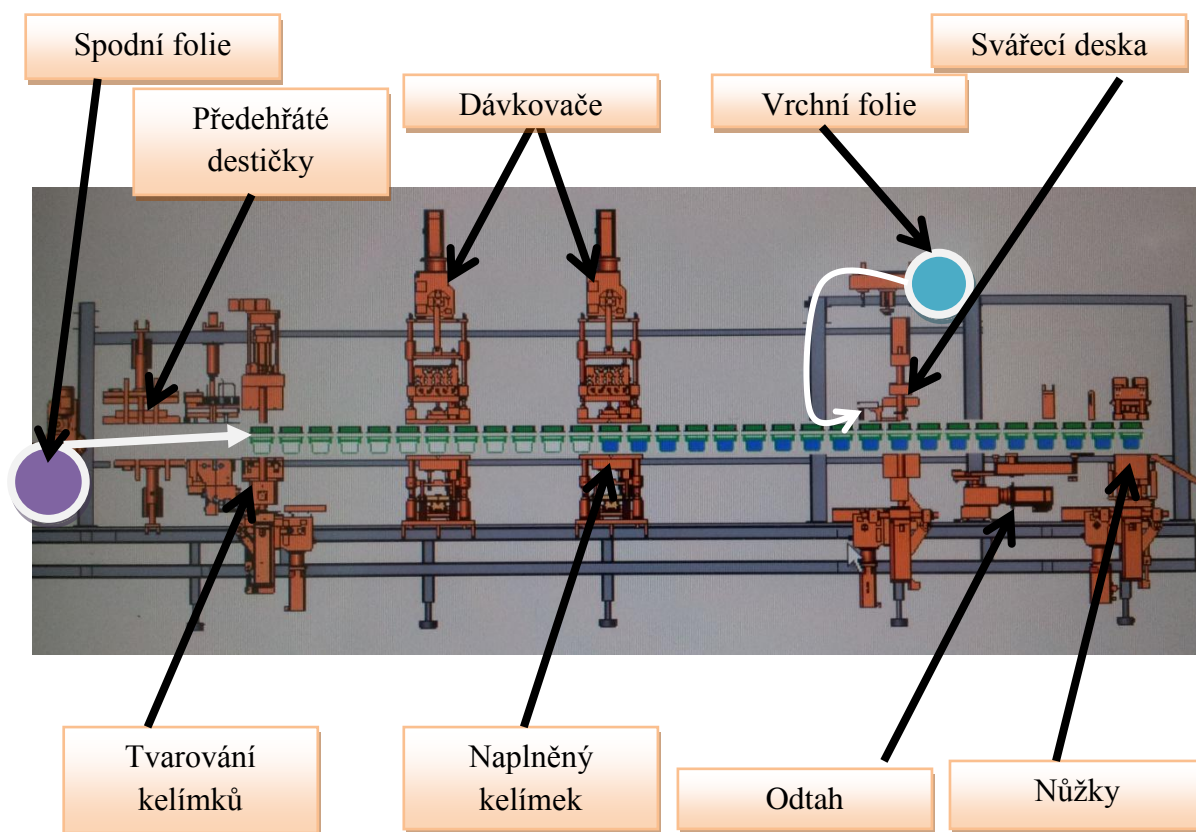


Obrázek 3.1 - Proces míchání

##### 3.1.2 Balení produktu

Pokud je nastaven správný přísun produktu pomocí míchacího zařízení, je třeba dále připravit obalovou linku. Obalová linka balí tvarohy do kelímku. Kelímky na produkt se vytváří z polystyrenových folií (spodní folie) zkráceně PS. Spodní folie je pomocí odtahu po určitých úsecích posouvána. Nejdříve se přehřáté destičky přitlačí na spodní folii

a tím získá teplotu kolem 150 stupňů a poté se pomocí tvarovací formy a trnů „vyfoukne“ kelímek. V další fázi balení se dávkovačem do kelímku naplní produkt a následně se na horní část kelímku za pomoci svářecí desky, která má teplotu přes 200 stupňů, přivaří hliníková folie (vrchní folie), na které jsou informace o výrobku. K vrchní folii se natiskne injetem datum spotřeby. Nakonec se nůžkami finální kelímek oddělí od zbytku folie a vloží do krabice. Krabice se po naplnění ukládá na paletu.



Obrázek 3.2 - Obalová-balící linka

## 3.2 Skladové zásoby

Pro evidování materiálu ve výrobě je nutné předem zadávání skladových zásob pomocí uživatelské role skladník. Evidování údajů do systému zásob probíhá na základě evidenčních štítků u každého materiálu nebo kontejneru.

### 3.2.1 Vrchní folie

Podle evidenčního štítku, který je na každé vrchní folii, se zapisují údaje o hmotnosti, čísla zakázky, čísla role, data výroby materiálu, šířky (rozměry) folie.



Obrázek 3.3 - Evidenční štítek vrchní folie

### 3.2.2 Spodní folie

U spodní folie se eviduje barva folie, hmotnost, číslo zakázky, číslo role, datum výroby materiálu, šířka a tloušťka folie.



Obrázek 3.4 - Spodní folie

### 3.2.3 Polotovary

Polotovary jsou zásoby, které podnik sám vyrábí a následně dále zpracovává. Pro evidování polotovarů do skladových zásob je nezbytné datum výroby, čistá hmotnost (netto), druh polotovaru, číslo kontejneru a jméno, kdo polotovar vyrobil.

### 3.2.4 Protlaky

Protlaky jsou zásoby, které jsou dováženy na zakázku, proto je nutné mít dostatečný přehled o jejich množství a trvanlivosti. U protlaků je nutné evidovat datum výroby společně s datem spotřeby, dále pak o jaký druh protlaku se jedná, číslo kontejneru a jeho čistou hmotnost (netto).



**Obrázek 3.5 - Připojené kontejnery ve výrobě**



## 4 Implementace aplikace

### 4.1 Přehled požadavků a základních funkcí v aplikaci

Mým úkolem je vytvořit webovou aplikaci, která uchovává evidovaná data (zmíněná v předcházející kapitole) v databázovém systému a pomocí webového rozhraní slouží zaměstnancům k evidenci vyrobených palet, použitého materiálu, skladových zásob a k přehledu o výrobě.

Základními požadavky na aplikaci jsou:

- Evidence skladových zásob,
- sledování a nastavování provozu jednotlivých balíčních linek,
- evidence a sledování kvality jednotlivých obalových sérií,
- reklamace od zákazníků,
- tvorba periodických reportů - přehledů.

Navržený databázový systém bude schopný provozu na více balíčních zařízeních a bude obsahovat několik uživatelských rolí, které budou obsahovat odlišné sady funkcí v systému.

### 4.2 Použité technologie a prostředky pro vývoj

Statická část webové stránky je kódována v jazyce HTML a k dosažení grafického návrhu je použito kaskádových stylů CSS. Pro tvorbu dynamické webové stránky jsem zvolil skriptovací jazyk PHP, který bude komunikovat mezi klientem a centrálním úložištěm dat (databází). Klientem se rozumí webová aplikace. Jako vývojové prostředí pro tvorbu aplikace jsem zvolil NetBeans IDE 7.3.1<sup>5</sup>. Dalším neméně důležitým krokem byl výběr relační databáze, k tomuto účelu jsem vybral MySQL 5.1<sup>6</sup>. Dále k návrhu relačního datového modelu a následného generování DDL skriptu pro vytvoření databáze jsem využil freeware verzi Toad data modeler 4.3<sup>7</sup>. Volba webového serveru probíhala na základě svobodné licence s možností použití MySQL a PHP modulu, proto byl vybrán Apache 2.2.22<sup>8</sup>. Apache je možné nainstalovat jak na unixových operačních systémech, tak v systémech Microsoft Windows. Další nezbytnou součástí byla volba softwaru pro správu databáze, zvolen byl PhpMyAdmin 3.5.1<sup>9</sup>.

---

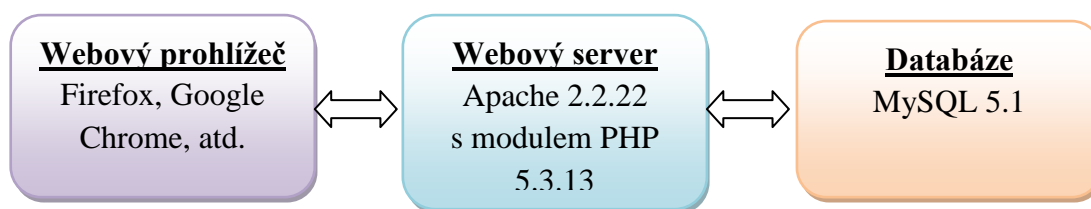
<sup>5</sup> <https://netbeans.org/>

<sup>6</sup> <http://www.mysql.com/>

<sup>7</sup> <http://www.quest.com/toad-data-modeler/>

<sup>8</sup> <http://httpd.apache.org/>

<sup>9</sup> [http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php)



Obrázek 4.1 - Vazba jednotlivých částí systému

### 4.3 Uživatelské role

Pro každý informační systém jsou důležité uživatelské role. Uživatelské role zpravidla představují určité zaměstnanecké pozice ve firmě. Každá role poskytuje patřičná oprávnění, umožňující vykonávat specifické činnosti v systému.

Mnou navržený systém bude také využívat několika uživatelských rolí. Základní role pro evidenci zásob je skladník, dále pro výrobu palet strojník ve výrobě a několik dalších řídicích rolí jako mistr, vedoucí výroby nebo výrobní ředitel. Poslední rolí je administrátor, který bývá nedílnou součástí většiny systému.

#### 4.3.1 Skladník

Přijímá materiál od dodavatelů, tedy protlaky, vrchní a spodní folie, atd. a zapisuje ho do systému. Na základě požadavků od mistrů doplňuje zásoby ze skladů do výroby.

#### 4.3.2 Strojník ve výrobě

Úlohou strojníka je podle denního plánu vyrobit patřičné množství palet výrobku, zapisovat vyrobené palety s evidenčním číslem, evidovat použitý obalový materiál a kontejnery u každé palety. Dále musí zapisovat spotřebu a zůstatek při vyrábění produktu pomocí protlaku, případně polotovaru z kontejneru a udávat jeho stav - umístění.

#### 4.3.3 Mistr ve výrobě

Mistr ve výrobě sestavuje denní rozpis na základě požadavku od vedoucího výroby. Informuje skladníka, aby zajistil dostatek materiálu a produktu pro výrobu. Řeší problémy ve výrobě a je schopný vykonávat roli strojníka ve výrobě.

#### 4.3.4 Vedoucí výroby

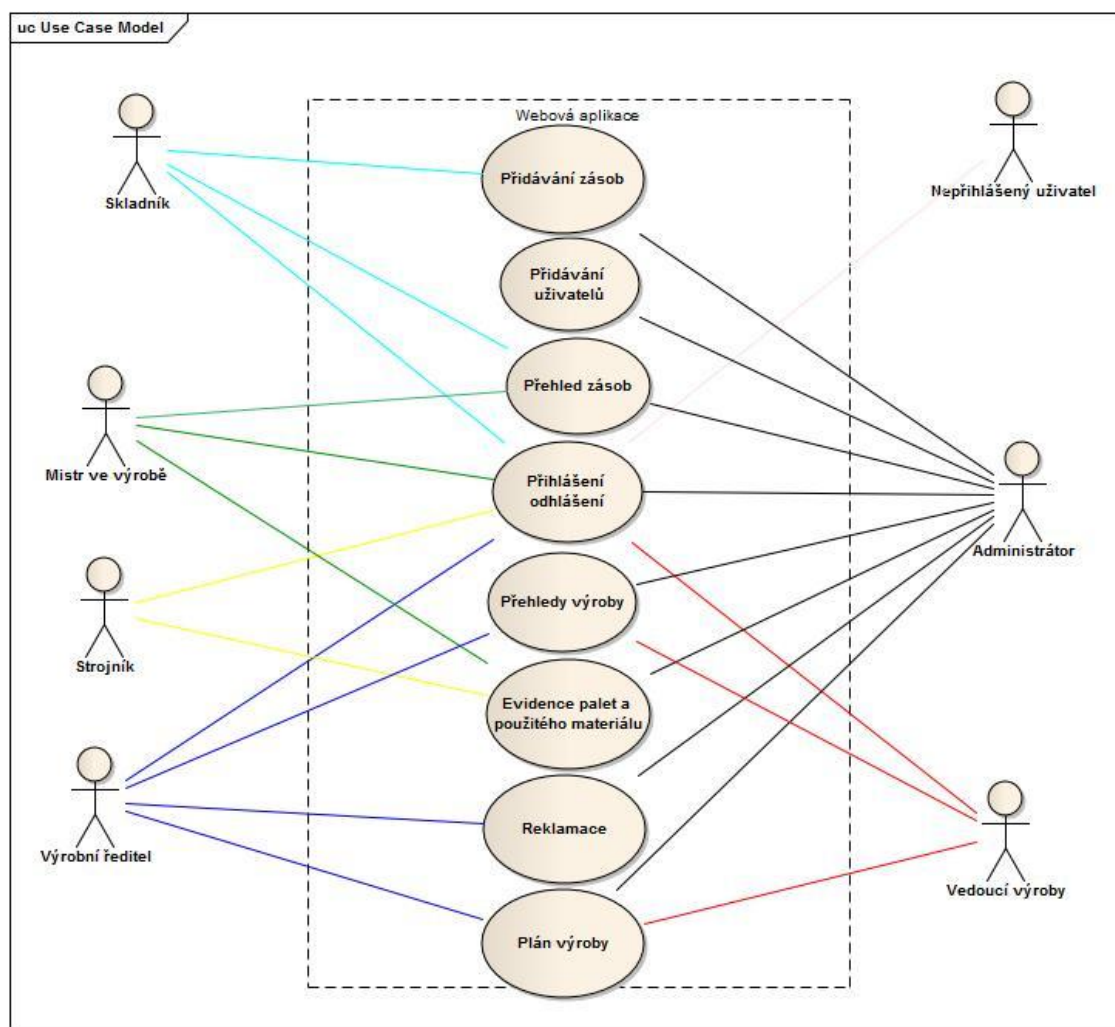
Na základě poptávky po produktech určuje množství vyrobených palet pro jednotlivé dny. Sleduje statistiky vyrobených palet, závady na balících linkách, reklamace a množství spotřebovaných protlaků.

### 4.3.5 Výrobní ředitel

Je zodpovědný za optimální chod ve výrobě. Řeší složitější problémy a stejně jako vedoucí výroby může sledovat statistiky o výrobě. Jeho úkolem je i evidovat a schvalovat reklamace od odběratelů a zjišťovat jejich příčiny.

### 4.3.6 Administrátor

Administrátorská role nabízí všechny části - činnosti v systému. Umožňuje přidávání nových uživatelů, manipulaci s databází a upravování uživatelských účtů a oprávnění.



Obrázek 4.2 - Use case diagram pro uživatelské role

## 4.4 Evidence klíčových záznamů při výrobě

Při výrobě je nezbytná evidence důležitých údajů o použitých materiálech a surovinách, některé je možné vyvodit z kapitoly *Proces výroby* a nad jinými je nutné se zamyslet. Z procesu míchání je patrné, že je třeba evidovat základní složku tvaroh a v případě použití dalších složek i protlak či polotovar. Z balení produktu je nutné evidovat spodní a vrchní folii. Dále jsem při požadavku na evidování vycházel z tištěných formulářů, které se ve výrobě používají. U následujících podkapitol uvedu příslušné tištěné formuláře.

### 4.4.1 Evidence palet

Je nutné evidovat údaje obsažené v Tabulce 4.1 pro každou vyrobenou paletu. Jedná se o evidování unikátního evidenčního štítku palety (kód SCSC) a zapisování vyráběného druhu. Je nutná průběžná kontrola váhy kelímku alespoň jednou za výrobu palety a evidování času pro začátek vyrábění palety.

Přehled o vyrobených paletách									Datum:	
Datum spotřeby	Výrobek Druh	Hmotnost g	číslo trysky				Paleta		Kartonů /kusů	Podpis
			1	2	3	4	číslo	SCSC		

Tabulka 4.1 - Formulář evidence palet

### 4.4.2 Evidence polotovarů

Mezi další důležité aspekty patří evidování použitého kontejneru polotovaru na základě formuláře z Tabulky 4.2. Zadává se číslo kontejneru, druh příchuti, aktuální čas při připojení.

Evidence spotřeby polotovarů					
Datum	Příchuť	Spotřeba	Číslo kontejneru	Čas připojení	Podpis

**Tabulka 4.2 - Formulář polotovarů**

#### **4.4.3 Evidence protlaků**

Evidování protlaku je obdobné jako u polotovaru podle údajů obsažených v Tabulce 4.3, zadává se číslo kontejneru, aktuální čas připojení a evidenční číslo palety, u které byl protlak připojen.

Evidence spotřeby protlaků					
Datum	Příchuť	Spotřeba	Číslo kontejneru	Čas připojení	Podpis

**Tabulka 4.3 - Formulář protlaků**

#### **4.4.4 Evidence spodní folie**

Spodní folie je základním obalovým materiálem pro výrobu kelímků. Evidování probíhá na základě evidenčního čísla zakázky a čísla role u folie. Dále se zapisuje i číslo palety, u které byla folie použita.

#### **4.4.5 Evidence vrchní folie**

Vrchní folie specifikuje přesné údaje o vyrobeném produktu, dává zákazníkům informace o složení a gramáži výrobku. Potřebné údaje pro evidování při výrobě jsou uváděny podle Tabulky 4.4, která je společná i pro evidenci spodní folie. Zapisuje se číslo zakázky, číslo role a evidenční číslo palety, u které byla folie použita.

Evidence použitého obalového materiálu						
Datum	Materiál	Název výrobku	Datum výroby materiálu	Číslo zakázky /LOT	Číslo role	Podpis

Tabulka 4.4 - Formulář obalového materiálu

#### 4.4.6 Evidence prostojů

Evidence prostojů je další údaj nutný k zapisování. Prostoje se dají dále rozdělit na čekání a poruchy.

Poruchy souvisejí se závadou na obalové lince. Eviduje se čas začátku závady a čas konce, neboli opravení závady, dále jméno strojníka, který závadu nahlásil a jméno údržbáře, který provedl opravu. Specifikuje se také druh vzniklé závady.

V kategorii čekání se zapisují prostoje, které vznikají z důvodu čekání na materiál, případně na kontejner a další nespecifikované údaje, které nemají co dočinení s poruchou obalové linky. U čekání se eviduje začátek čekání a ukončení čekání, dále pak specifikace druhu čekání a jméno strojníka, který čekání evidoval.

#### 4.4.7 Evidence doprodeje

Jedná se o evidování doprodeje neboli nestandardních výrobků. Tyto výrobky mohou vzniknout na základě poruchy nebo zapříčiněním lidským faktorem v případě špatného nastavení na obalové lince. Mezi nejčastější nestandardní výrobky patří ujetá folie nebo nízká váha výrobku. Nestandardní výrobky se dále prodávají v podnikové prodejně za sníženou cenu. Při evidenci se uvádí počet nestandardních výrobků, dále důvod jejich 'vyřazení', datum vytvoření a jméno strojníka, který doprodeje vyrobil. Tyto údaje jsou obsažené v Tabulce 4.5.

Evidence doprodejů								
Datum	Výrobek druh	Množství	Důvod				Upřesnění důvodu	Podpis strojník
			Ujetá folie	Nízká váha	Přejezd	Rozjezd		

Tabulka 4.5 - Formulář doprodejů

## 4.5 Databáze

Při návrhu vhodného modelu databáze jsem vycházel z procesu výroby a také zejména z důležitých evidenčních údajů pro výrobu, které byly zmíněny v kapitole *Předmět činnosti implementovaného systému*. Mnou navržený Entity Relationship Diagram, který byl použit pro realizaci databáze, splňuje 3. normální formu.

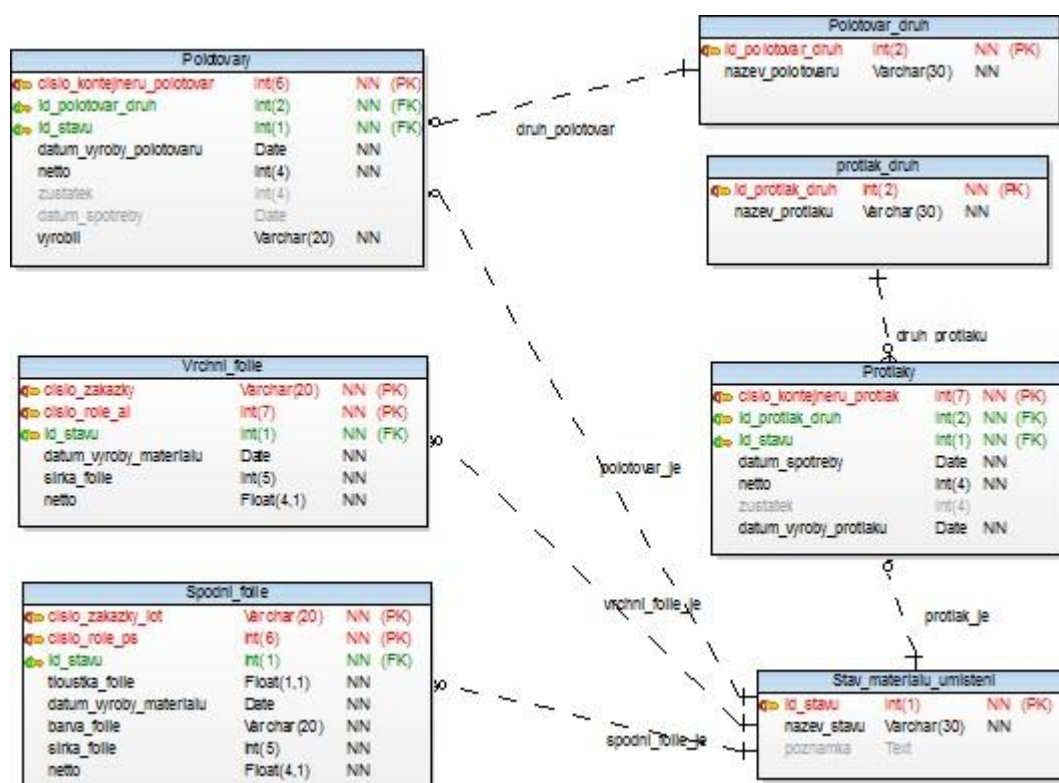
### Ukázka a popis některých vybraných částí databáze:

Vzhledem k tomu, že se jedná o vcelku rozsáhlý E-R Diagram *Příloha A*, provedu jeho rozbor na menší části, které povedou k objasnění a pochopení dílčích problémů.

- Realizace skladových zásob,
- evidence vyrobených palet,
- uživatelé v systému,
- reklamace výrobků.

### Realizace skladových zásob

Evidence skladových zásob je realizována pomocí tabulek *Polotovary*, *Protlaky*, *Vrchni\_folie* a *Spodni\_folie*. Údaje z těchto tabulek nám říkají, o jaký typ zásob se jedná, a definují jejich specifické údaje. Ke každé z těchto tabulek je na základě vztahu přidružena tabulka *Stav\_materialu\_umisteni*, díky které je možné zjistit, kde se dané evidované zásoby nachází, nebo v jakém jsou stavu.



Obrázek 4.3 - Realizace skladových zásob E-R Diagram

### Uživatelé v systému:

Další nezbytnou částí snad v každém systému jsou uživatelé. Pro tyto potřeby byla vytvořena tabulka *Uzivatele* a k ní dále tabulka *Opraveni*. V tabulce *Uzivatele* jsou údaje o uživateli, jaké je jméno, příjmení, přihlašovací údaje a na základě tabulky *Opraveni* také informace o jeho uživatelské skupině a prioritě, která určuje úroveň pravomocí v navržené aplikaci. Výše zmíněný popis je realizován na Obrázku 4.4.

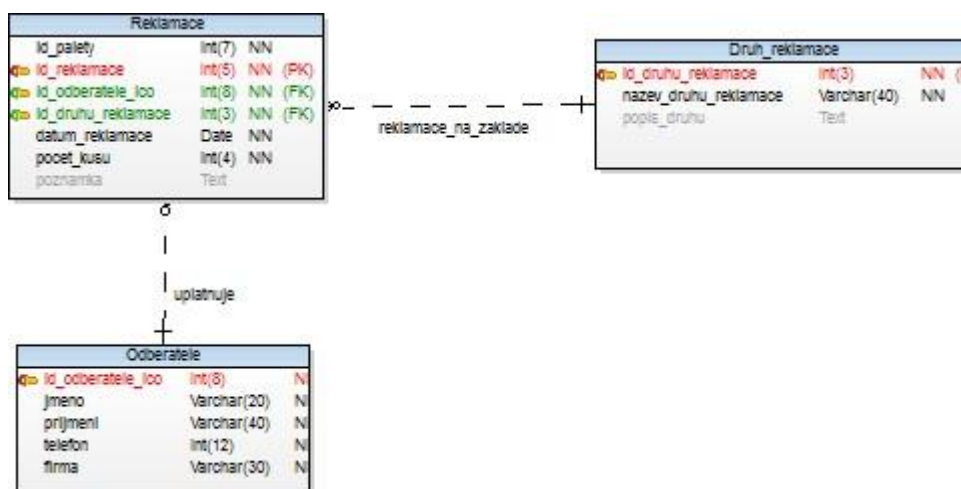


Obrázek 4.4 - Uživatelé E-R Diagram



### Reklamacie od odběratelů:

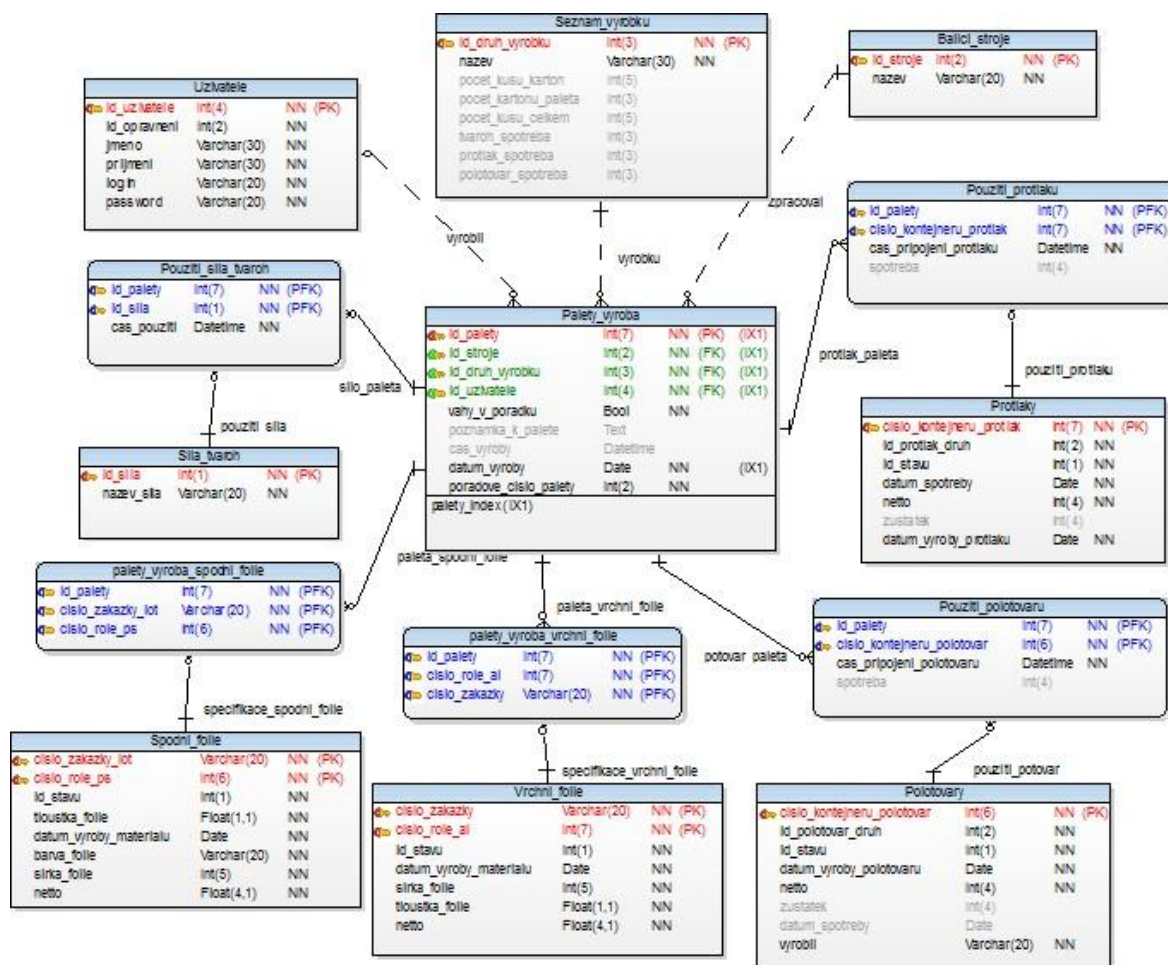
Žádný výrobní proces není stoprocentní, proto může docházet k chybám, které nejsou hned odhaleny. Z tohoto důvodu se může stát, že dojde k reklamacím od odběratelů. Tyto problémy je nutné evidovat, viz Obrázek 4.5. Na základě tabulky *Reklamacie* se zaznamenává, o jaký konkrétní druh reklamce se jedná, kdo reklamaci uplatňuje, z jaké palety reklamovaný výrobek pochází, datum dané reklamce, počet kusů k reklamaci a případné zapsání poznámky. V tabulce *Odberatele* se nachází údaje o odběratelích, jejich IČO, název firmy, telefonní kontakt, dále jméno a příjmení kontaktní osoby z firmy. Tabulka *Druhy\_reklamce* nám blíže specifikuje, na jakém základě se uplatňuje reklamce.



Obrázek 4.5 - Reklamacie E-R Diagram

### Evidence vyrobených palet:

Nejdůležitější částí v mnou navrhovaném systému je evidence vyrobených palet. Pro tento účel slouží jako základ tabulka *Palety\_vyroba*. Vzhledem ke složitosti této vybrané části provedu kompletní popis klíčových tabulek Obrázku 4.6.



Obrázek 4.6 - Evidence výroby palet E-R Diagram

Níže bude proveden popis tabulek *Seznam\_vyrobu*, *Balici\_stroje*, *Palety\_vyroba*, *Pouziti\_protlatku*, *Palety\_vyroba\_vrchni\_folie*, *Palety\_vyroba\_spodni\_folie*, *Pouziti\_polotovar*, *Pouziti\_sila\_tvaroh* a jejich atributů.

- **Seznam\_vyrobu** – V této tabulce je výčet všech výrobků, které je možné vyrábět.
  - **id\_druh\_vyrobu** – Jedná se o primární klíč tabulky *Seznam\_vyrobu*, identifikující výrobek.
  - **nazev** – Specifikuje přesný název výrobku.
  - **pocet\_kusu\_karton** – Udává, kolik je kusů kelímků v jedné krabici.
  - **pocet\_karton\_paleta** – Udává přesný počet kartonů - krabic na paletě.
  - **pocet\_kusu\_celkem** – Udává, kolik je kusů kelímků na paletě.
  - **tvaroh\_spotreba** – Udává, jaké množství tvarohu je potřeba na výrobu jedné palety výrobku.
  - **polotovar\_spotreba** – Udává, jaké množství polotovaru je potřeba na výrobu jedné palety výrobku.

- **protlak\_spotreba** – Udává, jaké množství protlaku je potřeba na výrobu jedné palety výrobku.
- **Balici stroje** – V této tabulce je výčet všech balicích zařízení ve výrobě.
  - **id\_stroje** – Jedná se o primární klíč tabulky *Balici\_stroje*, definující balicí stroj.
  - **nazev** – Specifikuje přesný název balicího stroje.
- **Palety vyroba** – Tato tabulka slouží pro evidenci vyrobených palet.
  - **id\_palety** – Jedná se o primární klíč tabulky *Palety\_vyroba*, definující evidenční číslo palety.
  - **id\_stroje** – Cizí klíč tabulky *Balici\_stroje* slouží pro určení, na jakém balicím stroji byla paleta vyrobena.
  - **id\_druh\_vyrobku** – Cizí klíč tabulky *Seznam\_vyrobku* slouží pro určení, jaký druh výrobku byl vyroben.
  - **Id\_uzivatele** – Cizí klíč tabulky *Uzivatele* je pro určení, kdo paletu vyrobil.
  - **vahy\_v\_poradku** – Složí k určení, zda došlo k namátkové kontrole vážení výrobku a zda bylo vše v pořádku.
  - **datum\_vyroby** – Určuje datum výroby dané palety výrobku.
  - **poradove\_cislo\_palety** – Určuje, v jakém pořadí daného dne byla paleta vyrobena.
  - **cas\_vyroby** – Určuje přesný čas začátku výroby dané palety.
  - **poznamka\_k\_palette** - Slouží pro zápis poznámky o vyrobené paletě.
- **Pouziti protlaku** – Tato tabulka slouží k získání informací o tom, z jakého protlaku byla paleta výrobků vyrobena.
  - **id\_palety** - Jedná se o primární cizí klíč tabulky *Palety\_vyroba*, definující evidenční číslo palety.
  - **cislo\_kontejneru\_protлак** – Jedná se o primární cizí klíč tabulky *Protlaky*, definující číslo kontejneru, který byl použit.
  - **cas\_pripojeni\_protлак** - Určuje přesný čas, kdy začala výroba z daného protlaku.
  - **spotreba** – Určuje, jaké množství bylo spotřebováno na danou výrobu.
- **Pouziti polotovar** - Tato tabulka slouží k získání informací o tom, z jakého polotovaru byla paleta výrobků vyrobena.
  - **id\_palety** - Jedná se o primární cizí klíč tabulky *Palety\_vyroba*, definující evidenční číslo palety.

- **cislo\_kontejneru\_polotovar** - Jedná se o primární cizí klíč tabulky *Polotovary*, definující číslo kontejneru, který byl použit.
  - **cas\_pripojeni\_polotovaru** - Určuje přesný čas, kdy začala výroba z daného polotovaru.
  - **spotreba** – Určuje, jaké množství bylo spotřebováno na danou výrobu.
- **Palety\_vyroba\_vrchni\_folie** - Tato tabulka slouží k získání informací o tom, která vrchní folie byla pro danou paletu výrobků použita.
    - **id\_palety** - Jedná se o primární cizí klíč tabulky *Palety\_vyroba*, definující evidenční číslo palety.
    - **cislo\_zakazky** - Jedná se o primární cizí klíč tabulky *Vrchni\_folie*, definující výrobní číslo zakázky.
    - **cislo\_role\_al** - Jedná se o primární cizí klíč tabulky *Vrchni\_folie*, definující výrobní číslo role.
- **Palety\_vyroba\_spodni\_folie** - Tato tabulka slouží k získání informací o tom, která spodní folie byla pro danou paletu výrobků použita.
    - **id\_palety** - Jedná se o primární cizí klíč tabulky *Palety\_vyroba*, definující evidenční číslo palety.
    - **cislo\_zakazky\_lot** - Jedná se o primární cizí klíč tabulky *Spodni\_folie*, definující výrobní číslo zakázky.
    - **cislo\_role\_ps** - Jedná se o primární cizí klíč tabulky *Spodni\_folie*, definující výrobní číslo role.
- **Pouziti\_sila\_tvaroh** - Tato tabulka slouží k získání informací o tom, z jakého tvarohového sila byla paleta výrobku vyrobena.
    - **id\_palety** - Jedná se o primární cizí klíč tabulky *Palety\_vyroba*, definující evidenční číslo palety.
    - **id\_sila** - Jedná se o primární cizí klíč tabulky *Sila\_tvaroh*.
    - **cas\_pouziti** – Určuje přesný čas, kdy začala výroba z daného sila.

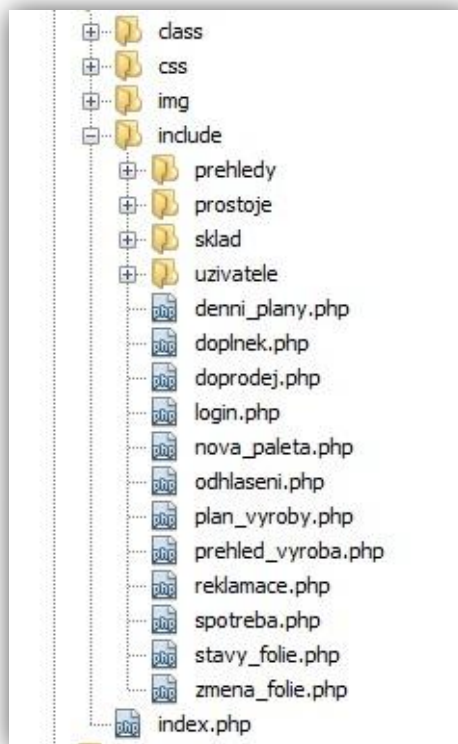
Pro vytvoření tabulek jsem použil funkce programu Toad data modeleru, který umožňuje tvorbu DDL skriptu na základě návrhu relačního modelu dat.

```
CREATE TABLE Palety_vyroba
(
  id_palety Int(7) NOT NULL,
  id_stroje Int(2) NOT NULL,
  id_druh_vyroby Int(3) NOT NULL,
  id_uzivatele Int(4) NOT NULL,
  vahy_v_poradku Bool NOT NULL,
  poznamka_k_palete Text,
  cas_vyroby Datetime,
  datum_vyroby Date NOT NULL,
  poradove_cislo_palety Int(2) NOT NULL
)
;
ALTER TABLE Palety_vyroba
ADD PRIMARY KEY (id_palety)
;
```

**Obrázek 4.7 - Ukázka DDL skriptu pro tvorbu tabulky Palety\_vyroba**

## 4.6 Adresářová struktura webové aplikace

Hlavní stránka *index.php* se nachází v kořenovém adresáři a ostatní stránky jsou uloženy v podadresářích *class*, *css*, *img*, a *include*, kde jsou umístěny i další podadresáře pro větší přehlednost.



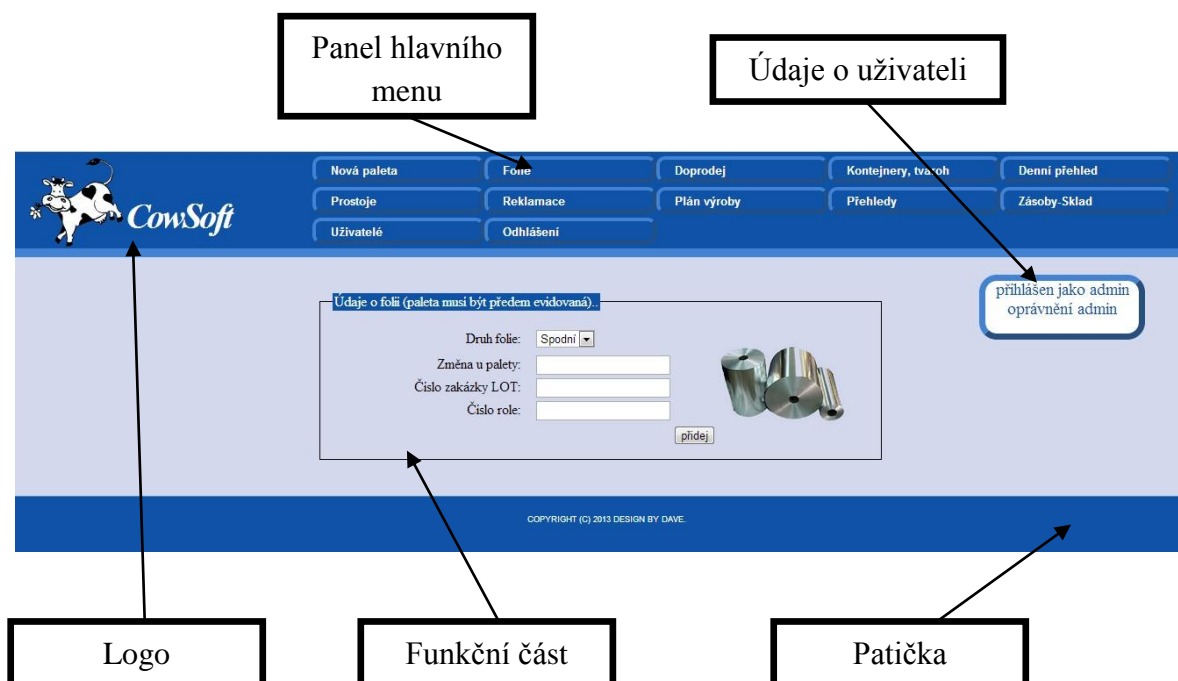
Obrázek 4.8 - Adresářová struktura projektu ve vývojovém prostředí Netbeans

Obsah podadresářů:

- **class** – Zde jsou uloženy třídy obsahující funkce pro připojení do databáze, funkce pro využívání SQL dotazů, atd.
- **css** - Obsahuje soubor kaskádových stylů.
- **img** – Zde jsou uloženy veškeré obrázky, které slouží pro vytvoření grafického prostředí webové aplikace.
- **include** – Obsahuje PHP skripty, které se využívají podle potřeb aplikace.

## 4.7 Webové rozhraní

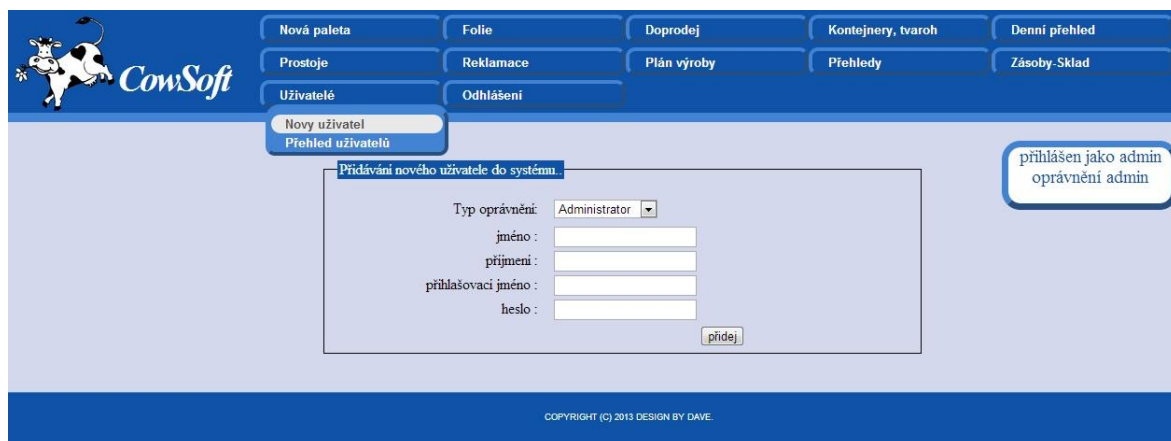
Na Obrázku 4.9 se nachází základní vzhled webového rozhraní administrátorského účtu. Při pohledu na horní část je vidět panel hlavního menu, ve kterém jsou přihlášenému uživateli dostupné funkce na základě jeho oprávnění. Levá část hlavního panelu obsahuje logo aplikace. Pod hlavním panelem v pravé části se nachází údaje o přihlášeném uživateli, jeho uživatelské jméno a přiřazená pracovní skupina neboli oprávnění. Mezi hlavním panelem a patičkou se vždy nachází příslušná funkční část podle výběru v hlavním menu.



Obrázek 4.9 - Vzhled webového rozhraní

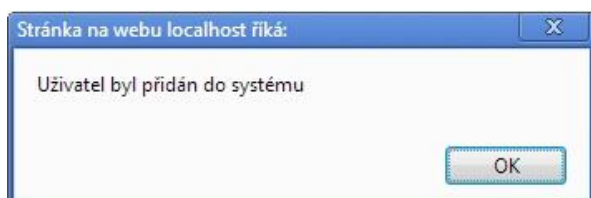
### Ukázka pravomocí účtu Administrátor – přidávání uživatelů

Obrázek 4.10 znázorňuje Administrátorský účet, kde byla z hlavního menu vybrána záložka *Uživatelé* a následně *Nový uživatel*. V této části je umožněno přidávání nových uživatelů do systému. Nejdříve se z menu zvolí uživatelská role pro vytvářeného uživatele, dále jeho jméno, příjmení, přihlašovací jméno a heslo pro ověření identity. Po vyplnění všech potřebných údajů, se pro zavedení uživatele do systému provede kliknutí na tlačítko *přidej*.



**Obrázek 4.10 - Přidávání nových uživatelů**

Pokud uložení nového uživatele do systému proběhlo v pořádku, objeví se vyskakovací tabulka znázorněná na Obrázku 4.12 v opačném případě Obrázek 4.11.



**Obrázek 4.12 - Pozitivní vyskakovací okno**



**Obrázek 4.11 - Negativní vyskakovací okno**

### **Ukázka pravomocí účtu Strojník ve výrobě – přidávání nově vyrobené palety**

Evidování vyrobených palet Obrázek 4.13 se provádí na uživatelském účtu *strojník ve výrobě* po výběru z hlavního menu na záložku *Nová Paleta*. Při evidování vyrobené palety se vybírá balicí stroj, kde se paleta vyrobila, dále druh výrobku, zkontrolování vah, datum výroby (automaticky se do daného pole vyplňuje podle aktuálního dne), pořadové číslo palety a evidenční číslo palety (kód SCSC), začátek času výroby palety a jméno, kdo paletu vyrobil, tento údaj je doplňován automaticky na základě přihlášeného uživatele. Po vyplnění všech potřebných údajů, se pro uložení do systému stiskne tlačítko *přidej*.



**Údaje o vyrobené paletě..**

Balička/středisko: Hassia 2 ▼

Druh výrobku: Tvarohacek vanilka 24 ▼

Vážení: v pořádku ▼


Datum výroby: 08.08.2013

Pořadové číslo palety:

Kód SCSC:

Čas výroby: 08.08.2013 11:47:33

Jméno strojníka - id: strojnik



Obrázek 4.13 - Přidávání nové palety- výřez pohledu z webového rozhraní

### **Ukázka pravomocí účtu vedoucí výroby – plán výroby, přehled plánu**

Pro určení výroby a specifikaci, co se má daný den vyrábět, slouží uživatelská role *Vedoucí výroby*. Tato uživatelská role má v hlavním menu na výběr záložku *Plán výroby*, kde je po jejím rozkliknutí možné vybrat *zadávání plánu* nebo *přehled plánu*.

U zadávání plánu pro výrobu se specifikuje datum plánu výroby, balicí stroj, druh výrobku a počet palet.

Záložka *přehled plánu* zobrazuje přehled plánu podle zadaného dne. V přehledu je pak možné zjistit, jaký druh je potřeba vyrobit, množství palet a kolik bude potřeba celkem surovin na takový objem výroby. Přehled zobrazuje Obrázek 4.14.

**Zobrazení plánu výroby..**

Datum výroby: 08.08.2013

nazev	pocet_palet	tvaroh	protlak	polotovar
Tvarohacek vanilka 24	10	1200	700	400
Tvarohacek kakao 24	8	960	480	400
0	18	2160	1180	800

Obrázek 4.14 - Přehled plánu výroby - výřez pohledu z webového rozhraní

## Ukázka některých vybraných částí PHP kódu

- **Výpis dat pomocí tabulky** - Na Obrázku 4.15 je uveden PHP kód, který byl použit pro uspořádání a výpis dat do tabulky.

```
<?php
if ($navrat) {
    $i = 0;
    echo "<table border=3,5>";
    echo "<tr>";
    foreach ($navrat->fetch_fields() as $meta)
        printf("<th>%s</th>", htmlentities($meta->name));
    echo "</tr>\n";

    // obsah tabulky
    while ($row = $navrat->fetch_row()) {
        $i++;
        if($i%2==1){
            echo "<tr id=\".'suda'.'.>";
        }else
            echo "<tr>";

        foreach ($row as $col)
            printf("<td>%s</td>", htmlentities($col));
        echo "</tr>\n";
    }
    echo "</table>\n";
}
?>
```

Obrázek 4.15 - Výpis dat do tabulky - PHP kód

- **Přihlášení do systému** - Pro přihlašování uživatelů do systému je použita funkce *login*. Nejdřív se uživatel připojí do databáze pomocí defaultního účtu, který má oprávnění pouze pro čtení z tabulky *Uzivatele*. Na základě tohoto připojení se ověří přihlašovací údaje, pokud jsou údaje správné, uloží se informace o uživateli do session. Následně dojde k připojení do databáze a k webovému rozhraní s funkcemi odpovídajícími danému uživatelskému oprávnění. Funkce PHP kódu je na Obrázku 4.16.

```

function login($username, $password) {
    $db = new db();
    // zakladni pripojeni default pro hosta
    $database = $db->connect();
    $result = mysqli_query($database, "SELECT id_uzivatele,id_opravneni,
        opravneni.skupina,login,password,priorita from uzivatele join
        opravneni using(id_opravneni) where login = '$username'");
    if ($result == false) {
        return false;
    } else {
        $row = mysqli_fetch_assoc($result);
        if ($row["password"] == $password) {
            $profil = array('JMENO' => $row["login"],
                'ID_UZIVATELE' => $row["id_uzivatele"], 'PRIORITA' =>
                $row["priorita"], 'SKUPINA' => $row["skupina"]);
            $_SESSION['identity'] = $profil;
            self::$identity = $profil;
            // prihlaseni do DBS s patricnym opravnenim
            $db->connectGroup(self::$identity['SKUPINA']);
            return true;
        }
        else
            return false;
    }
}
}

```

Obrázek 4.16 - Přihlášení do systému - PHP kód

- **Konverze data** – Vzhledem ke skutečnosti, že u databáze MySQL je datum v jiném formátu než je u nás zvykem, jsem pro přehlednost ve webovém rozhraní používal funkci na konverzi. Tato funkce byla vždy použita před uložením dat do databáze. Použitou funkci je možné vidět na Obrázku 4.17.

```

function replaceDatum($datum) {
    $datumConverted = preg_replace('~^([0-9]+)\.\.([0-9]+)\.\.([0-9]+)~',
        '\\3-\\2-\\1', $datum);
    return $datumConverted;
}

```

Obrázek 4.17 - Konverze data - PHP kód

## Závěr

V bakalářské práci jsem se v teoretické části zabýval především historií databázových systémů. Shrnul jsem jejich vývoj od počátku až dodnes. Dále jsem psal o relačním databázovém systému, jeho složení a produktech, jimiž jsou například uvedené MySQL, Oracle Database a PostgreSQL. Zaměřil jsem se na Coddova pravidla, jejichž autor dal základ relačním databázovým systémům. Uvedl jsem jeho 12 pravidel, které jsou tzv. měřítkem, zda je databázový model opravdu relačním.

Dále jsem ve své práci popisoval proces výroby a klíčové údaje pro její evidenci. Uvedl jsem zde uživatelské role v procesu výroby, které jsou zde nezbytnou součástí. Mezi uvedené role patří skladník, strojník ve výrobě, vedoucí výroby, mistr ve výrobě, výrobní ředitel a administrátor systému. Zveřejnil jsem používané tištěné formuláře, které sloužili i jako částečná inspirace pro návrh systému, na jejichž základě se snadno odvodilo, jaká data je nutné evidovat. Tyto formuláře evidovaly používaný materiál a suroviny, které se podílely na výrobě daného výrobku. Zmíněné evidence se týkají vyrobených palet a k nim používaných polotovarů a protlaků. Eviduje se i obalový materiál. Při výrobě se mohou vyskytnout v případě poruch nebo čekání na materiál prostoje. Vznikají i doprodeje, děje se tak chybou nastavení obalové linky nebo vadou materiálu. I tyto vzniklé události se musí evidovat. Do procesu výroby zahrnujeme i skladové zásoby, ze kterých se vyrobí finální výrobek. Mezi skladové zásoby patří vrchní a spodní folie, polotovary a protlaky.

Navrhl a implementoval jsem databázový model, který slouží k evidenci dění ve výrobě. Nejdříve jsem navrhl relační datový model, u kterého jsem vycházel ze zmíněných formulářů a dalších mnou přidáných kritérií pro evidenci výroby, sloužících ke zlepšení systému. Z tohoto modelu jsem dále vygeneroval databázi. K této databázi jsem navrhl vhodné webové rozhraní, zajišťující spolupráci s touto databází. Uživatelé, kteří se přihlásí do tohoto rozhraní, mají funkce přidělené na základě jejich oprávnění.

Po splnění zadání a vytvoření databázového systému mohu říci, že tento systém by se dále mohl vyvíjet na základě požadavků využívající společnosti. Lze do systému přidat několik dalších funkcí, aby se výroba dala ještě více kontrolovat. Bylo by možné přidat funkce, které by se zabývaly například laboratorními výsledky ve výrobě. Osobním přínosem pro mě bylo vytvoření rozsáhlejšího projektu, ve kterém jsem využil znalosti z oblasti databázových systémů a prohloubení mých znalostí v jazyce PHP.

## Literatura

- [1] **LECKY-THOMPSON, Ed a Steven D NOWICKI.** PHP 6: programujeme profesionálně. Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 2010, 718 s. Programujeme profesionálně. ISBN 978-80-251-3127-5.
- [2] **CASTRO, Elizabeth a Steven D NOWICKI.** HTML, XHTML a CSS: názorný průvodce tvorbou WWW stránek. Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 2007, 438 s. Programujeme profesionálně. ISBN 978-80-251-1531-2.
- [3] **LACKO, Ľuboslav a Steven D NOWICKI.** 1001 tipů a triků pro SQL: názorný průvodce tvorbou WWW stránek. Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 2011, 416 s. Programujeme profesionálně. ISBN 978-80-251-3010-0.
- [4] **KOFLER, Michael a Bernd ÖGGL.** PHP 5 a MySQL 5: průvodce webového programátora. Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 2007, 607 s. Programujeme profesionálně. ISBN 978-80-251-1813-9.
- [5] **Žák, David.** Databázové systémy I – Přednáška č. 10. Pardubice: Univerzita Pardubice. 2012. [cit. 2013-07-25].
- [6] **KAPOUN, Jan.** Průkopníci informačního věku (30.) - Edgar Codd. In: *Business World* [online]. Praha: IDG Czech, 2012 [cit. 2013-07-25]. Dostupné z: <http://businessworld.cz/analyzy/prukopnici-informacniho-veku-30-edgar-codd-10089>.
- [7] **Žák, Karel.** Historie relačních databází. *ROOT.CZ*. [Online] 2001. [cit. 2013-07-21]. Dostupné z WWW: <http://www.root.cz/clanky/historie-relacnich-databazi/>.
- [8] **STĚHULE, Pavel.** PostgreSQL. In: *Czech PostgreSQL wiki* [online]. 2013 [cit. 2013-07-28]. Dostupné z: <http://postgres.cz/wiki/PostgreSQL>.
- [9] **Panský, Mikoláš.** Jedenáctý Oracle poprvé. *Databázový svět*. [Online]. 2007. [cit. 2013-07-29]. Dostupné z: <http://www.dbsvet.cz/view.php?cisloclanku=2007091401>.
- [10] **Jonáš, Marek.** Oracle SQL Developer. *LSD SPŠE Ječná*. [Online] 2011. [cit. 2013-08-01.]. Dostupné z: [http://lsd.spsejecna.net/wiki/Oracle\\_SQL\\_Developer](http://lsd.spsejecna.net/wiki/Oracle_SQL_Developer).
- [11] **Kučerová, Helena.** Databázové systémy - sylaby ke kurzu. Praha: Vyšší odborná škola informačních služeb. 2004 [Citace: 25. Červenec 2013.].
- [12] **ŠIMONOVÁ, Stanislava a Jan PANUŠ.** Databázové systémy I: pro kombinovanou formu studia [online]. Vyd. 1. Pardubice: Univerzita Pardubice, 2007, 106 s. [cit. 2013-08-11]. ISBN 978-80-7194-988-6.

