Ostravská univerzita v Ostrave

Fakulta prírodovedecká

Studijní program: Informatika

DIPLOMOVÁ PRÁCE

Internetové služby - implementace a optimalizace.

Radovan KALUŽA

Vedoucí diplomové práce: **Mgr. Jaroslav Knybel**
Rok: 2006

„Prohlašuji, že jsem tuto diplomovou práci zpracoval samostatne a veškerá použitá literatura a další prameny jsou uvedeny v seznamu“.

Dekuji svému vedoucímu diplomové práce panu **: Mgr. Jaroslav Knybelovi** za jeho odborné vedení a podnetné pripomínky pri tvorbe mé diplomové práce.

V Ostrave dne 13. dubna 2006

1. Úvod

Cílem této práce bylo vytvorit kvalitní produkt – tzv. e-shop, který je schopen nabídnout neco nového. Existuje mnoho tzv. opensource produktu (produktu, které jsou poskytovány zdarma), tyto produkty jsou masove využívány, jsou charakteristické tím, že náklady na jejich porízení jsou nulové, avšak již na první pohled jeví známky neprofesionality a nemohou být tudíž ani úspešné.

OPENSOURCE

+ Zdarma

- Uživatel (neznalý programování a optimalizací pro vyhledávace a hw) si vetšinou vše delá sám z presvedcení, že se nejedná o složitý proces. Výsledek zpravidla **vypadá neprofesionálne**
- nedá se zabudovat do vlastního vzhledu

E-SHOP

+ nízká paušální cena (rádove napríklad Kc 2000,--)
+ možnost zabudování do vlastního grafického systému behem nekolika hodin (rádove 2)
+ manuály v ceském jazyce

ukázky mnou vytvorených e-shopu mužete shlédnout na:

http://www.kosmetika-lucie.cz/

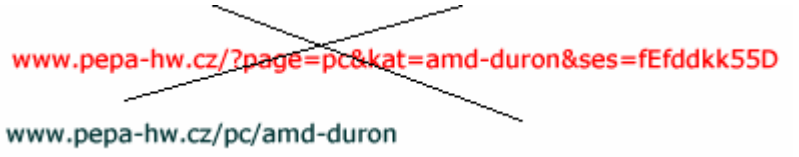
http://www.skolaplus.cz/

E-SHOP NA PRÁNÍ KLIENTA

+ systém delaný tzv. na klíc
- vetšinou nutnost programovat vše od základu
- cena vysoká (rádove 30 – 70 tisíc korun)

novinky (prípadne doposud nezažité veci) v tomto e-shopu:

- obchod bude úspešný ve vyhledávacích – **tím pádem i celkove úspešný**
- obchod má jednoduché URL – možnost verbáního predávání mezi lidmi



- pripraven pro masové použití s možností rychlého individuálního rešení
- veškeré nové funkce a moduly zajištují kompatibilitu s puvodní verzí

PLÁN VÝVOJE

- stanovení funkcnosti e-shopu
- stanovení práv, kdo je oprávnen co delat
- urcení vhodného databázového modelu
- programování – implementace – NEJDELŠÍ CÁST
- systémové optimalizace
- testování a optimalizace pro vyhledávace
- využití marketingových možností

1.1. Základní pilíre systému

JEDNODUCHOST

pro klienta (nakupujícího)

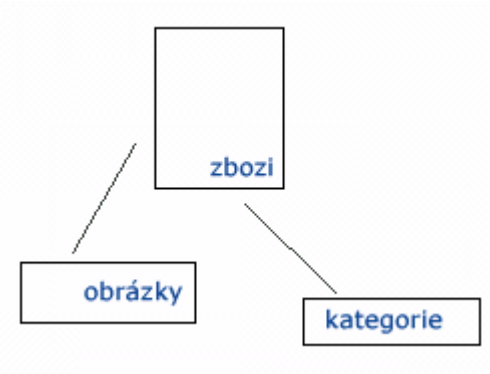
- jednoduché a intuitivní ovládání (nekteré e-shopy neumožnují intuitivní ovládání)

pro správce (prodávající)

- velice rychlá a intuitivní správa e-shopu, vše je ovladatelné z jedné, hlavní stránky
- vzhled pro správce je podobný vzhledu jednoduchého e-shopu

pro implementátora (programátora)

- jednoduchý databázový model, jednoduše popsána funkcnost

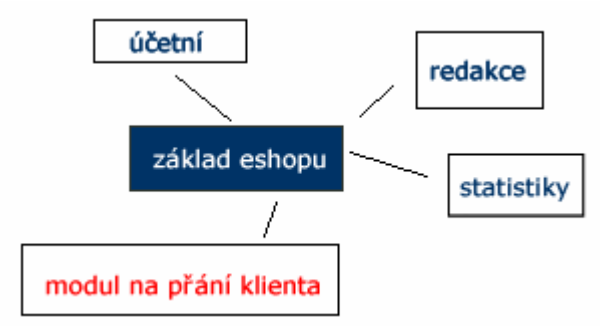


- 3-vrstvý model (data, aplikace, klient)
- jednoduchá konfigurace – implementace

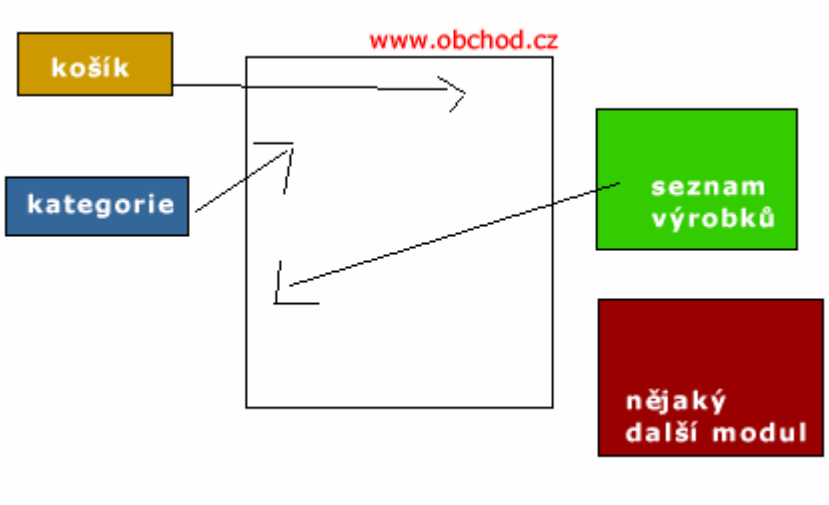
```
#    1=obrazek vlevo
#    3=obrazek nahore, ikonky dole v

$sirka_pro_vypis_vyrobku   = 370;
$sirka_pro_text_vyrobku    = 250;

$zobrazovat_ikonku_detail  = true;

$format_nahledu_obrazku      = "na
#$format_nahledu_obrazku      = "
```

PRUŽNOST

- je zde možnost pripojení **prídavných modulu** k jakékoliv cásti e-shopu (fakturacní modul, modul clánky..) a to kdykoliv v budoucnosti

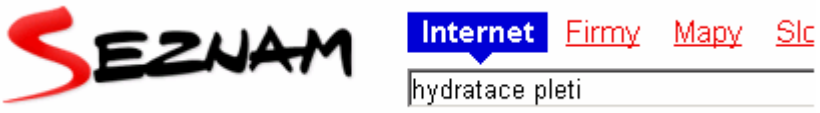


- rychlé a pružné nastavení funkcí a modulu
- sestavení e-shopu funguje **na principu - PUZZLE**



ÚSPEŠNOST

- na stránky e-shopu by se meli dostat lidé, kterí dané zboží/problém hledají (SEO),
  (vetšina podobných projektu na internetu, nemá žádnou optimalizaci pro vyhledávace)

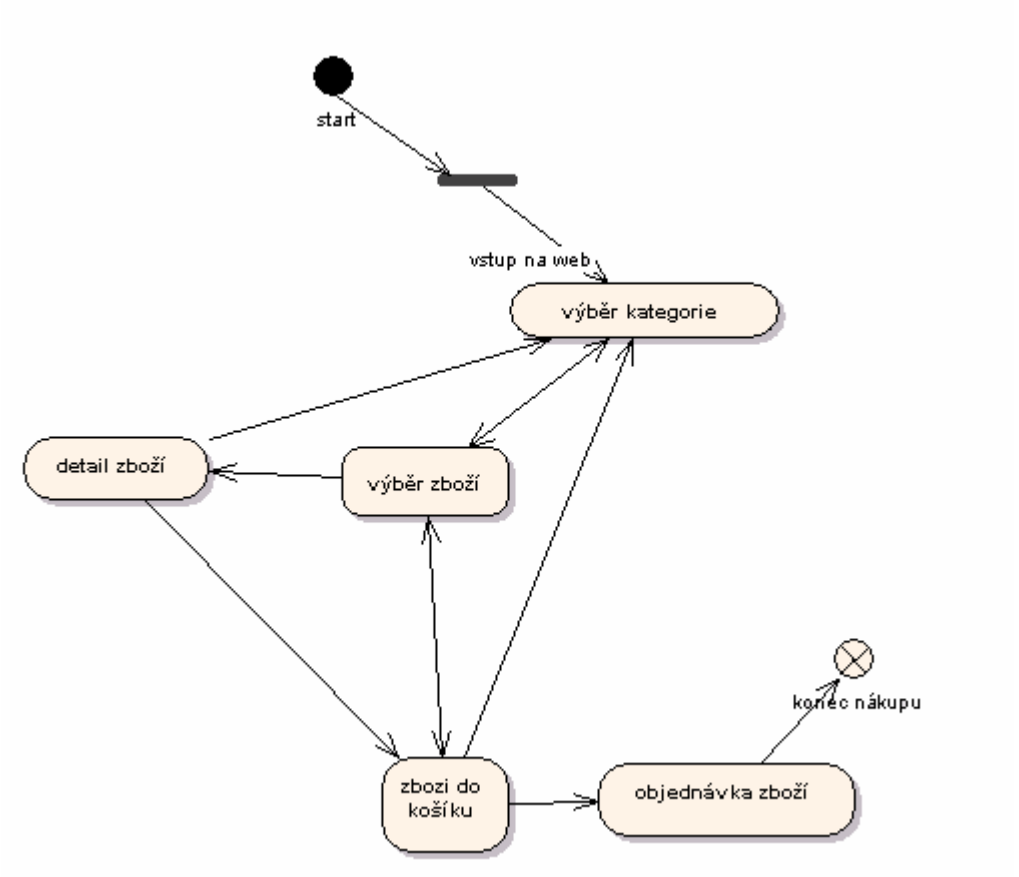


2. Implementace internetové služby - elektronického obchodu

2.1. Aktivity diagram

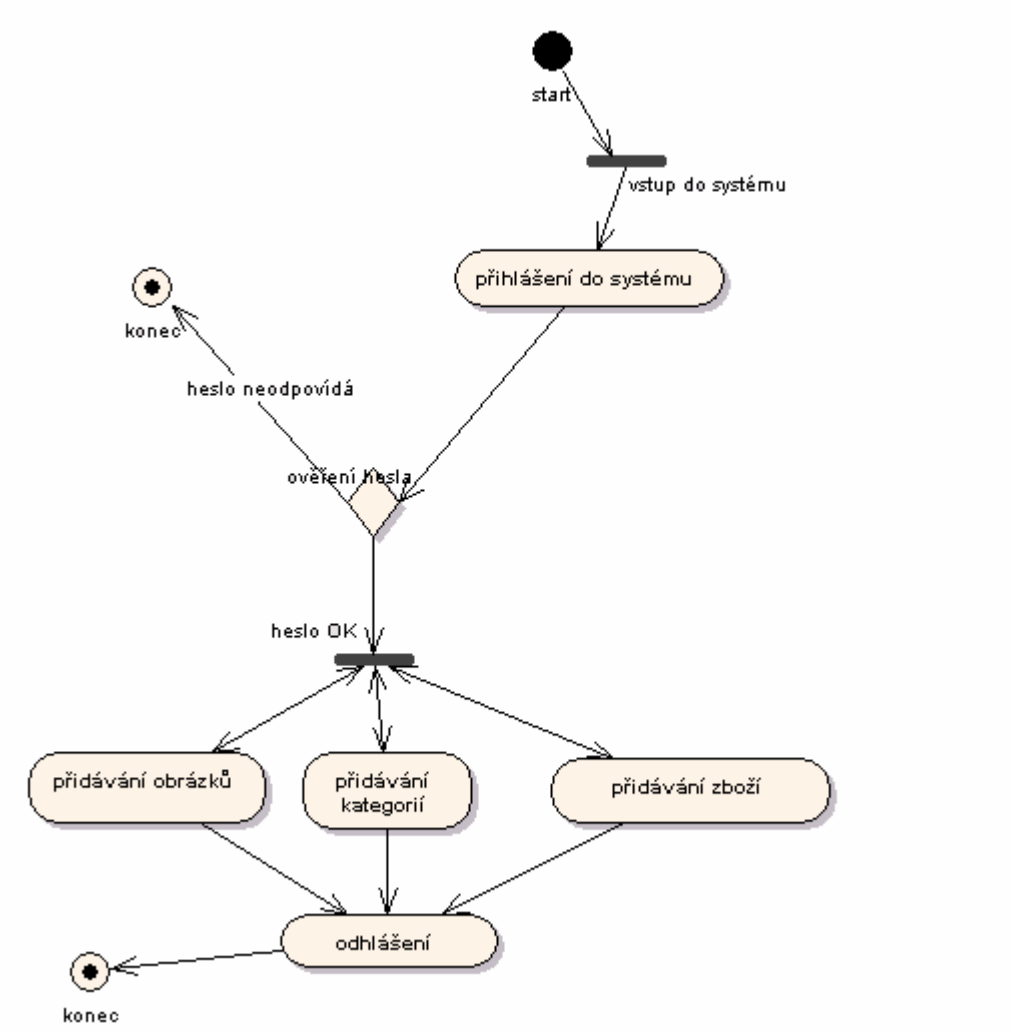
2.1.1. Nákup v e-shopu

Klient vybere príslušnou kategorii, ve které se výrobek nachází, poté se mu zobrazí seznam výrobku. Klient má možnost vybraný výrobek vložit do košíku, nebo si nechat zobrazit detailní informace o daném výrobku. V detailním popisu výrobku má klient rovnež možnost výrobek vložit do košíku. Tento proces muže opakovat tak dlouho, dokud se nerozhodne, že uskutecní objednávku.



2.1.2. Administrace e-shopu

Správce e-shopu (prodavac/-ka) je povinnen se pro administraci v systému prihlásit. Prihlášením vstoupí správce do systému. Zde má možnost pridávat do systému obrázky, ruzné kategorie nebo zboží. Tyto cinnosti je možno vykonávat tak dlouho, dokud se správce ze systému neodhlásí.

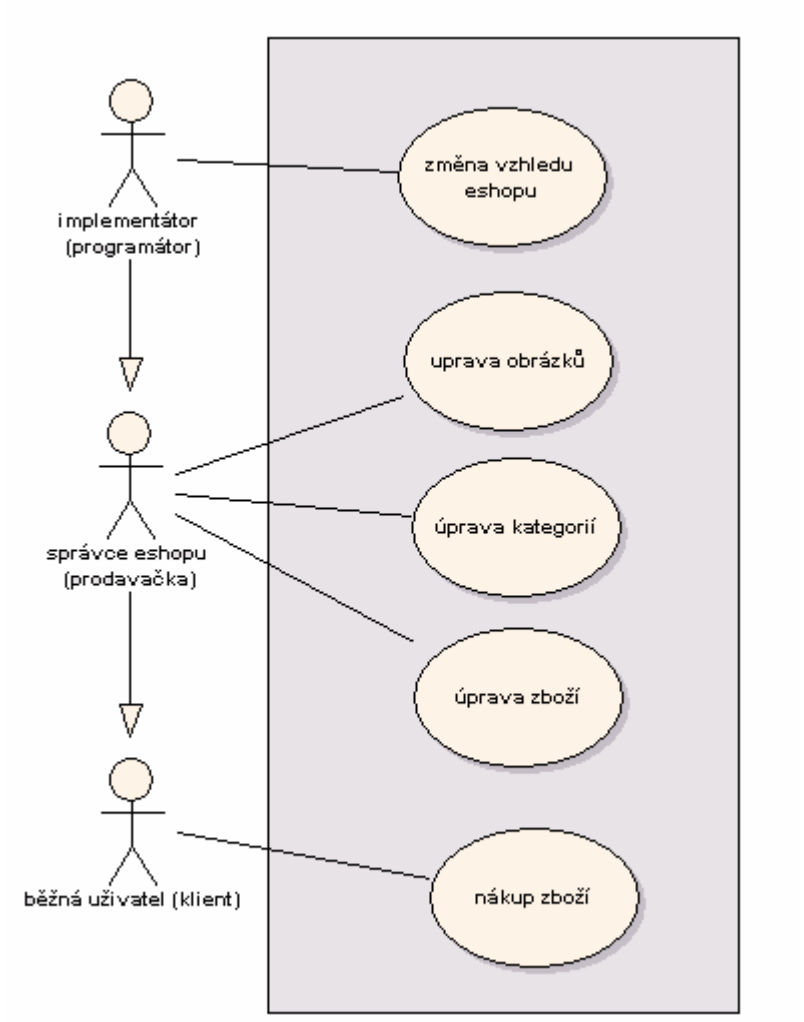


2.2. Use case

V celém systému figurují 3 typy „postav“:

- **klient** (nakupující)
- **správce** (prodavac, skladník)
- **implementátor** (programátor)

Nejmenší práva má klient, který má pouze možnost nákupu zboží.
Nad ním je správce e-shopu (prodavac/-ka), který má všechna práva bežného klienta a navíc má i možnost vstoupit do administrace.
Nejvetší práva má implementátor, který má možnost menit vzhled systému pomocí konfiguracních souboru.



2.3. Modelování databáze

2.3.1. Scénár

Eshop je založen na nutnosti pouze základních prvku. A to jsou **výrobek**, **kategorie** výrobku a **galerie obrázku**.

Kategorie a galerie obrázku, jsou pro všechny klienty navrženy stejne. Pouze výrobek je mírne komplikovaný, výrobek muže obsahovat ruzné parametry, napríklad nekdo potrebuje uvést výrobce, nekdo modelovou radu, jiný e-shop muže obsahovat pouze popis.

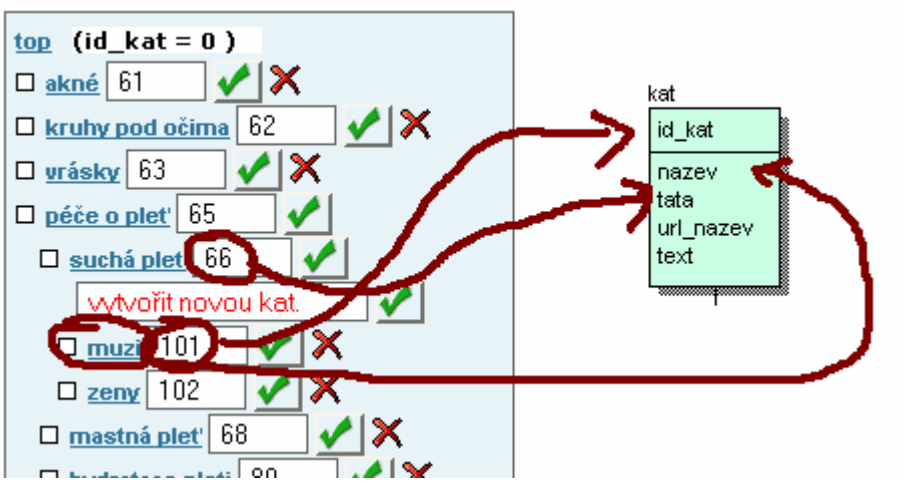
Rešení by mohlo být takové, že sloupce u **výrobku** budou mít spoustu bezejmenných parametru a mnoho ruzných datových typu, kterým by klient pridelil jednotlive jméno a typ, ale jelikož každá implementace e-shopu bude individuální (nikoliv masová), bude tabulku **zboží** vytváret implementátor dle požadavku klienta.

2.3.2. Nezávislé entitity

2.3.2.1. Kategorie

NÁZEV TABULKY : KAT

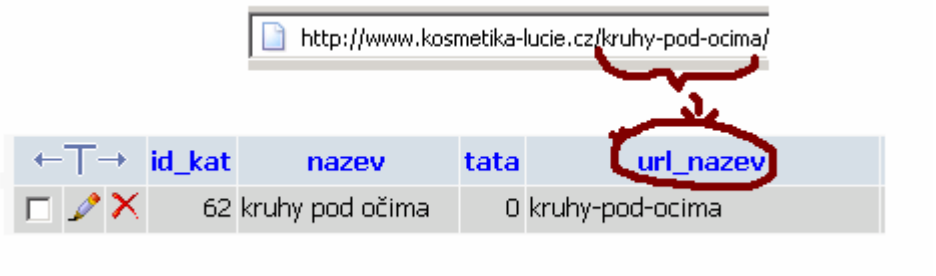
Nekteré internetové obchody, mají rešení kategorií statické. A to takovým zpusobem, že je možné mít kategorie jen do n-té úrovne. V našem prípade jsem kategorii pojal z pohledu rekurze, pricemž hlavní kategorií je kategorie císlo 0, která má své „potomky“, a tito mohou mít rovnež své další „potomky“.



id_kat – je jednoznacne urcené identifikacní císlo kategorie

tata – prevzato ze sveta lidí, má každý syn svého otce, jelikož v našem prípade musí být nejaký hlavní otec (otec všech kategorií), je to top kategorie s id_kat = 0. Jednoduše znázorneno na obrázku:

url-název - je text, který uvidíme v prohlížeci. Viz obrázek, pokud tento parametr je prázdný.



text – parametr s výchozí hodnotou nastavenou na NULL, není tedy nutné využívat tento parametr pro správný chod systému, je to doplnující popis k urcité kategorii. Napríklad odkazy na nejaké recenze k dané kategorií produktu, speciální podmínky atd.

2.3.2.2. Výrobek

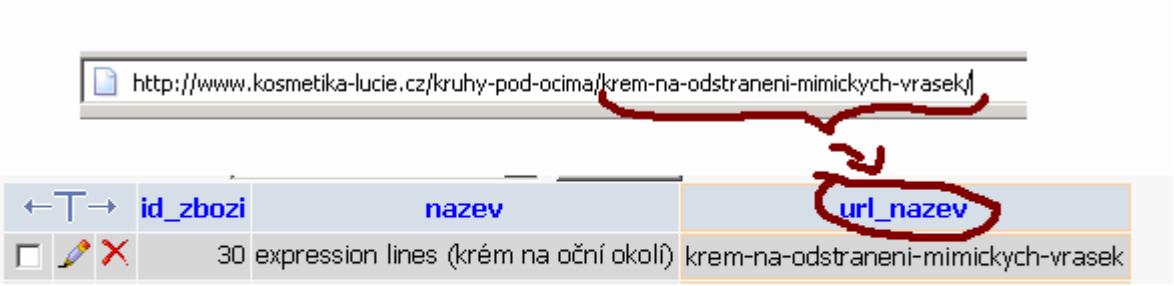
NÁZEV TABULKY : ZBOZI

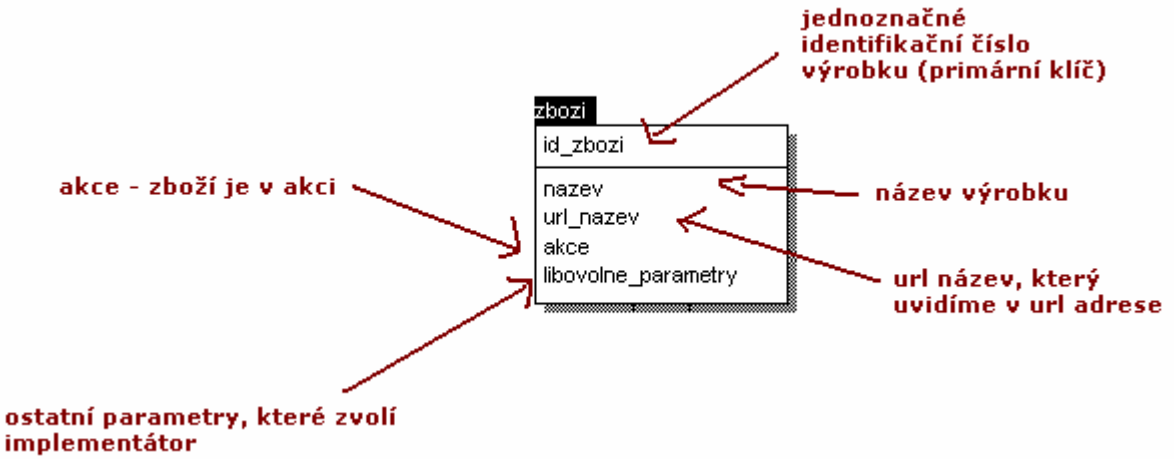
Entita zbozi je klícovým faktorem, cílem je prizpusobit entitu tak, aby bylo možno e-shop využít v ruzných oblastech. Proto témer veškeré její parametry jsou volitelné.

Pro všechny instance e-shopu jsem tyto parametry nastavil jako povinné (není však nutné všechny parametry využívat).

	primární klíc	**Popis**	**implicitní hodnota**
id_zbozi	ano	jednoznacné císlo výrobku	automaticky zvýšené ID o jedna
nazev		jméno výrobku	N/A
akce		výrobek je v akci	NO (není v akci)
url_nazev		parametr, který uvidíme v URL	NULL (jako url název potom bereme

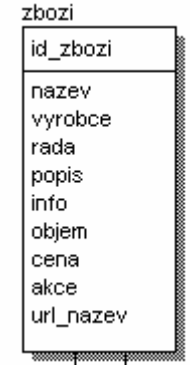
Ukázka parametru url_nazev v praxi, vidíme zde adresový rádek browseru s url adresou stránky.





Obecný príklad. A takto si muže implementátor upravit entitu zboží sám.

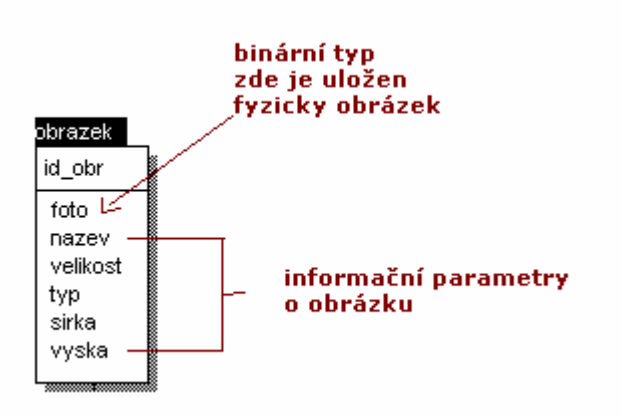
Naše konkrétní implementace, pro vzorový e-shop s kosmetikou.



2.3.2.3. Obrázky

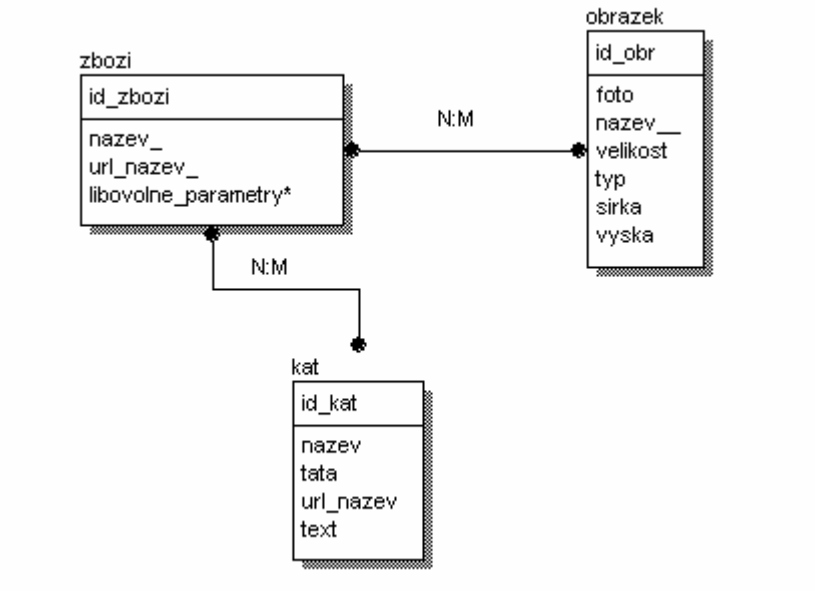
název tabulky **: obrazek**

Tabulka, ve které budou uloženy jednotlivé obrázky.Obrázky jsou rovnež nezávislá entita, jako zboží a kat. Pro webové prezentace by bylo výhodnejší nacítání souboru prímo z disku, nikoliv z databáze, velikost obrázku bude limitována a bude možné nad ním vytváret sql operace a transformace.



2.3.2.4. Vztahy nezávislých entit

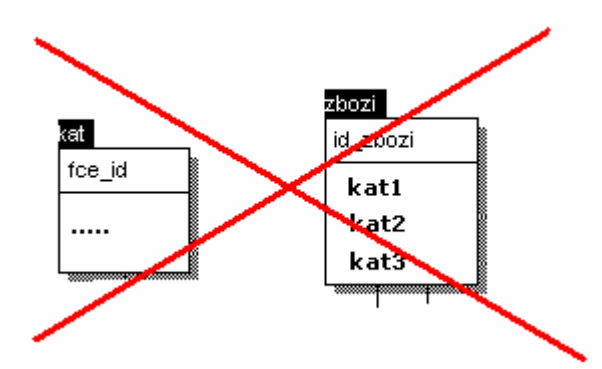
Každé zboží muže být zarazeno ve více kategoriích (napríklad PC komponent-pocítacová pamet muže být jak soucástí kategorie pameti, tak soucástí kategorie volitelné doplnky k PC sestavám). Každé zboží muže rovnež mít k jednomu popisu více obrázku. Každý jednotlivý obrázek muže být prirazen u jednoho nebo více výrobku (príkladem ilustracní foto pocítacové pameti, mužeme použít u všech typu pameti). Každá kategorie muže rovnež obsahovat jeden nebo více druhu zboží.



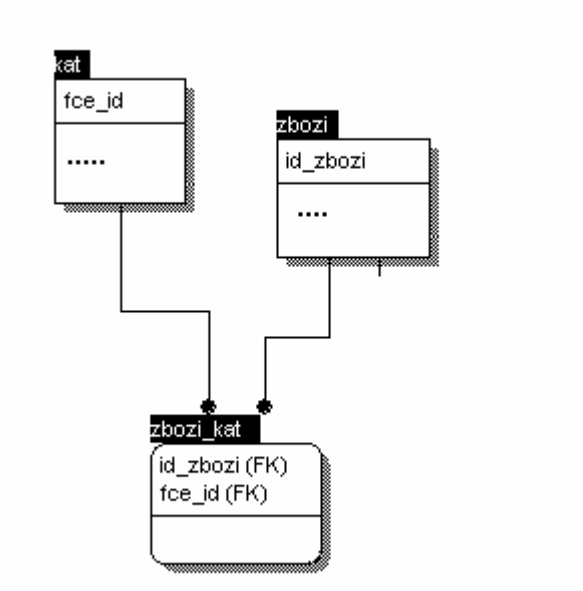
2.3.3. Závislé entity

2.3.3.1. Prirazení zboží do kategorie

Každý výrobek je vetšinou prirazen do jedné, maximálne dvou nebo trí kategorií. Abychom ušetrili jednu entitu, mohli bychom do tabulky zbozi pridat 3 parametry s názvy kat1,kat2,kat3, toto by pravdepodobne vyrešilo realitu. Jsou mi známy projekty na internetu,které touto „neduhou“ trpí.

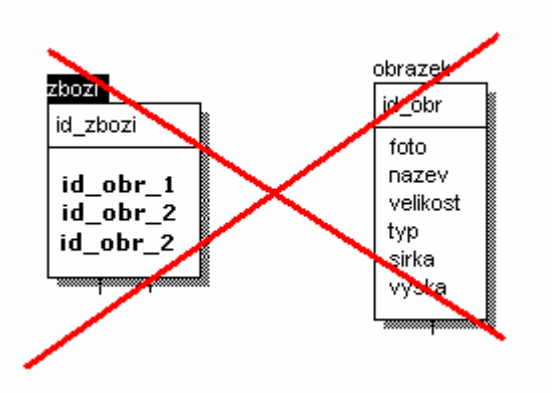


Nerad pripojuji i tabulku, která sloužila jako císelník kategorie a zbozi. Za cenu mírne složitejších sql dotazu, máme však jistotu funkcnosti i pri atypických situacích - když bude jeden druh zboží z více kategorií.

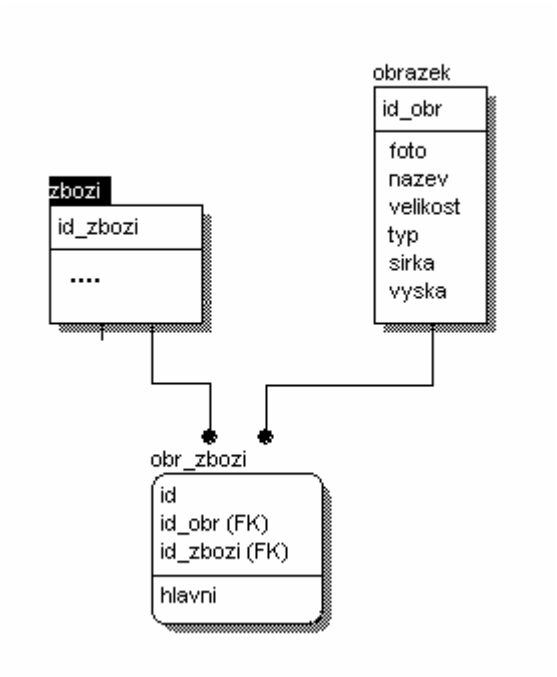


2.3.3.2. Prirazení obrázku ke zboží

Analogicky k predchozí úvaze, jak priradit zboží ke kategorii, by se dalo uvažovat i v prípade, jak priradit obrázek ke zboží. V praxi pravdepodobne nebude zapotrebí více jak trí obrázku k jednomu výrobku, vetšinou to vyreší jeden obrázek.

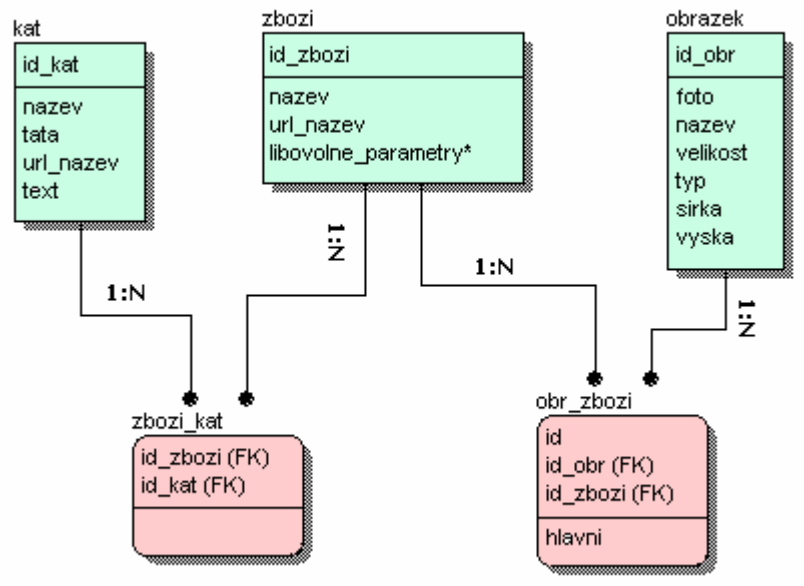


Avšak pro univerzálnost a správnost z hlediska normalizacních forem je nutné zvolit tretí tabulku, která bude sloužit jako císelník zboží a obrázku.



2.3.4. Definice závislých entit , kompletní model

Výsledný model vycházející z predchozích dvou bodu bude obsahovat 5 tabulek (entit).



2.3.5. Splnitelnost 0NF, 1NF, 2NF, 3NF, Boyce - Coddova normální forma, 4NF, 5NF

0NF (nultá normální forma):

Tabulka je v nulté normální forme, existuje-li alespon jedno pole, které obsahuje více než jednu hodnotu.

1.normální forma

- všechny atributy jsou atomické, nemáme však primární klíc.

Príkladem nesplnitelnosti 1NF muže být tabulka obrázek, ve kterém budeme ukládat atributy šírka a výška v jednom sloupci.

Atributy nejsou atomické (atributy šírka i výška by mely být uvedeny oddelene)

Náš model splnuje 1NF, vždy, nebot všechny atributy jsou atomické.

2.normální forma

máme primární klíc, avšak existuje atribut, který není primárním klícem, a je tranzitivne závislý na primárním klíci.

Príkladem tranzitivity muže být následující tabulka, obsahující mesto, stat a ulici.

Pokud budeme predpokládat, že název mesta je unikátní, potom :

Id -> MESTO

Mesto -> Stat

(Id -> MESTO) & (Mesto -> Stat) ├ Id -> Stat (tranzitivita)

Náš model splnuje 2NF, vždy máme definovaný primární klíc.

3.normální forma

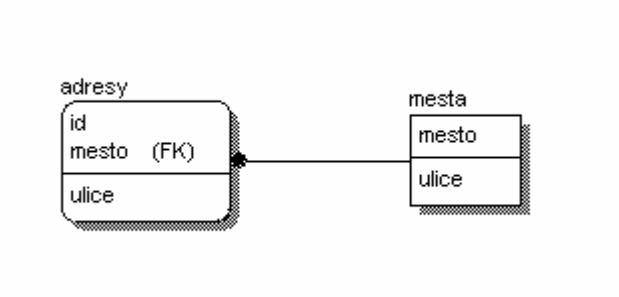
Splnuje 2 NF a neexistuje atribut, který není primárním klícem, a není tranzitivne závislý na žádném klíci.

Príklad 2.normální formy a …

id	mesto	stat	ulice
1	praha	cz	Novakova
2	ostrava	cz	Ostravská
3	New york	usa	Main st.
4	brno	cz	Na polickách

… a prevod na 3NF

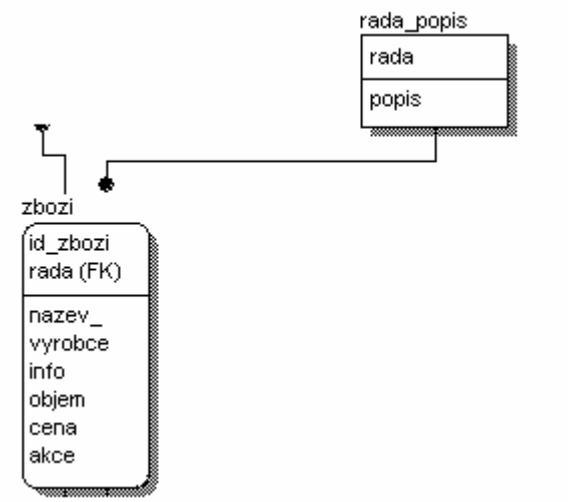
Odstranení tranzitivity pridáním další tabulky, obsahující atributy MESTO a STAT. A odstranení stávajícího atributu stat. Výsledkem bude relace:



Náš model splnuje i 3NF. Mohlo by se zdát, že splnen není, nebot název výrobku a výrobce bychom mohli rozdelit do další tabulky, avšak vycházím z toho, že jeden název výrobku produkují dve spolecnosti, stejným zpusobem mohou jednu radu zboží produkovat ruzné spolecnosti. Dalším duležitým faktem je, že implementátor bude pridávat parametry pouze do tabulky ZBOZI, nikoliv vytváret nové tabulky. Je to z duvodu zachování funkcnosti, aplikace je tedy pripravena na jednu tabulku s ruznými parametry (ty si implementátor nakonfiguruje v konfiguracním souboru). Pokud bychom chteli 3NF zachovat pri jakékoliv implementaci e-shopu i u tabulky ZBOZI, konfigurace by byla mnohem nárocnejší.
Ostatní tabulky nejsou tímto ohroženy. Avšak náš konkrétní prípad platí v 3NF pro všechny tabulky.

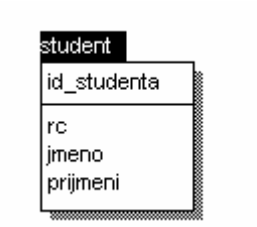
Príkladem

Tabulka zbozi nebude pri uvažované realite v 3NF, kdy každou RADU popisujeme stejným INFO. Rešením by byla další tabulka. S touto možností však nepocítáme, nebot info bude vždy unikátní.



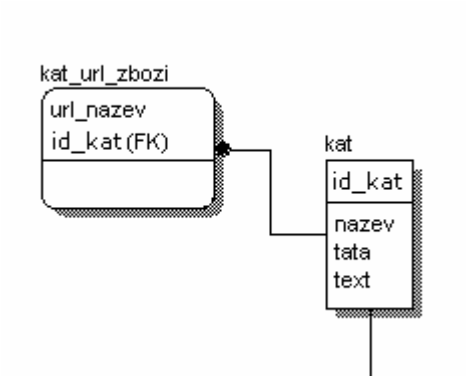
Boyce - Coddova normální forma

tato forma reší nedostatky vzájemných závislostí klícových atributu. Necht máme tabulku Student, primárním klícem je id_studenta, tedy i samotné rc muže být primárním klícem. Je možno jeden z atributu vypustit, pokud je však zapotrebí obou atributu , potom je nutné vytvorit další tabulku se sloupci id_studenta a rc. Príklad, kdy není splnena BCNF:



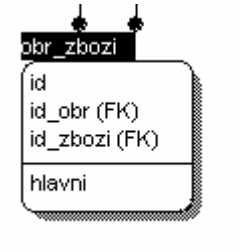
Id_studenta	rc	Prijmeni
1	801101/1234	Borovský
2	811205/2224	novák
3	800702/2323	dobrácek

Tabulky kat a zbozi nejsou v BCNF, nebot **ID_ZBOZI** je primární klíc a rovnež **URL_NAZEV** musí být názvem unikátním, avšak url_nazev není povinný parametr. Další tabulka s id_zbozi (resp id_kat) a url_nazev by mela za následek výraznou neprehlednost pri správe tabulek. Správne by tedy bylo…..



Ostatní tabulky splnují BCNF.

Mohlo by se zdát, že tabulka obr_zbozi není v BCNF, pocítám však se situací kdy id_obr a id_zbozi bude v tabulce dvakrát, jednou s nastaveným hlavni ANO, podruhé s nastaveným hlavni NE.



4 NORMÁLNÍ FORMA

Tabulka je ve ctvrté normální forme, je-li ve tretí normální forme a popisuje-li pouze prícinnou souvislost (jeden fakt). Príklad:

zamestnanec	schopnosti	jazyky
Novák	Kuchar	
Novák	Ridic	
Novák		Anglictina
Novák		Nemcina

Tabulka není v 4.NF, nebot nepopisuje pouze prícinnou souvislost.
Aby byla tabulka v 4NF, mela by vypadat následovne:

Zamestnanec	schopnosti
Novák	Kuchar
Novák	Ridic

zamestnanec	jazyky
Novák	Anglictina
Novák	Nemcina

Tabulky jsou v 4NF, krome tabulky zbozi, ve které muže nastat problém s 3NF, nebo jiný konkrétní návrh tabulky bude podobný výše uvedeným. Náš konkrétní návrh tabulky zbozi je v 4NF.

5NF (pátá normální forma)

Tabulka je v páté normální forme, pokud je ve ctvrté normální forme a není možné do ní pridat nový rádek (rádky) tak, aby se vlivem skrytých závislostí rozpadla na nekolik dílcích tabulek.

zamestnanec	funkce	plat
Novák	Kuchar	15000
Josefík	Malír	12000
Novotná	Asistentka	17000

Pridáním posledního rádku

zamestnanec	funkce	plat
Novák	Kuchar	15000
Josefík	Malír	12000
Novotná	Asistentka	17000
Kokocka	Malír	12000

se nám tabulka rozpadne na dílcí casti. **Bereme v úvahu, že každé povolání je honorováno stejnou cástkou** (napríklad v rámci jedné spolecnosti).

zamestnanec	funkce
Novák	Kuchar
Josefík	Malír
Novotná	Asistentka
Kokocka	Malír

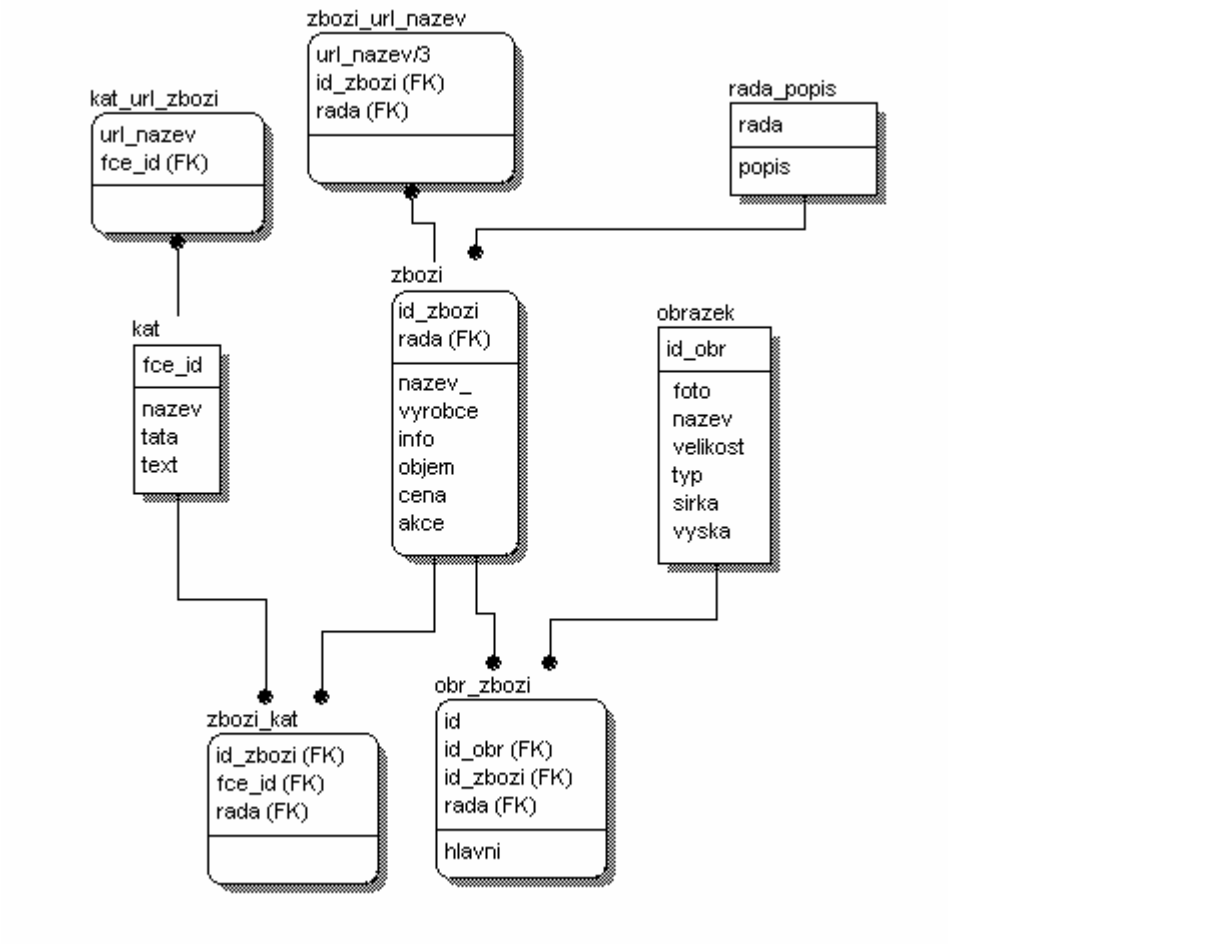
funkce	plat
Kuchar	15000
Malír	12000
Asistentka	17000

Tyto dve tabulky, jsou již v páté normální forme.

V našem konkrétním prípade, jsou všechny tabulky v 5NF, krome tabulky zbozi, ve které muže nastat problém s 3NF, nebo jiný konkrétní návrh tabulky bude podobný výše uvedeným.
Náš konkrétní návrh tabulky zbozí je ovšem v 5NF.

DOPORUCENÍ

V praxi bychom se meli snažit dodržovat alespon tretí normální formu. Nejlepší je vytváret v co nejvyšší normální forme. Nekdy je obtížné dodržet i vyšší formy, tak jak je tomu v tomto prípade, kdy se správa a programování aplikace stává výrazne složitejší. Pro ukázku, jak by vypadal návrh databáze v 5NF (pokud bychom realitu rozebrali podrobneji, zjistíme, že i tento návrh by se musel rozložit do více entit).



Jak je možno videt, ztrácíme jistou možnost srozumitelnosti a univerzálnosti e-shopu, kdy univerzálností rozumíme možnost vytvorit si vlastní tabulku zbozi, možnost používat ci nepoužívat promennou url_zbozi a možnost úpravy nekterých údaju i méne zkušeným implementátorem v nekterém z prostredí pro správu databází. (V našem prípade v PhpMyAdmin)

	0 NF	1 NF	2 NF	3 NF	4 NF	5 NF	BCNF
kat	ano	ano	ano	ano	ano	ano	ne **
zbozi	ano	ano	ano	ano	ano	ano	ne **
obrazek	ano	ano	ano	ano	ano	ano	ano
zbozi_kat	ano	ano	ano	ano	ano	ano	ano
obr_zbozi	ano	ano	ano	ano	ano	ano	ano
zbozi *	ano	ano	ano	ne	ne	ne	ne

*upravené implementátorem, a nejhorší prípad, který muže nastat

** pokud použijeme u všech rádku URL_NAZEV, potom jsou zde dva primární klíce

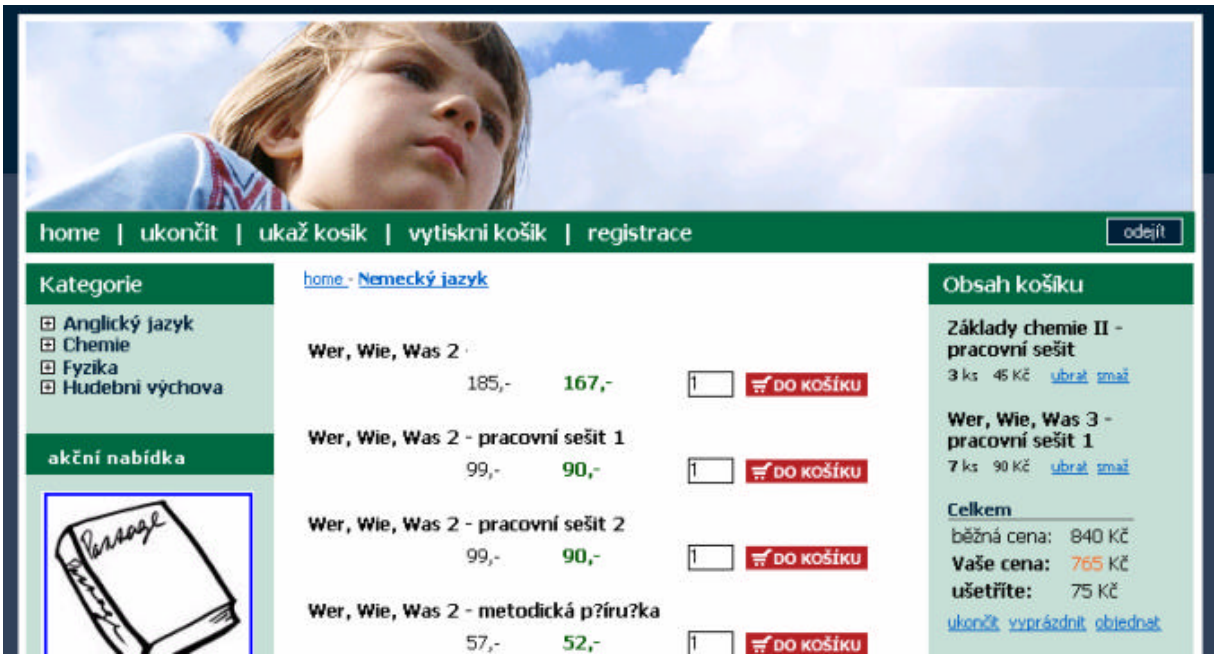
2.4. Puzzle efekt pri implementaci nové instance e-shopu (vkládání jednotlivých modulu a funkcností)

2.4.1. Proc puzzle efekt - malý príklad

Každý e-shop je zcela odlišný, avšak vetšina e-shopu potrebuje pro fungování pouze nekteré části (funkcnosti) a nepožaduje jiné. Príkladem mohou být dva internetové obchody, využívající tento produkt.

E-shop s prodejem knih

Internetová prodejna implementována v praxi, jednalo by se o internetový obchod, který by byl jakýmsi velkoskladem knih. Ocekává se, že klient bude ve znacné míre používat nákupní košík, proto je nutné jej videt v pravém panelu, rovnež se predpokládá, že majitel bude své klienty lákat na slevy a akcní nabídky – viz. panel vlevo.



Internetový obchod s kosmetikou

Cílem internetového obchodu s kosmetikou bylo poskytnout nejen výrobky, ale i informace. Nebylo tedy zapotrebí mít modul s náhledem košíku ani modul slev výrobkových akcí, ale bylo nutno aktivovat zcela nezávislý modul, umožnující publikování clánku.

2.4.2. Technologická implementace puzzle efektu

Hlavní myšlenkou realizace puzzle efektu a snadné a prehledné práce pri vytvárení aplikace bylo maximální zprehlednení a strukturalizace kódu. Pred vytvorení jakékoliv funkce se nactou všechny dostupné funkce a promenné (jako císlo katagorie, id výrobku, strana pri výpisu kategorie atd.).

Veškeré procesy e-shopu, které využívají uživatelé (tedy hlavní prezentace) se odehrávají v jednom souboru - **index.php.** Výjimkou jsou malé jednoduché procesy, napríklad generování náhledu obrázku, odeslání emailu k objednávce. Tyto skripty mohou být a jsou oddelené, nebot nemají zapotrebí využívat grafické prezentace e-shopu a zároven se jejich ukoncením redirectuje (vrací) zpet na stránku.

Ukázka souboru - zacátku souboru **index.php.**

```
<?
include("inc/system/nacteni_funkci_promenych.php");
# promenne, title, css verze, session start, prihlaseni, funkce do potrebujeme
?>


<body leftmargin=0 topmargin=0>
<div class="web">
```

Ukázka nactení promenných a funkcí . Na zacátku se nacítají funkce, poté se tvorí pripojení k databázi. Pri nacítání promenných se nacítá nejdríve košík se zbožím, který máme v SESSION, poté nacítáme parametry z url, které získáváme díky modu rewrite.

Ukázka souboru **nacteni_funkci_promennych.php:**

```
require("fce/fce_kat.php");            # posloupnost kategorii, generovani reku
require("fce/fce_zbozi_vypis_a_katalog.php");

require("inc/system/mysql_pripoj.php");


session_eshop(); # nastartuje pouzivani session, vytvori nove, pokud nejsou

if (!$pouzivat_css_pro_vsechny_browsery) $css = nacti_css_verzi($_SERVER['HTTP_
                                    else $css = $pouzivat_css_pro_vsechny_brows

##############################################################################
##############################################################################
# nacteni promennych
##############################################################################
$kosik  =  $_SESSION['kosik'];      # obsah kosiku
$par1 = $_REQUEST['p1'];
$par2 = $_REQUEST['p2'];
$par3 = $_REQUEST['p3'];

#print "par1 = $par1<br>";
#print "<pre>";print_r($kosik);

# [cislo vyrobku] => pocet kusu
```

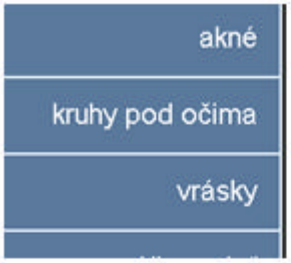
2.4.3. Popis jednotlivých částí puzzle efektu

Tato část je klícovou pro implementátory , kterí budou tento produkt implementovat do urcité grafické predstavy svého klienta.

název	Skript	Povinné	popis
seznam kategorií	seznam_kategorii.php	Ano	
náhled košíku	kosik_seznam_polozek.php	Ne	vykreslí náhled košíku, ze kterého lze ubírat nebo mazat zboží
celková suma košíku	kosik_celkem.php	Ano	
horní menu	horni_menu.php	Ano	
vypis_zbozi.php	vypis_zbozi.php	Ano	vypíše seznam zboží dané kategorie, stránku dle nastavení v konfiguracním souboru
seznam clánku	clanky_seznam.php	ne	prídavný modul, vypíše seznam všech clánku
úvod clánku	clanek_uvod.php	ne	první cást clánku, úvod
text clánku	clanek_text.php	ne	kompletní text clánku, bez úvodu
merící kód	merici_kod.php	ne	mužeme zde vložit ruzné merící kódy, nebo jiné skripty umožnující napríklad logování
text kategorie	text_kategorie.php	ne	nejaký popis dané kategorie
posloupnost k obsahu	posloupnost_k_obsahu.php	ne	vypíše posloupnost ke hlavní kagetorii

* nepovinné dají se považovat za urcitý prídavný modul, který není nutný pro nejzákladnejší chod aplikace e-shopu

seznam_kategorii.php

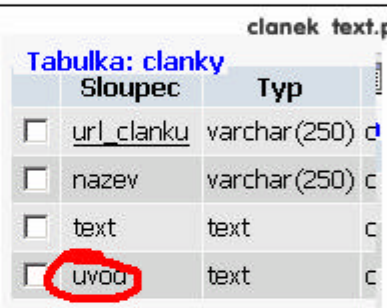


clanky_seznam.php

clanek_uvod.php

kosik_seznam_polozek.php

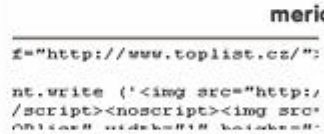
clanek_text.php



kosik_celkem.php

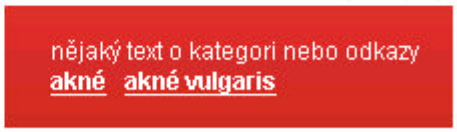
horni_menu.php

merici_kod.php



vypis_zbozi.php

text_kategorie.php



posloupnost_k_obsahu.php

2.5. Nastavení, parametrizovatelnost jednotlivých modulu a funkcností

2.5.1. Nastavení databáze

soubor: **nastaveni.php**

Databázi nastavíme v souboru:

Nastaveni.php (umísteném v adresári **conf**)

```
$SQL_Server   = "localhost";  (nebo IP adresa serveru, kde je db)
$SQL_Uzivatel = "uzivatel";
$SQL_Heslo    = "tajne_heslo";
$Databaze     = "jméno databaze“;
```

2.5.2. Parametry e-shopu

soubor: **parametry_eshopu.php**

Nastavení parametru jednotlivých funkcností a modulu upravujeme v souboru:

Parametry_eshopu.php (umísteném v adresári **conf**)

Soubor je velice dobre popsán, jména promenných jsou intuitivní a casto obsahují i „zakomentované“ vzorové príklady.

Použití je následující:
1/ znakem # zacínají komentáre, tzn., že daný rádek bude ignorován, komentáre obsahují popisy, nebo parametry, které nechceme použít

2/ u parametru, které mají jen logickou hodnotu (ano,ne) píšeme true (ano chceme) nebo false (nechceme). Napr. `$chceme_uvodni_stranu = true;` nám ríká, že na úvodní stránce bude úvodní text, který, jak se dozvíme v popisu prídavných modulu, bude uložen v tabulce texty.

3/ ostatní parametry zmeníme tak, že prepíšeme stávající, u textových je nutné ponechat uvozovky

```
<?
# v tabulce TEXTY editujeme text obsahující název home-horni a home-dolni nebo
pouze home
$chceme_uvodni_stranu  = true;

# nechat nastaveno na true, parametr, který umožní vstup do eshopu jen pro
registrované
# prozatím není implementovano
$neni_nutna_registrace = true;

# cena postovneho a jak tuto polozku uvidi ve fakture
$postovne = 75;
$postovne_info = "poštovné a balné";

# heslo do administrace
$admin_heslo = "xxx";


# titulek, který pujde videt na hlavni strane, nebo na stranach, ktere neobsahuji zadny
$implicitni_titulek = "kosmetika, kosmetická poradna, eshop";

# pokud chceme pouzivat ruzne styly pro ruzne browsery, nechame hodnotu false
# ale potom musi existovat ie.css, mozila.css, opera.css
$pouzivat_css_pro_vsechny_browsery = "mozilla.css";


# tabulka CLANKY obsahuje nazvy clanku
# jeden URL_CLANKU vsak muze nést název pro SEZNAM VSECH CLANKU
# zde se vypise seznam clanku a pouze polozka TEXT
$nazev_rozcestniku_clanku = "kosmetická poradna";

# mozno pouzit az 7 css souboru
$css1 = "zbozi";
$css2 = "formulare";
$css3 = "lucka";
$css4 = "";
$css5 = "";
$css6 = "";
$css7 = "";

# doporuceno ponechat 75 procent
$komprese_nahledu  = 75;   # pro obrazek

#####################################################
###   velikost stranky
#####################################################

$sirka_vlevo  = 188;
$sirka_pravo  = 188;
$sirka_celkem = 788;
```

```
# automaticky upravi sirku uprostred, nemusite upravovat
$sirka_center = $sirka_celkem-$sirka_vlevo-$sirka_pravo;



#####################################################
###   vzhled vypisu zbozi
#####################################################

$na_strane                = 5;     # kolik je na jedne strance vyrobku

$styl = 2;
#     1=obrazek vlevo                              2=obrazek vlevo, ikonky pod textem
#     3=obrazek nahore, ikonky dole vedle textu        4=obrazek, zbozi a ikony pod
sebou

$sirka_pro_vypis_vyrobku    = 370;
$sirka_pro_text_vyrobku     = 250;

$zobrazovat_ikonku_detail   = true;

$format_nahledu_obrazku       = "nahled=false&x=110&";     # zobrazeni obrazku,
#$format_nahledu_obrazku       = "nahled=false&x=110&y=110&";
                                        # pokud dame jen x nebo y, druhy parametr
se dopocita

$format_nahledu_obrazku_v_akci = "nahled=false&x=100";     # zobrazeni obrazku,
# pokud nechame y volne, tak parametr y se dopocita


$obrazky_nehlavni = false;
$obrazky_nehlavni_detail_vyrobku = true;
# $obrazky_nehlavni=false                                VYPISOVAT I NEHLAVNI
OBRAZKY
# $obrazky_nehlavni_detail_vyrobku = true;   VYPISOVAT I NEHLAVNI OBRAZKY v
detail vyrobku

$vykreslit_hlavni_obrazek=true;
$vykreslit_hlavni_obrazek_detail_vyrobku=true;
# $vykreslit_hlavni_obrazek=true                         VYPISOVAT OBRAZEK
(HLAVNI OBRAZEK)
# $vykreslit_hlavni_obrazek_detail_vyrobku=true; VYPISOVAT OBRAZEK (HLAVNI
OBRAZEK) v detail vyrobku



#####################################################
###   kategorie
#####################################################
```

```
$oddelovac_kat_start =  "li class=horni"; # napr "span class=xh"; a dle urovne
hloubky stromu to je  xh1, xh2, xh3 ...
$oddelovac_kat_end   =  "/li";          # /span
$oddelovac_mezi_kat  =  ""; # napr <br>
$zobrazovat_ikonky_pro_rozklik_kategorii = false;

#####################################################
###   administrace
#####################################################

$administrace_obr_pocet_sloupcu=8;

#####################################################
###   objednavaci email
#####################################################

$dekujeme="Dekujeme za Vaši objednavku a tesime se na dalsi
spolupraci.\n\nKosmetika Lucie";

#########################################################################
# id musi byt nemenne, musi to byt vzdy id_zbozi
# minimalne musi byt definovany 0,1,2
# CENA musi byt definovana
# u ADMINA u vyrobku se nahled dela na pole 0,1,2
# PARAMETR je název SLOUPCE v tabulce ZBOZI
# AKCE je definovana jako parametr
# vypsat vsechny promenne tabulky ZBOZI

# $p_vyrobek[1]["parametr"]="nazev";
# $p_vyrobek[1]["start_tag"]="<h1>"; - html tag pred zacatkem vypisu tohoto
parametru
# $p_vyrobek[1]["end_tag"]="</h1>";  - html tag na konci vypisu tohoto parametru
# $p_vyrobek[1]["type"]="text";      - v administraci, v jakem formatu ma byl formular
#                                                          - moznosti
{text,textarea}
# $p_vyrobek[1]["popis"]="název";    - v administraci, popis u formulare

# pouze tyto polozky uzivatel uvidi
$zobraz_promennych          = 6;    # nula pro zobrazeni vsech promennych vsechny
$zobraz_promennych_v_detailu = 7;

$p_vyrobek[0]["parametr"]="id_zbozi";
$p_vyrobek[0]["start_tag"]="<h1>";
$p_vyrobek[0]["end_tag"]="</h1>";

$p_vyrobek[0]["tridit"]="nazev";       # podle ceho budeme tridit zbozi (order by)
$p_vyrobek[0][-10]["where"]="vyrobce"; # vypis vsech vyrobku od jednoho vyrobce
# nazev i vyrobce jsou promenne tabulky (tedy je to parametr)

$p_vyrobek[1]["parametr"]="nazev";
```

```
$p_vyrobek[1]["start_tag"]="<h1>";
$p_vyrobek[1]["end_tag"]="</h1>";
$p_vyrobek[1]["type"]="text";
$p_vyrobek[1]["popis"]="název";

$p_vyrobek[2]["parametr"]="vyrobce";
$p_vyrobek[2]["start_tag"]="<div style='margin:0;margin-top:5px;margin-
bottom:2px;'>výrobce: <b>";
$p_vyrobek[2]["end_tag"]="</b></div>";
$p_vyrobek[2]["type"]="text";
$p_vyrobek[2]["popis"]="vyrobce";

$p_vyrobek[3]["parametr"]="rada";
$p_vyrobek[3]["start_tag"]="<div style='margin:0;margin-bottom:5px;'>kosmetická
rada: <b>";
$p_vyrobek[3]["end_tag"]="</b></div>";
$p_vyrobek[3]["type"]="text";
$p_vyrobek[3]["popis"]="rada";

$p_vyrobek[4]["parametr"]="popis";
$p_vyrobek[4]["start_tag"]="<h3>";
$p_vyrobek[4]["end_tag"]="</h3>";
$p_vyrobek[4]["type"]="text";
$p_vyrobek[4]["popis"]="popis";

$p_vyrobek[5]["parametr"]="objem";
$p_vyrobek[5]["start_tag"]="<b>";
$p_vyrobek[5]["end_tag"]="</b> <br>";
$p_vyrobek[5]["type"]="text";
$p_vyrobek[5]["popis"]=" objem";

$p_vyrobek[6]["parametr"]="cena";
$p_vyrobek[6]["start_tag"]="<b>";
$p_vyrobek[6]["end_tag"]="</b>,- Kc";
$p_vyrobek[6]["type"]="text";
$p_vyrobek[6]["popis"]="cena";

$p_vyrobek[7]["parametr"]="info";
$p_vyrobek[7]["start_tag"]="<div style='margin-top:15px;color:white'>";
$p_vyrobek[7]["end_tag"]="</div>";
$p_vyrobek[7]["type"]="textarea";
$p_vyrobek[7]["popis"]="info";

$p_vyrobek[8]["parametr"]="url_nazev";
$p_vyrobek[8]["start_tag"]="";
$p_vyrobek[8]["end_tag"]="";
$p_vyrobek[8]["type"]="text";
$p_vyrobek[8]["popis"]="url_nazev";
?>
```

2.6. Prídavné moduly

Jelikož lze ocekávat, že mnoho uživatelu bude mít své specifické požadavky na svuj konkrétní e-shop, pokusil jsem se vytvorit nekolik prídavných modulu vhodných pro náš príklad s internetovou kosmetikou. Všechny nove vytvorené moduly budou dostupné pro nové instance e-shopu a prechod z nižší verze e-shopu na vyšší bude zcela bezproblémový.

2.6.1. Texty v e-shopu

Zajisté bude zapotrebí mít v e-shopu ruzné prídavné texty. Rozložení textu zajistí implementátor jednoduchou funkcí vypsat_text(*"kontakt"*). **Uživatel nemá možnost menit rozložení textu, ale má možnost menit jejich obsah. Muže používat html tagy.**

nazev	text
kontakt	<h1>provozovatel:</h1> Lucie Krayzlová 512745 - Těškovice IČO: 73002445 <p> Kosmetička s několikaletou praxí<p> info@kosmetika-lucie.cz

Implementace je velice jednoduchá.

Pokud budeme chtít použít libovolné parametry a vytvorit vlastní texty použijeme první parametr page (napr. http://www.moje_domena.cz/?page=nejaky_page&další_parametry), poté v souboru index.php doplníme rádek, kde k urcitému parametru page priradíme výpis. Napríklad:

```
elseif ($page=="nejaky_page")        vypsat_text("kontakt");
```

V tomto prípade dojde k vypsání stránky, kde na urcité pozici, kde je tento kód, bude vypsán text. Vetšinou to bude stred stránky.

2.6.2. Clánky

Tento modul nemá žádné speciální administracní rozhraní, správa probíhá pres phpmyadmin. Jeho podoba je následující:

url_clanku : jakou podobu bude mít v URL, pokud neexistuje clánek s touto URL, hledá se kategorie s danou URL, zda nacíst clánek nebo kategorii se urcuje automaticky, není nutné mít jediný clánek

nazev: název clánku

text: hlavní text clánku (nacítá se require(*"inc/web/clanek_uvod.php"*);)

uvod: úvodní text (nacítá se require(*"inc/web/clanek_text.php"*);)

speciální hodnotou URL_clanku je název pro **rozcestník** seriálu (definovaný parametrem $nazev_rozcestniku_clanku), krome výpisu všech clánku muže obsahovat i nejaké další informace nebo prídavný formulár, tyto prídavné údaje se nachází ve sloupci text.

2.7. Implementace

2.7.1. Úvod

Cílem bylo vytvorit skripty vhodné pro každý webový stroj s ponecháním implicitního nastavení a za použití mod rewrite.
Na rozdíl od jiných e-shopu a internetových produktu jsem tato pravidla dodržoval.

1/ web musí bežet s nastaveným **open base dir**

nekteré aplikace, se kterými jsem se setkal potrebovaly nastavení open base dir na NONE, pricemž open base dir urcoval, od které úrovne smí skript obsluhovat další soubory (napr. pokud je to /home/www/muj-obchod.tld/ tak tento skript nikdy neprecte /home/www/frantisek.com)

2/ **safe mod ON,** zapnutý safe mod, nebo-li bezpecný režim, pri vypnutém safe-mod (off) jsou nekteré nebezpecné funkce pro bezpecný chod povoleny ci neomezeny (exec, **mkdir**..)

3/ možnot fungovat s **register_globals nastavenými na OFF**, nebo-li každá promenná musí být registrována (pokud v url máme ?page=abc, musíme ve skriptu volat $_REQUEST[page], nebezpecí spocívá v tom, že by uživatel mohl parametr „podstrcit“ napríklad v url)

4/ nastavení chybových stránek (stránky 404, 503 ..)

2.7.2. Popis struktury

Soubory a adresáre jsem rozdelil podle následujících pravidel:

Korenový adresár

V korenovém adresári jsou pouze skripty, které potrebují využívat ostatní funkce, tak jako hlavní soubor index.php pro web a další skripty pro nastavení kategorií. Tyto skripty se volají jako samostatná url. (napr. mujobchod.cz/email.php – odešle email s objednávkou)

Tímto zpusobem je to rozdeleno proto, aby se pri zmene grafické části nemusel implementátor zajímat o nekteré duležité funkcnosti a nastavoval pouze veci, které nastavovat smí.

Implementátor by mel v tomto adresári menit pouze soubor index.php (výjimecne contact_email.php a email.php).

Conf

Adresár conf obsahuje základní konfiguraci. Soubor parametry_eshopu.php byl popsán výše.

Oba soubory jsou urceny pro zásah implementátora.

Fce

Veškeré funkce aplikace. Není možné jej upravovat.

Img

Adresár s obrázky, krome obrázku, které jsou uloženy v databázi pro výrobky. Mohou se zde uložit i pdf skripty a další multimediální části prezentacní části webu.

./inc/admin

záležitosti administrace, nemelo by docházet k dalším úpravám souboru

./inc/systém

systémové záležitosti

./inc/web

zde jsou uloženy hlavní stavební kameny pro sestavení e-shopu

styles

css styly

vzory_nastavení

znacné ulehcení pro implementátory a ruzná vzorová nastavení, treba klasický vzhled e-shopu s pravým a levým menu

Zde je popis jednotlivých skriptu, není zapotrebí jej popisovat, v levém sloupecku je poznámka, zdali se jedná o skript pouze pro administraci nebo pro web.

	skript název	**funkce**
admin	admin_heslo.php	vstup do administracniho rozhrani
admin	admin.php	ADMINISTRACNÍ ROZHRANÍ
admin	akce_nastav.php	NASTAVI AKCI VYROBKU
web	contact_email.php	pošle dotaz (akce pro formulár pro modul clanky)
web	email.php	ODESLE email s objednavkou na KLIENTA i DODAVATELE + ulozi zaznam do tab. OBJEDNAVKA
admin	img_delete.php	smaže obrázek
admin	img_change.php	ZMENÍ OBRÁZEK (starý vymaže, nahraje nový)
web	img_load.php	NACTENÍ OBRÁZKU
admin	img_save.php	ULOŽÍ NOVÝ OBRÁZEK
web	index.php	hlavní stránka
admin	kat_del.php	MAŽE KATEGORII
admin	kat.php	pridávaní a editace kategorií a výrobku
admin	kat_save.php	ULOŽÍ NOVOU KATEGORII
admin	kat_zmen.php	ZMENÍ CÍSLO KATEGORIE, POKUD JE TO MOŽNÉ
web	kosik_odeber_cely_jeden_produkt.php	odstrani vsechny polozky u jednoho vyrobku
web	kosik_odeber_jeden_vyrobek.php	odstrani 1kus od 1 vyrobku z kosiku
web	kosik_pridej.php	PRIDÁ VÝROBEK DO KOŠÍKU
web	kosik_txt.php	kosik v textove forme
web	nahled_obrazku.php	NAHLED OBRAZKU
admin	obrazek_nastav.php	PRIDA OBRAZEK K VYROBKU
admin	obr_nastav_jako_hlavni.php	obrazek u vyrobku nastavi jako hlavni
admin	obr_nezobrazovat.php	ODEBERE OBRAZEK OD vyrobku,ale

		NESMAZE JEJ !
admin	obr.php	pridávaní a editace obrázku
admin	obr_zrus_jako_hlavni.php	zruší obrázek jako hlavní, a bude to bežný obrázek
web	vyprazdni_kosik.php	vyprázdní celý košík
admin	zbozi_del.php	SMAŽE VÝROBEK
admin	zbozi_save.php	VLOŽÍ NOVÝ VÝROBEK
	./conf:	
vše	nastaveni.php	nastavení databáze
vše	parametry_eshopu.php	nastavení parametru eshopu
	./fce:	
	fce_kat.php	
	fce_zbozi_vypis_a_katalog.php	
	obecne_funkce.php	
	session.php	
	telo_dokumentu.php	
	./img:	obrázky (mimo obrázky u výrobku eshopu)
	./inc/admin:	
	admin_blank_page.php	text v administraci na uvodní stránce
	admin_help.php	help administrace
	admin_kat.php	seznam kategorii
	admin_menu.php	menu administrace
	admin_pridej_vyrobek.php	pridá jeden výrobek
	admin_seznam_vyrobku.php	seznam výrobku
	admin_web_vyrobky_kat_obr.php	stred administrace, vyrobky (admin_seznam_vyrobku.php), kategorie, obrazky
	form_new_obr.php	pridani noveho obrazku
	zbozi_info.php	detailní info o zbozí
	./inc/system:	
	admin_tajne_heslo.php	priradí heslo do session
	kosik_nahled.php	náhled košíku
	kosik.php	obsah košíku + OBJEDNÁVKA
	mysql_pripoj.php	pripojeni k db
	nacteni_funkci_promenych.php	pro web, nacte funkce a promenne a hlavicku webu
	redir.php	REDIRECT, z nejaké stránky po vyvolání skriptu se vrací zpet
	zbozi_informace.php	detailni info o zbozi na webu (admin ma /zbozi_info.php)
	./inc/web:	
	clanek_text.php	modul clánek - hlavní text
	clanek_uvod.php	modul clánek - úvodní text
	clanky_seznam.php	modul clánek - seznam všech clánku
	horni_menu.php	nastaveni horniho menu

kosik_celkem.php	kosik_celkem.php
kosik_seznam_polozek.php	kosik_nahled.php + STYL
merici_kod.php	zde vložíme vlastní merící kod
posloupnost_k_obsahu.php	napr home - PC - AMD - atlon
seznam_kategorii.php	seznam kategorii (oddelovac_kat_start nastavime v conf)
text_kategorie.php	text kategorie
vypis_zbozi.php	vypíše výrobek

./styles:	css styly
formulare.css	
lucka.css	
Nepouzite	
zbozi.css	

./vzory_nastaveni:	vzorová nastavení
stred_webu_dve_okna.php	

2.7.3. Základní funkce

Není zapotrebí popisovat rekurzivní prístup k výpisu adresáre, ani jakým zpusobem se obrázky ci výrobky nacítají. Tato cást patrí spíše k urcitému náhledu na zdrojový kód.

2.7.3.1. Funkce pro správu kategorií (fce_kat.php)

Tyto funkce jsou základem pro správu kategorií. Umožnují rekurzivní výpis kategorií, jejich editaci a prídávání v administraci, a ješte nekolik dalších funkcí.

Krátký popis funkcí.

#1# function vrat_id_kat($url_nazev)

```
SELECT id_kat FROM `kat` WHERE  url_nazev =  '$url_nazev'
```

#2# function vrat_nazev_url($id)

```
SELECT id_kat,url_nazev FROM `kat` WHERE  id_kat =  '$id'
```

#3# function posloupnost_k_obsahu($tata,$vypsat=false)

vrací strom posloupnosti k obsahu od urcite kategorie

pokud je promenna vypsat = true, potom i vypise (home - pc - amd)

$strom = posloupnost_k_obsahu($aktualni_kat); # *zjisteni posloupnosti ke korenove kategorii*

Array
(
[0] => 0 - korenova kategorie
[1] => 65 - kategorie
[2] => 71 - podkategori
)

#4# function najdi_syny($tata)

vraci seznam vsech podkategori (do libovolne urovne)

Array
(
[0] => 65
[1] => 78
[2] => 71
)

#5# function mezera($velikost)

vykresli mezeru o dane velikosti

#6# function ma_syna($tata)

zjisti, zdali dana kategorie ma podkategorii (sveho syna)

*#7# function vypis($kdo,$strom,$admin,$oddelovac_kat_start="span class=xh",$oddelovac_kat_end="/span",$oddelovac_mezi_kat="
",$zobrazovat_ikonky_pro_rozklik_kategorii=true)*

vypise seznam kategorii az do kategorie kdo (zadaveme kdo = 0)

#8# function vypsat_text_kat($kat)

vypise text dane kategorie, pokud existuje

#9# function existuje_clanek($url_clanku)

zjisti, zdali clanek s url_clanku existuje

pouziva se k urceni, jestli pouzit clanek, nebo vypsat danou kategorii

2.7.4. Funkce pro výpis a správu zboží (fce_zbozi_vypis_a_katalog.php)

Druhý z pilíru e-shopu - funkce pro správu zboží. Umožnuje dle urcitého stylu vypsat výrobky, vypsat výrobky v akci.

Krátký popis funkcí:

/*

#1# nacti_katalog()

SELECT * FROM `zbozi` order by id_zbozi

nacet kompletni katalog vyrobku do pole

#2# vypsat_vyrobky($ak, $id,$p_vyrobek, ...

nacte vyrobku, ktere ma zobrazovat a pouzije funkci - vypsat_vyrobky_jeden_komplet

($ak==-1) and ($id==-1) vypise vsechny vyrobky

($ak ==-1) and ($id==-10) vypise jen jeden druh vyrobku (treba od jednoho vyrobce),

nastavuje se v $p_vyrobek[0][-10]["where"]

($id >= 0) vypise vyrobky aktualni kategorie a jeho podkategorii

jinak vypisuje dany vyrobek

#3# vypsat_vyrobky_jeden_komplet

vypise jeden vyrobek

dle stylu ($styl) nacte $styl

pouzije funkce vypsat_jeden_vyrobek a vypsat_hlavni_obrazek

#4# vypsat_jeden_vyrobek

bez obrazku

#5# vypsat_hlavni_obrazek

pouze hlavni obrazek

```
#6#  vypsat_ostatni_obrazky
vypise obrazky, ktere nejsou hlavni, hlavni je jenom jeden

#7#  vypsat_akci
vypise vyrobky na ktere je akce

#8#  ikonky_detail_kosik
vykreslí ikonka detail, nastav akci, ubrat, smaz

#9#  vrat_id_zbozi($url_nazev)
SELECT  id_zbozi FROM `zbozi` WHERE  url_nazev =  '$url_nazev'

#10# vrat_nazev_url_zbozi($id)
SELECT  id_zbozi,url_nazev FROM `zbozi` WHERE  id_zbozi =  '$id'

#11# vypsat_text($nazev)
SELECT  text FROM `texty` WHERE  nazev =  '$nazev' LIMIT 1"

*/
```

2.7.5. Další funkce

ostatní funkce, uložené v souborech

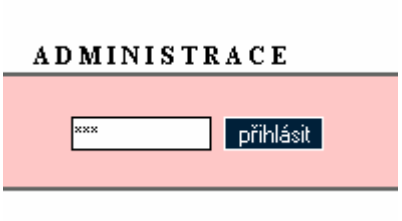
obecne_funkce.php – zde jsou uloženy funkce, které se dají použít pro jiné aplikace, například zjištení o jakou verzi prohlížece se jedná a nactení príslušné verze css.

session.php – uložena funkce pro správu session pro tento produkt

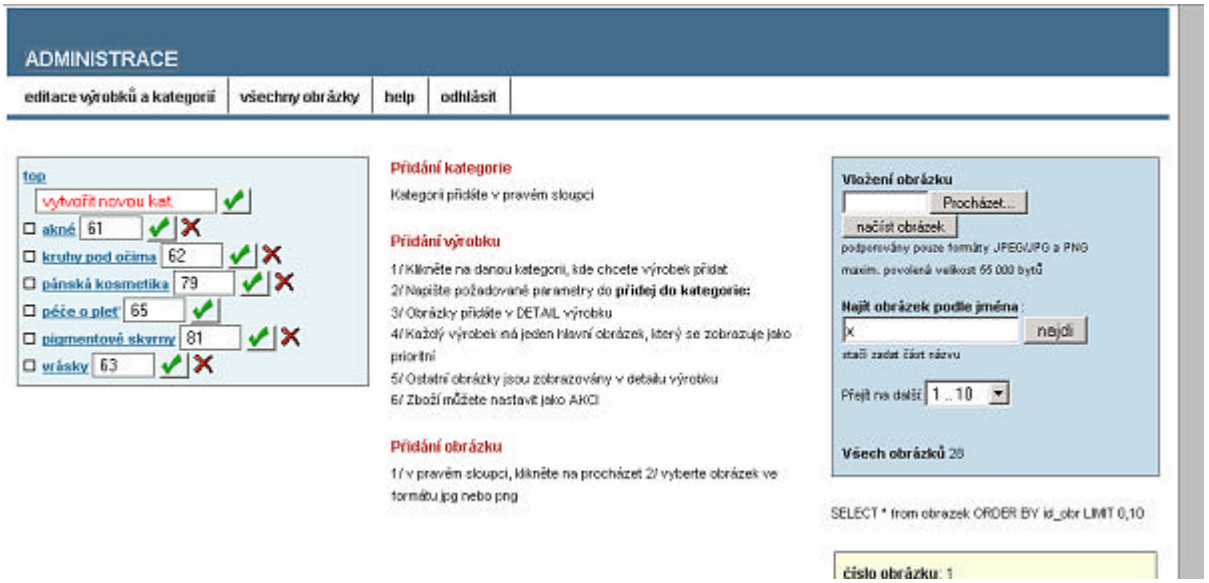
telo_dokumentu.php – funkce pro vytvorení a výpis hlavicky www dokumentu

2.8. Administrace e-shopu

Prihlášení do administrace probíhá na predem stanovené url, pro vetší bezpecnost je možné jméno tohoto skriptu kdykoliv zmenit. I když se jedná pouze o zcela „paranoidní“ krok, avšak každý bezpecností krok je dobrý.



Po prihlášení do administrace se dostáváme na stránku, kde probíhá základní správa e-shopu. V menu se rovnež nachází nápoveda.



pridání/editace/mazání kategorie

V levé části je seznam kategorií - zde mužeme pridávat, mazat a editovat kategorie. Pokud není treba, císla kategorií neprepisujeme, nebot vazba kategorie-zbozi se již nemení.

pridání výrobku

Po nalistování urcité kategorie je možno v levém sloupci dole rovnež pridat výrobek. Vzhled formuláre pro vložení výrobku se nastavuje v konfiguracním souboru (parametry_eshopu.php) a pridává se do promenné.

$p_vyrobek[cislo]["type"] ="textarea"; - daný formulár bude textové pole
$p_vyrobek[cislo]["popis"]="info"; - popisek bude INFO

U výrobku lze nastavit, zda je výrobek v akci.
Prirazení obrázku k výrobku se provádí v detailním zobrazení výrobku.

mazání výrobku

Výrobek nelze editovat pres administracní rozhraní, lze jej pouze smazat a vytvorit nový. Pokud je opravdu nutné výrobek editovat, použijte phpmyadmin.

pridání/editace/mazání obrázku

V pravé cásti administrace je formulár pro vložení obrázku.

Každý obrázek je možné smazat a vymenit za jiný.

Obrázky lze vyhledávat podle jejich názvu.

V menu je položka **všechny obrázky,** zde je seznam všech obrázku. Tuto položku se nedoporucuje spouštet v prípade, pokud máme velkou databázi obrázku.

Prídávné moduly

Prídavné moduly se obsluhují prímo v phpmyadmin. Postupne budeme pridávat obslužné prvky prídavných modulu také u této administrace.

3. Systémová optimalizace (optimalizace databáze a webového serveru)

3.1. Optimalizace MySQL

3.1.1. Proc optimalizovat?

Problémem velkých databázových systému je pomalý prístup k datum, vzhledem k poctu položek, kterých bývají nekdy i tisíce. Je proto nutné rídit se danými pravidly, abychom mohli práci s aplikací maximálne zrychlit.

Doporucuje se rídit následujícími pokyny:

- Peclive zvážit typ sloupcu
- OPTIMALIZOVAT TABULKU
- Mimo primární klíc, používat i další klíce
- Požívat tabulky s pevnou délkou záznamu
- Pokusit se vyhnout složitým dotazum do nekolika tabulek najednou
- Opakující se dotazy zachovat

3.1.1.1. Peclive zvážit typ sloupcu

Celocíselné typy

Pokud ocekáváme v tabulce císlo, je vhodné zvážit, jakých bude nabývat hodnot, napríklad výška cloveka nebude nabývat hodnot do 2 147 483 647, proto nám bude stacit pouze tinyint. Hodne programátoru automaticky vkládá int, je to svým zpusobem pohodlnost, pouze u malých aplikací tento rozdíl ani nepoznáme. Rovnež je vhodné se rozhodnout, zda použijeme i záporné hodnoty, potom mužeme použít unsigned typ hodnot, tímto se nám rozsah v kladných hodnotách zdvojnásobí. Pro názornost, pokud máme tisíce položek, ve kterých použijeme místo unsigned tinyint , zcela zbytecne budeme potrebovat i 3 Byte více.

Type	Bytes	Minimum Value	Maximum Value
		(Signed/Unsigned)	(Signed/Unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INT	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615
FLOAT(M,D)	4	Dle nastavení M,D	Dle nastavení M,D
DOUBLE(M,D)	8	Dle nastavení M,D	Dle nastavení M,D
DECIMAL(M,D)	M+2	Dle nastavení M,D	Dle nastavení M,D

Jen pro úplnost, dalšími číselnými typy jsou Decimal, napr. cena DECIMAL(5,2), kde první parametr udává pocet císlic, a druhý pocet desetinných míst, v našem prípade cena bude mezi -999.99 a 999.99. Obdobne lze použít Float a Double.

Retezce

Castou chybou je nastavování textových promenných na TEXT, doporucuji nastavovat typ CHAR nebo VARCHAR.

3.1.1.2. Krome primárního klíce používat i další

Berme v úvahu databázi s milióny záznamu a nekolika sloupci. Potom vyhledáváme-li urcitý záznam, MySQL projíždí celou tabulku a záznam po záznamu zjištuje, zda-li záznam vyhovuje daným podmínkám. Pokud casto hledáme pouze podle jednoho sloupce, je dobré daný sloupec indexovat MySQL, potom nebude projíždet celou tabulku, ale pouze jeden indexový soubor.

3.1.1.3. Optimalizovat tabulku

Casto máme tabulky vetší, i pres dobre vyrešený výber datových typu. Funkce

OPTIMIZE TABLE `tabulka`; optimalizuje tabulku temito zpusoby:

Pokud v tabulce bylo mazáno nebo záznamy byly rozdeleny, opraví tabulku.

- Pokud indexy jednotlivých položek nejsou serazené, seradí je.
- Pokud statistiky nejsou aktuální (a oprava nemuže být provedena serazením indexu), aktualizuje statistiku.

3.1.1.4. Používat sloupce s pevnou délkou záznamu

Další zrychlení práce MySQL lze provést vhodným výberem u sloupcu, jejichž délka je pevne stanovena. MySQL si spocítá celkovou délku jednoho záznamu a pri procítání *skáce* prímo na zacátek dalšího záznamu. Pokud je v tabulce sloupec typu *text* nebo *varchar*, MySQL nejdríve pri procítání tabulky musí zjistit u prohlíženého záznamu jeho velikost a teprve potom se posune na další.

MySQL si v tomto prípade prevádí datové typy automatizovane. Pokud budeme vytváret tabulku

```
create table klient (
  jmeno char(30),
  poznamka text
);
```

... potom typ sloupce jmeno nebude **char**, ale **varchar**. To z toho duvodu, že tabulka již obsahuje sloupec typu **text**, takže se bude pri procházení tabulky tak jako tak pocítat délka každého záznamu. Rozdíl mezi **char** a **varchar** je takový, že pri ukládání do char doplní MySQL neviditelnými znackami na velikost charu. Automatická konverze probíhá pouze od typu s pevnou délkou na variabilní.

3.1.1.5. Složité sql dotazy, zachování stránek

Budeme-li používat složité sql dotazy, selektující pres nekolik tabulek.

Napr.:

SELECT a.jmeno, b.* FROM klient a, dodavatel b

je dobré si uvedomit, že si mysql vytvorí docasnou tabulku, která bude mít pocet rádku roven soucinu rádku obou tabulek. Tedy pokud tabulky mají 1000 a 1000 záznamu, vytvorí to docasnou tabulku s milionem rádku. Je potom mnohem lepší

ukládat si mezivýsledky a provádet dotazy pomocí optimalizovaných tabulek. Mysql neumí vytvorit klíc pres dve tabulky, proto není možné výše uvedený sql dotaz optimalizovat. Mezivýsledky si mužeme ukládat do promenných ve skriptu.

Dalším rozumným rešením je v prípade, že chci napr. jednou za hodinu zjistit mysql dotazem, urcitou statistiku, která obnáší vykonání složitého sql dotazu, vykonám tento jednou za hodinu a uložím do promenné - treba v log souboru.

3.1.2. Nastavení webového serveru

Pri instalaci webového serveru, již máme nastaveny konfiguracní skripty s parametry daných programu. Vetšinou se však pocítá s tím, že na pocítac instalujeme verzi spíše pro domací použití, než pro servery. Proto je nutné nekde hodnoty omezit, jinde naopak limity zvýšit, nebo úplne zmenit nastavení. Budeme se rídit následující filosofií:

1/ chceme, aby každý mel k dispozici dostatecný výkon
2/ rovnež potrebujeme urcit zlomek skriptu (resp. klientu s weby), který generuje enormní zátež

3.1.2.1. Apache

veškeré nastavení se provádí v souboru apache2.conf
,který se nachází v /etc/apache2

Vedle promenné jsou uvedeny mnou doporucené hodnoty, predpokládající vetší pocet klientu (www stránek, informacních systému), nebot provoz aplikace na jedné doméne by byl nákladný.

Timeout 5
Pocet vterin, po jejichž uplynutí se predpokládá prerušení spojení.
Snížíme na co nejnižší hodnotu, ale opatrne, aby nám pak vubec šla nacítat spojení.

KeepAlive On
Aktivuje tzv. persistentní (trvalé) spojení, kdy muže prohlížec pri jednom spojení žádat více dotazu.Velmi to urychlí prenos, nebot klient nemusí otevírat zbytecne spojení a v jednom sledu muže stáhnout jak stránku, tak i napr. obrázky do stránky vložené.
Povolíme.

MaxKeepAliveRequests 100
Maximální pocet povolených žádostí behem stálého spojení. Nastavení 0 dovoluje neomezené množství.
Nastavíme vysokou hodnotu, v žádném prípade ne neomezenou.

KeepAliveTimeout 5
Je to doba, ve které musí dojít od stejného klienta na jednom spojení další dotaz, aby se vyrídil v rámci jednoho persistentního spojení.
Snížíme na co nejnižší hodnotu, ale opatrne, abychom využili persistentní spojení.

MinSpareServers 8
MaxSpareServers 40
Nastavení týkající se prevážne pouze Unix platformy, které urcuje, kolik instancí serveru bude cekat v dobe necinnosti. Výsledkem je zrychlení odezvy, než kdyby se proces vytvárel až na základe dotazu. Drobnou nevýhodou je zvýšení zatížení systému a odcerpání cásti systémových zdroju.
Pokud máme výkonný stroj, zvýšíme.

StartServers 16
Pocet serveru (virtuálních), který se vytvorí po spuštení serveru.

MaxClients 100
Direktiva omezuje maximální pocet soucasne bežících procesu serveru. Tím se limituje i maximální pocet soucasných spojení. Není vhodné nastavovat príliš nízké císlo, nebot pak pri zatížení mnoho klientu nepríjemne "zatuhne" pri cekání na spojení nebo nahlásí chybu pripojení. Po prekrocení poctu procesu by se server zacal zpomalovat, takže by nakonec nedokázal v limitu vyrizovat pricházející dotazy. Konkrétní nastavení tedy záleží na výkonu pocítace.
Nastavíme vysokou hodnotu, v žádném prípade ne neomezenou.

3.1.2.2. Mysql

Podstatne jednodušší je konfigurace mysql databáze. Je nutno ji nastavit v souboru **my.cnf** (umísten v /etc/mysql).

set-variable = max_connections=1000
(pokud rádek neexistuje, doplníme za [mysqld])

udává, pocet povolených pripojení k databázi mysql

max_questions
Následne omezíme klienty, aby nemohli pretížit databázi špatnými skripty, nebo jakýmkoliv jiným zpusobem. Toto omezení se provádí v databázi mysql, v tabulce user, sloupci max_questions a udává maximální pocet sql dotazu za hodinu. Pokud daný uživatel databáze limit prekrocí, znemožní mu to další práci s databází a obnoví se až v další hodine.

Uvažujme, že každé nactení sql stránky bude potrebovat max. 50 sql dotazu (extrémní prípad www aplikace), a prepokládejme 1000 zobrazení za hodinu (což je dle mého názoru dostacující i pro casto navštevovaný portál). Tedy 50 x 1000 = 50 000sql dotazu za hodinu. Toto nastavíme a poté reloadneme (nebo restartujeme) mysql .

Pokud v nekterých prípadech bude docházet k prekracování limitu, lze tyto zvýšit, pod podmínkou, že je aplikace dobre napsána a že nedochází k zbytecnému zatežování systému.. V žádném prípade nenastavujeme neomezenou hodnotu (nulu).

3.1.2.3. Php

veškeré nastavení provádíme v souboru php.ini,
který se bude pri použití apache2 nacházet v /etc/php4/apache2/

Vedle promenné jsou uvedeny mnou doporucené hodnoty, predpokládající vetší pocet klientu (www stránek, informacních systému), nebot provoz aplikace na jedné doméne by byl nákladný.

Hodnoty, které se snažíme maximálne snížit, mohou výjimecne vyvolat komplikace, napr. problém muže nastat pri importu dat do databáze pres phpmyadmin, proto tuto hodnotu vždy krátkodobe zvýšíme.

max_execution_time = 5
PHP skript má maximálne 5 sekund na svou cinnost.
Maximálne snížíme.

memory_limit = 4M
Tento parametr urcuje, kolik pameti smí maximálne PHP skript zabrat. Standardne bývá nastaveno 8 MB.
Maximálne snížíme.

3.1.3. Návrh hw optimalizace

Testy, které byly provádeny se týkají pouze operacního systému LINUX, konkrétne distribuce Debian (debian.org), s jádrem 2.6.

Uptime :

08:53:33 up 177 days, 10:14, 2 users, load average: 2.90, 3.52, 3.43
První cást udává, jak dlouho beží server bez restartu (177dní), druhá - kolik uživatelu je prihlášeno.

Poslední hodnotou je load za poslední minutu, pet minut a za posledních 15 minut. Load znamená, kolik procesu soucasne pristupuje k procesoru.

0-0.9999 - procesor se chvílemi nudí.
1 - procesor je neustále vytížen, ale žádné procesy necekají.
1< - najednou se snaží více procesu pristupovat k procesoru.

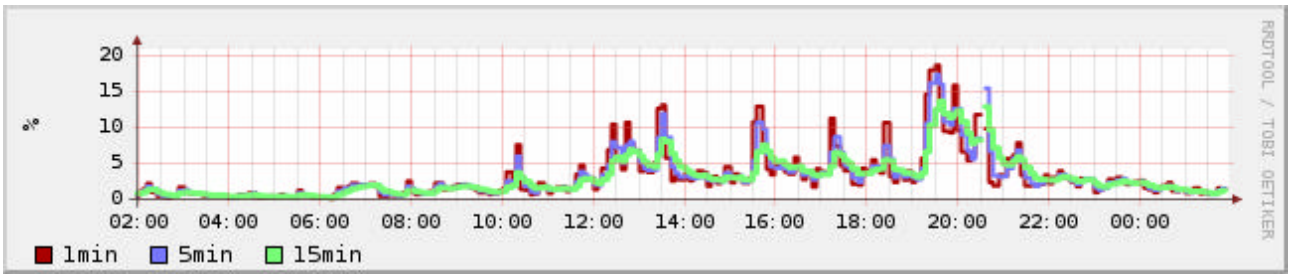
Load a prumerné zatížení nemají moc spolecného. Load ukazuje pouze, kolik procesu chce práve obsloužit od procesoru, nic víc. Server, který bude mít load 1 muže být rádove více zatížen, než server s load 100.

Pokud tedy bude load stabilne výrazne vyšší než jedna, meli bychom rozhodne uvažovat o upgrade hw.

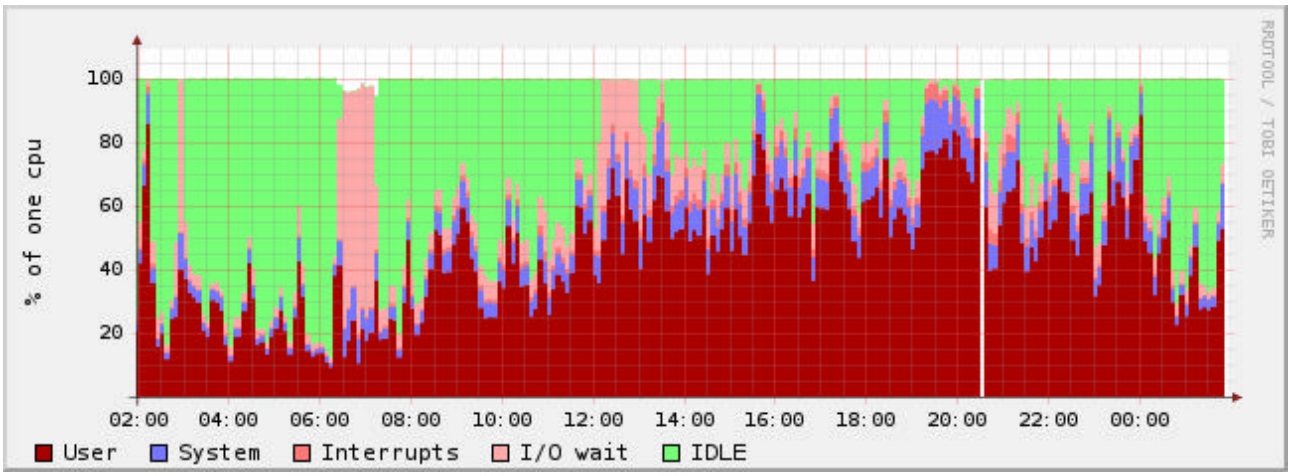
Rozebereme nekteré statistiky (nejlépe za celý den), k dispozici byl server s

Board Intel 815
CPU PIII 1GHz
2 x HD 200GB UATA

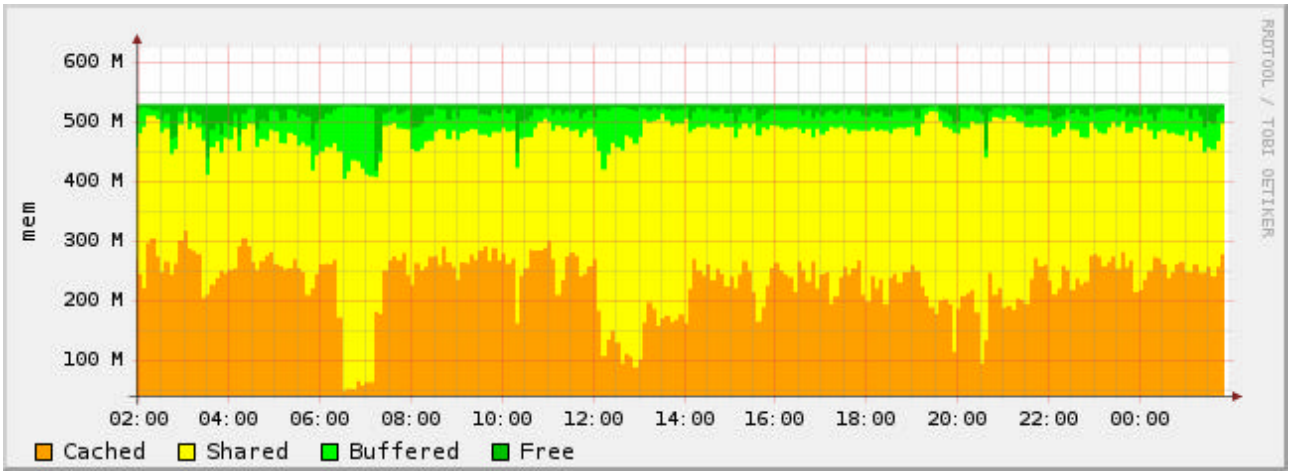
Prumerný denní load, jak je možno videt, tak ve špicce cekalo více jak 10 procesu ve fronte, z cehož vyplývá, že soucasná hardwarová konfigurace je nedostacující.



Jedná se o jednoprocesorový server, z druhého grafu vyplývá, že procesor (CPU) byl v nekterých situacích velice zaneprázdnen, provádel predevším procesy uživatelu (v našem prípade Apache a Mysql). Situace však není nejhorší, je zde jistá rezerva.



Dobrým krokem je upgrade operacní pameti, máme k dipozici celkove 512MB pameti. Z grafu je videt, že pamet byla témer celá využita behem celého dne a nikoliv pouze ve špicce. Jelikož je aplikace provozována na php na linuxovém stroji a o linuxu je známo, že dokáže dobre využívat pamet, bude tedy vhodné pamet rozšírit.



Velice duležitým faktorem je i volba pevných disku, jelikož cena SCSI disku je stále nekolikanásobne vyšší než-li SATA disku. Je vhodné si zjistit, zda nám treba nevystací disky o velikosti 73Mb. Jelikož pro nás bude duležitá bezpecnost, mely by disky být v RAID 1, z tohoto duvodu bude zapotrebí mít dva. Pokud bude nutné mít

vetší datový prostor a nemáme-li zrovna k dispozici 50 tisíc, budeme si muset vystacit se SATA disky. (V linuxu se zjištení místa provádí príkazem df).

Rovnež jsem mel k dispozici server s 3GB pametí. Pri pohledu na tento server, je zrejmé, že linux opravdu dokáže výborne využít operacní pamet, z 3GB operacní pameti nám nechává volný pouze zlomek.

```
[root@b07]:[~]# free -m
             total       used       free     shared    buffers     cached
Mem:          3035       2915        120          0        155       1859
-/+ buffers/cache:        901       2134
Swap:          999          0        999
```

V prípade, že bychom meli pouze nejaké nárazové zmeny chování pocítace, enormní zátež by byla znatelná pouze v urcitých okamžicích, mužeme použít príkaz top, nebo vylepšený príkaz htop (viz. obrázek , pod klávesou F1 je strucná nápoveda), a dohledat si príslušný program, usera, který daný problém zpusobil.

```
  CPU[|||||||||                    26.0%]     Tasks: 85 total, 1 running
  Mem[||||||||||||||||||||168/503MB]     Load average: 0.24 0.43 0.68
  Swp[||                           86/1906MB]     Uptime: 118 days, 09:02:41

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
14439 root       15   0 26180 13012 18956 S 16.2  1.9  0:00.72 /usr/sbin/apache2
14700 root       18   0  187M 66724  9000 S  5.8  9.7  0:00.09 /usr/sbin/mysqld -
14691 root       16   0  2152  1116  1956 R  1.3  0.2  0:00.22 htop
25508 root       16   0  187M 66724  9000 S  0.6  9.7 26:57.85 /usr/sbin/mysqld -
14292 root       16   0 25892 12736 18972 S  0.0  1.8  0:01.89 /usr/sbin/apache2
14428 root       15   0 25744 11596 19784 S  0.0  1.7  0:00.28 /usr/sbin/apache2
14681 root       16   0  6236  1772  5800 S  0.0  0.3  0:00.04 sshd: root@pts/0
14631 root       15   0 25732 12360 18972 S  0.0  1.8  0:00.28 /usr/sbin/apache2
14469 root       15   0 30048 16152 19784 S  0.0  2.3  0:01.63 /usr/sbin/apache2
14632 root       16   0 24656 11008 18972 S  0.0  1.6  0:00.02 /usr/sbin/apache2
14492 root       15   0 25716 12992 18984 S  0.0  1.9  0:00.89 /usr/sbin/apache2
13895 root       15   0 30376 16504 19784 S  0.0  2.4  0:04.65 /usr/sbin/apache2
14478 root       15   0 25700 12336 18956 S  0.0  1.8  0:00.51 /usr/sbin/apache2
    1 root       16   0  1504   408  1352 S  0.0  0.1  0:35.07 init [2]
    2 root       34  19     0     0     0 S  0.0  0.0  0:00.88 ksoftirqd/0
    3 root        5 -10     0     0     0 S  0.0  0.0  0:03.13 events/0
    4 root        7 -10     0     0     0 S  0.0  0.0  0:00.00 khelper
 1Help  2Setup  3Search  4Invert  5Tree  6SortBy  7Nice -  8Nice +  9Kill  10Quit
```

program htop (vylepšená verze programu top)

```
top - 20:06:04 up 119 days,  2:19,  5 users,  load average: 2.61, 2.49, 3.95
Tasks: 122 total,   5 running, 117 sleeping,   0 stopped,   0 zombie
Cpu(s): 56.2% us, 11.4% sy,  0.0% ni, 12.1% id, 16.7% wa,  0.3% hi,  3.3% si
Mem:    515712k total,   511984k used,     3728k free,    32852k buffers
Swap:  1951800k total,    93448k used,  1858352k free,   240964k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
16010 www-data  15   0 28076  10m  19m S  9.5  2.1   0:00.44 apache2
16093 www-data  16   0 28168  10m  19m R  8.2  2.1   0:00.56 apache2
16202 www-data  15   0 25528 9000  18m S  7.2  1.7   0:00.40 apache2
16084 www-data  16   0 26032 9452  18m S  3.9  1.8   0:00.29 apache2
16219 www-data  15   0 25084 8396  18m S  3.9  1.6   0:00.12 apache2
16013 www-data  15   0 25804 9284  18m S  2.9  1.8   0:00.55 apache2
16216 www-data  15   0 25052 8400  18m S  2.6  1.6   0:00.09 apache2
16228 www-data  15   0 25944 8528  19m S  2.3  1.7   0:00.13 apache2
16229 www-data  15   0 25076 8488  18m S  2.3  1.6   0:00.14 apache2
15956 www-data  15   0 25028 8536  18m S  1.6  1.7   0:00.42 apache2
16237 www-data  16   0 24452 7696  18m S  1.6  1.5   0:00.05 apache2
15644 www-data  16   0 25148 8452  18m R  0.7  1.6   0:00.38 apache2
16070 www-data  16   0 28792  11m  19m S  0.7  2.3   0:00.65 apache2
16239 root      16   0  2064 1080 1856 R  0.7  0.2   0:00.06 top
 2037 root      15   0  4972  872 4772 S  0.3  0.2   1:49.66 master
18143 root      16   0 24144 6592  18m R  0.3  1.3   0:20.50 apache2
15743 www-data  16   0 28184  10m  19m S  0.3  2.1   0:02.13 apache2
```

top

nebo mužeme použít pouze výpis pameti **free -m**

```
[root@b08]-[~]# free -m
             total       used       free     shared    buffers     cached
Mem:           503        490         13          0         34        233
-/+ buffers/cache:        222        281
Swap:         1906         91       1814
```

nebo výpis load a uptime serveru

```
[root@b08]-[~]# uptime
 20:09:53 up 119 days,  2:23,  5 users,  load average: 3.65, 3.21, 3.94
```

pouze procesy príkazem

ps aux

```
[root@b08]-[~]# ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   1504   408 ?        S     2005   0:35 init [2]
root         2  0.0  0.0      0     0 ?        SN    2005   0:00 [ksoftirqd/0]
root         3  0.0  0.0      0     0 ?        S<    2005   0:03 [events/0]
root         4  0.0  0.0      0     0 ?        S<    2005   0:00 [khelper]
root        23  0.0  0.0      0     0 ?        S<    2005  30:20 [kblockd/0]
root        46  0.0  0.0      0     0 ?        S<    2005   0:00 [aio/0]
root        45  0.0  0.0      0     0 ?        S     2005 102:26 [kswapd0]
root       182  0.0  0.0      0     0 ?        S     2005   0:00 [kseriod]
root       270  0.0  0.0      0     0 ?        S     2005   0:01 [md1_raid1]
root       273  0.0  0.0      0     0 ?        S     2005   2:28 [md0_raid1]
root       305  0.0  0.0      0     0 ?        S     2005  37:56 [kjournald]
root       602  0.0  0.0      0     0 ?        S     2005   3:41 [jfsIO]
root       603  0.0  0.0      0     0 ?        S     2005  17:50 [jfsCommit]
root       604  0.0  0.0      0     0 ?        S     2005   0:03 [jfsSync]
root       844  0.0  0.0      0     0 ?        S     2005   0:00 [khubd]
root      1821  0.0  0.1   2260   712 ?        Ss    2005   9:09 /sbin/syslogd
www-data 19323  0.0  1.8  25728  9364 ?        S    20:22   0:00 /usr/sbin/apach
www-data 19325  0.0  1.7  25424  9052 ?        S    20:23   0:00 /usr/sbin/apach
www-data 19327  0.0  1.8  25912  9588 ?        S    20:23   0:00 /usr/sbin/apach
www-data 19334  0.0  1.8  25844  9540 ?        S    20:23   0:00 /usr/sbin/apach
www-data 19335  0.0  1.6  25044  8572 ?        S    20:23   0:00 /usr/sbin/apach
www-data 19336  0.0  1.7  25436  9024 ?        S    20:23   0:00 /usr/sbin/apach
www-data 19338  0.0  1.7  25712  9268 ?        S    20:23   0:01 /usr/sbin/apach
```

3.1.4. Ukázka optimalizace na strane uživatele

Pro náš príklad máme tabulku klient:

```
CREATE TABLE `klient` (
 `id` bigint(20) NOT NULL auto_increment,
 `vek` bigint(20) NOT NULL default '0',
 PRIMARY KEY  (`id`)
);
```

Vyplníme ji napr. 1000 položkami s vekem 25

```
INSERT INTO `klient` VALUES (null, 25);
```

Velikost tabulky je nyní 48 296 bajtu. Z toho:

1000 zaznamu, bigint,
bigint

data	26 792
index	21 504
navíc	9 792
efektivni	38 504
celkem	**48 296**

Hodnoty jsou uvedeny v bajtech.

Nyní optimalizujeme tabulku sql príkazem:

```
OPTIMIZE TABLE `klient`;
```

po optimalizaci
tabulky

Data	17 000
Index	14 336
Celkem	**31 336**

Je zde videt, že jsme ušetrili více místa, než bylo puvodne avizováno.

Nyní se zameríme na cást týkající se úpravy datových typu. V obou sloupcích lze použít kladné hodnoty. Vek zmeníme na TINYINT UNSIGNED (dostacující budou hodnoty 0 – 255) a primární klic nastavíme na INT UNSIGNED (0 až cca 4,3 miliardy). Po zmene typu a po optimalizaci tabulky, získáváme tabulku zhruba 3x menší.

(príkaz pro zmenu datového typu)

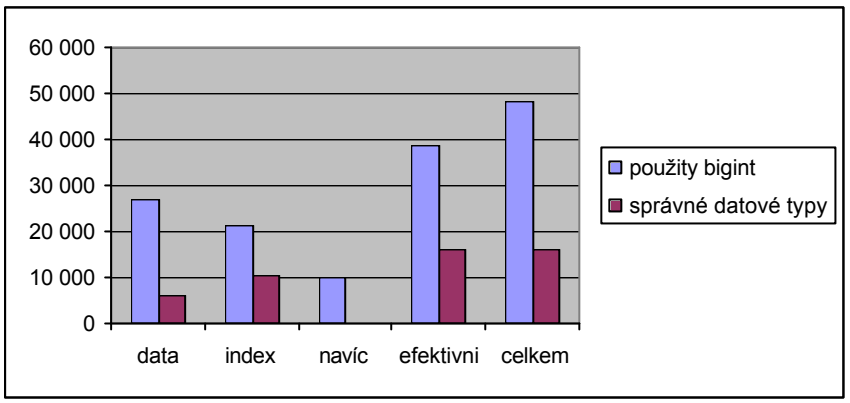
```
ALTER TABLE `klient` CHANGE `id` `id` INT UNSIGNED NOT NULL
AUTO_INCREMENT ,
CHANGE `vek` `vek` TINYINT UNSIGNED NOT NULL DEFAULT '0'
```

po uprave data typu

data	6 000
index	10 240
celkem	**16 240**

celkem ušetríme na 1000 položkách

místo	**32 056**
neoptimalizovaná tabulka je	2,97 x vetší



V prípade, kdy bychom meli pracovat s tabulkou o velikosti 48 MB, budeme díky optimalizaci pracovat s tabulku o velikosti pouhých 16MB. I za predpokladu, že máme silný server, se to bude lépe zálohovat.

Nyní k tabulce pridáme sloupce jméno, s implicitním jménem Ladislav.

Celá tabulka:

```
CREATE TABLE `klient` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `vek` tinyint(3) unsigned NOT NULL default '0',
  `jmeno` varchar(50) NOT NULL default 'ladislav',
  PRIMARY KEY  (`id`),
  KEY `jmeno` (`jmeno`)
);
```

Byla zmenena jména na dvou pozicích, na 555-té pozici (radovan) a 750-té pozici (lucie). Nyní se ptáme na jméno radovan a lucie pomocí sql dotazu.

```
SELECT * FROM `klient` WHERE `jmeno` LIKE 'lucie'
```

Detailní informace o selectu získáme príkazem EXPLAIN.

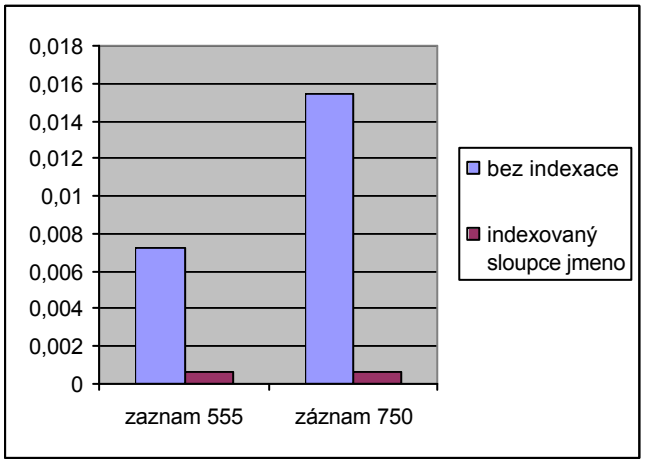
```
EXPLAIN SELECT *  FROM `klient` WHERE jmeno LIKE "lucie"
```

	indexované sloupce jmeno	bez indexace	Vysvetlení
id	1	1	id vysvetlujícího SELECTu
select_type	SIMPLE	SIMPLE	druh selektu (simple, primary,union …)
table	klient	Klient	tabulka na kterou provádíme select
type	RANGE	ALL	typ spojeni s tabulkou, od nejlepšího k nejhoršímu : system, const, eq_ref, ref, range, index, all.
possible_keys	jmeno	NULL	možné klíce, které mužeme použít
key	jmeno	NULL	použitý klíc
key_len	50	NULL	jakou délku klíce se mysql rozhodlo použít, menší je lepší
ref	NULL	NULL	ref sloupce ukazuje, který sloupec nebo konstanta je porovnávána s názvem indexu v klíci
rows	5	9538	pocet rádku, které mysql použila pro zpracování sql dotazu
extra	Using where	Using where	bližší informace

Bez indexace potrebuje mysql pro zpracování všechny rádky, pokud však máme indexaci na dané sloupce, použije jen 5 rádku k vykonání sql dotazu, jak jsme si ukázali.

Poté byla provedena indexace tabulky jméno a došlo k nekolikanásobnému zrychlení vyhledávání.

```
ALTER TABLE `klient` ADD INDEX ( `jmeno` )
```



Jednalo se o cca 9500 záznamu, indexový soubor obsahoval pouze nekolik jmen, casy jsou uvedeny v sekundách.

Pokud bychom zde meli napríklad i položku príjmení a chteli bychom vyhledávat soucasne podle jména i príjmení, je nutné vytvorit klíc s obema atributy.
ALTER TABLE `klient` ADD INDEX **(jmeno, prijmeni)**

Po odebrání indexu na sloupec jméno, nastavíme sloupec jméno na datovou hodnotu TEXT. Abychom overili duležitost správného nastavení textového typu.

ALTER TABLE `klient` CHANGE `jmeno` `jmeno` TEXT NOT NULL
(musíme zrušit defaultní hodnotu)

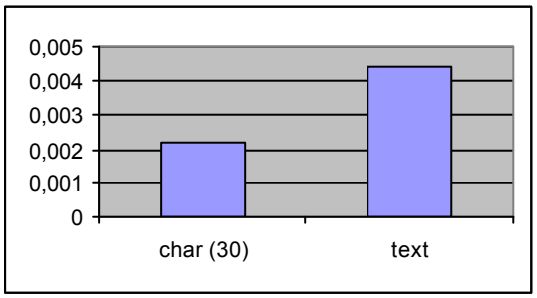
Provedeme jednoduchý sql dotaz,

```
SELECT *
FROM `klient`
WHERE jmeno LIKE "radovan"
```

Dotaz se vykonal za 0,0022 sec., po zmenení datového typu na typ s pevnou délkou.

ALTER TABLE `klient` CHANGE `jmeno` `jmeno` CHAR(10)

Se daný dotaz provádel 0,0044 sec., tedy dvakrát tak dlouho.



4. SEO (optimalizace pro vyhledávací služby)

S velkým rozmachem fulltextového vyhledávání prišla zcela logicky snaha webmasteru, umístit se ve vyhledávacích pokud možno na predních místech. Tyto techniky dostaly odborný název SEO.

4.1. Úvod

Pokud chceme získat urcité informace na internetu, napr.o firme provozující internetovou kosmetickou poradnu, máme nekolik možností, jak se k požadovaným informacím dostat:

- nekdo nám sdelí adresu webstránky (málo pravdepodobné)
- nalistujeme si nejaký katalog, napriklad seznam.cz, zde procházíme strukturou katalogu, abychom se k urcité stránce dostali, je zapotrebí, aby majitel, nebo tvurce stránek firmu do katalogu zaradil. Na zarazení do katalogu na centrum.cz a seznam.cz jsme v našem prípade **cekali zhruba týden**. Pricemž krátce po zarazení do katalogu vetšinou ihned nekdo z techto portálu volá a pokouší se **vás premluvit**, abyste využili placenou reklamu.

- velmi castý zpusob, jak získat požadovanou informaci je takový, že zadáme výraz (slovo, nebo více slov), který hledáme do nekterého z vyhledávacu. Za nejlepší považuji www.google.com , který obsahuje i vyhledávaní v ruzných jazycích. Z ceských vyhledávacu je to juxo.cz a také fulltextový vyhledávac od seznam.cz. Pri optimalizaci pro vyhledávace se stránka objeví vetšinou ve všech velkých vyhledávacích, temto technikám jak získat prední pozice ve vyhledávacích se ríká SEO, tedy "Search Engine Optimization" **optimalizace pro vyhledávace**. SEO (optimalizace pro vyhledávací služby)

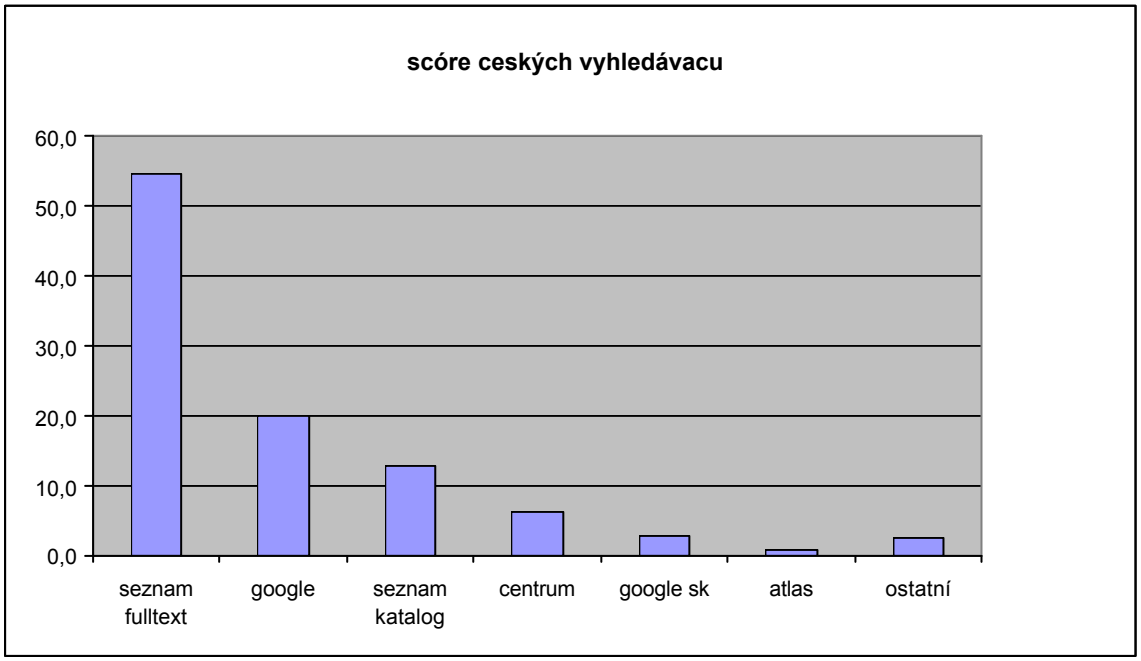
4.2. Jak optimalizovat, na co se zamerit (duležitost google a seznam.cz)?

4.2.1. Scóre ceských vyhledávacu

Pri optimalizaci pro vyhledávace je vhodné si uvedomit, jakým pomerem lidé jednotlivé vyhledávace využívají. Využil jsem verejná data z http://www.toplist.cz/global.html ze dne 5.února 2006, a je zcela zrejmé, že více než polovina lidí chodí z vyhledávace na seznam.cz, a dalších cca 13 procent prichází z jeho katalogu, tedy seznam.cz má cca 68 procent lidí pricházejích ze seznam.cz, 20 procent nás navštíví z vyhledávace google.com, kupodivu pouze 6 procent má centrum.cz využívající vyhledávace morfeum. A zcela zanedbatelné necelé jedno procento má altas.cz, vyživající jyxo. 2,5 procenta patrí google.de, toplist.cz, google obrázky a zoznam.sk, což pro nás nemá již témer žádnou hodnotu, nebot se jedná prevážne o cizince.

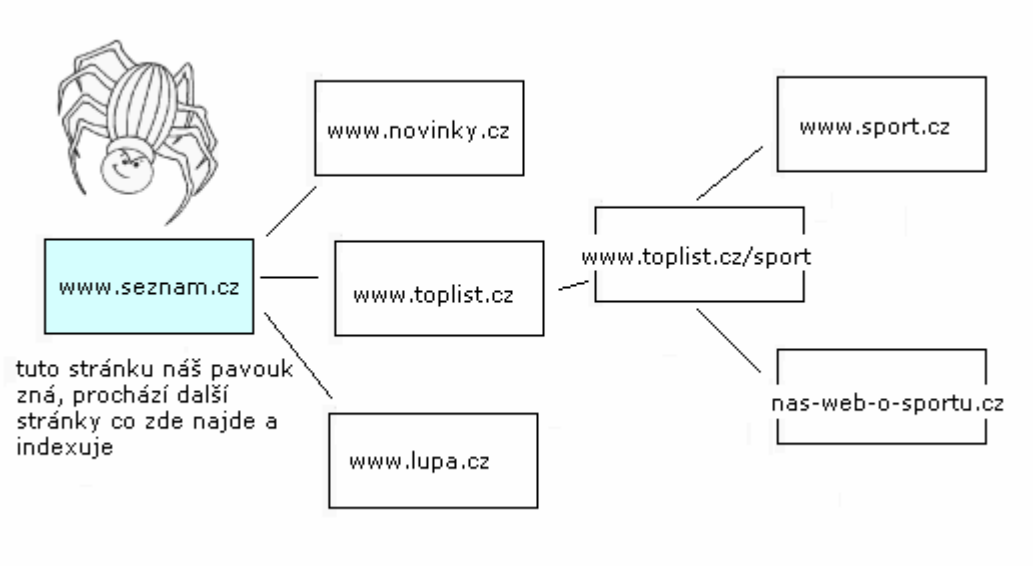
Hledání v katalogu seznam.cz nepatrí mezi optimalizace pro vyhledávace, pri zápisu stránky do katalogu na seznam.cz nebo centrum.cz se snažíme o tutéž optimalizaci. V popisu stránky, by mely být uvedeny všechny hlavní fráze a slova, která budeme potom optimalizovat i pro fulltextové vyhledávace.

Seznam fulltext	google	seznam katalog	centrum	google sk	atlas	ostatní	
54,6	20,0	12,8	6,4	2,9	0,8	2,5	100,0



4.2.2. Jak si nás vyhledávac najde – budování zpetných odkazu

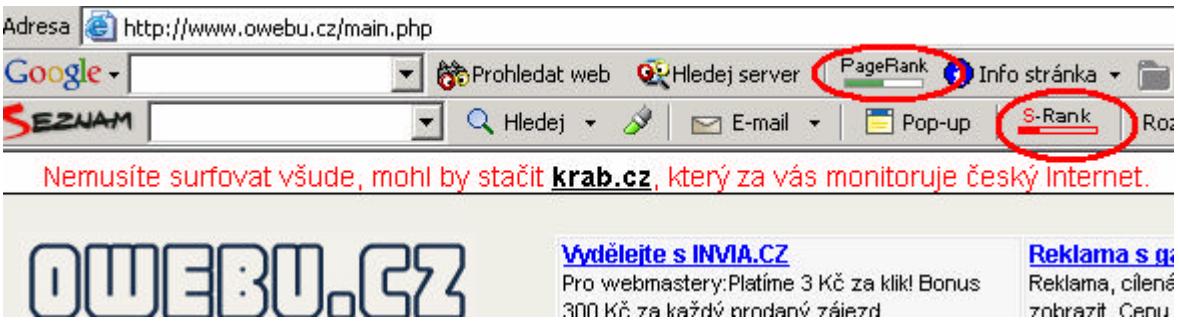
Než vubec dojde k prvnímu návštívení stránek z vyhledávacu, musí si nás vyhledávací služby zaradit do svých katalogu. Vyhledávací pavouk tedy musí objevit naši stránku. Musíme se zaradit na nekterou ze stránek, kterou již zná. Což není až takový problém, protože pokud se uložíme do nekterého z vetších katalogu, nebo udeláme príspevek do diskuze na známém webu, zajisté dosáhneme úspechu.



Další možností je zapsání www stránek do poradníku k indexaci stránek. Toto však vetšinou není duležité. A pokud nemáme urcitý RANK (viz dále), je to zbytecné.

4.2.3. Pagerank, s-rank

Pred samotnou strukturou textu na webu je rovnež duležité mít pagerank (pro vyhledávání na google.com) nebo s-rank (pro vyhledávání na seznam.cz). Predstavme si dve stránky se stejne kvalitním obsahem prezentace, potom by o prednostním zarazení pri vyhledávání mel rozhodovat i další faktor, a tím je zminovaný pagerank, s-rank (dále jen rank). Tento rank zajištuje urcitou verohodnost stránky.



Hodnotu ranku lze zjistit napr. pomocí prídavných lišt webového prohlížece.

Hodnota ranku je od nuly do jedné, pro uživatele je potom prezentován na stupnici 0 - 10. Každá stránka predává ostatním stránkám urcitý rank, to kolik predá závisí na tom, kolik odkazu obsahuje. Cím více má stránka odkazu, tím predá jednotlivým stránkám menší rank. Sama stránka však o svuj rank neprichází. Z predchozích slov plyne, že zvýšení vlastního ranku závisí na poctu webu s vysokým rankem, na kterých bude náš odkaz umísten.

Duležité je si uvedomit, že každá stránka má svuj rank, nikoliv jedna doména. (tedy frantaramus.cz/novinky a frantaramus.cz/politika mohou mít ruzný rank).

Rank není nejaké perpetum mobile, musela tedy zajisté existovat nejaká stránka, spíše skupina stránek, které mely page rank predem urcený. Z nultého ranku, by se asi težko dále generovaly kladné hodnoty.

Výpocet pagerank a s-rank není verejný, kdyby byl verejnosti prístupný, webmasteri by toho využili a upravili by své stránky presne podle tohoto algoritmu.

Další info o
s-rank http://fulltext.seznam.cz/url.py/infoScreen
pagerank http://www.google.com/technology/

4.2.4. To nejduležitejší, obsahová struktura stránky

Ze všeho nejduležitejší je získat klienty, které ocekáváme, tím, že je prilákáme na daná klícová slova a fráze. Optimalizace pro vyhledávace je velice složitá disciplína, je ztížena i tím, že vyhledávace velmi casto mení své algoritmy. Jé témer nemožné dostat se na první pozice vyhledávacu pri použití nekterého z nejvyhledávanejších slov.

Vzorove vezmeme seznam.cz (http://www.seznam.cz/topwords.html) a vidíme, že mezi nejcasteji vyhledávané destinace patrí Chorvatsko, Brno, Praha. Necháme-li vyhledat Chorvatsko, prvním odkazem je doména, která se tak jmenuje, následují stránky, které mají vysoký s-rank. Na druhou stranu, pokud naše firma pusobí v Ostrave a soucasne podniká napr. v oblasti pedikury, lze ocekávat, že pravdepodobne nebude mnoho takových konkurencních webových stránek a pokud ano, zajisté ne tak dobre optimalizovaných pro vyhledávací služby.

Optimalizace pro vyhledavací služby z pohledu on-page faktoru (toho co jsme schopni ovlivit), není žádná veda. Nejduležitejším prvkem je psát ctivý a **prirozený obsah** webových stránek, který bude obsahovat informace, na základe kterých chceme klienty získat a které klienti ocekávají. Klícová slova a fráze se snažíme psát do nadpisu, titulku a v lepším prípade i do url stránky.

Title - titulek

Nejduležitejší prvek pro vyhledávace je titulek stránky. Do title bychom meli umístit nejakou frázi, nebo nekolik slov odpovídajících celkovému zamerení stránky. Hustota slov v titulku je však omezena. V titulku muže být napr. **pedikúra, Ostrava Poruba .**

Nadpisy – H1,H2

Je velice duležité mít dostatek nadpisu, v techto nadpisech by mela být klícová slova nebo fráze, ne však na úkor obsahu webu. Osobne považuji nadpisy H3 a nižší za neduležité pro vyhledávace.

URL – stránky

Domnívám se, že seznam.cz dává textu v url stránky mnohem vetší význam, než napríklad google.com. Hezká url jsou pro návštevníka stránek více

zapamatovatelná. Predstavme si www.pedikura-ostrava.cz/kontakt a nebo stránku www.blazenanovakova/?page=1&info , v první adrese máme klícová slova a zároven je i zapamatovatelnejší pro uživatele stránek.

ALT – popisy obrázku

Jistou merou jsou prínosem i popisy obrázku <img src="adresa_obrazku" alt="popisek"> . Minimálne pri nacítání stránky, uživatel nejdríve uvidí popisek obrázku a poté až samostatný obrázek. Používá-li nekdo webový prohlížec s vypnutým zobrazováním obrázku, uvidí alespon titulek.

META tagy

Kdysi byly meta-tagy velice duležitým elementem, nyní se stávají nepodstatným. Bylo velice jednoduché napsat do meta-tagu klícová slova a fráze. Ale není na škodu je zmínit:

<META content="Naše pedikúra v porube vám pekne ostríhá nehty" name=description>
<META content="pedikura, ostrava poruba " name=keywords>

První meta tag udává popis stránky, který nekteré vyhledávace zobrazují u stránky, pokud ji vyhledají.

Druhý tag jsou klícová slova.

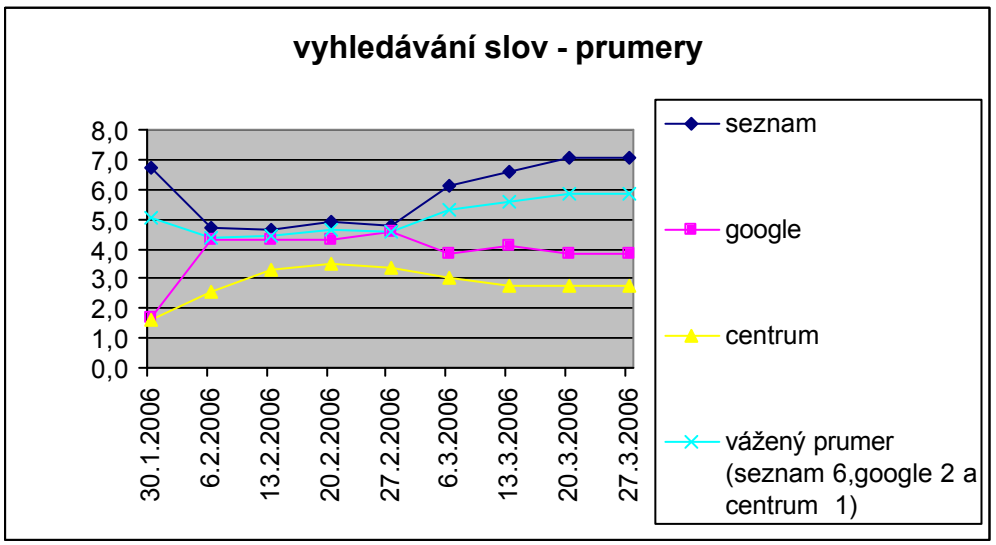
4.3. Konkrétní optimalizace internetové služby

Nejprve jsem si zvolil slova, slovní spojení, díky kterým chci, aby nás klienti našli. Následne jsem v týdenních intervalech provádel vyhledávání na nejduležitejších vyhledávacích : seznam.cz, google.com a centrum.cz a zapisoval data do tabulky. (viz príloha v xls)

Poté jsem se rozhodl pro bodování 0 až 10, dle toho, jaký výsledek nám dané vyhledávání poskytlo. Systém bodování jsem stanovil takto:

body	poradí
10	1
7	2
5	3
4	4,5,6
3	7,8,9,10
2	11 až 20
1	21 až 100
0	nenalezeno

Rovnež pro výpocet celkového váženého prumeru jsem postupoval podle využívání vyhledávacu, a váhy jsem stanovil pro seznam.cz na 6, google.com na 2 a centrum.cz na 1.



Po urcité dobe, ve které došlo ke zhoršení výsledku u nekterých slov a slovních spojení, jsem tento jev prisuzoval neaktualizaci stránek, kdyby stránky byly pravidelne meneny, zajisté by výsledky byly ješte lepší. Tento nedostatek se dá vyrešit napríklad vložením diskuze.

5. (Marketing založený na vyhledávacích)

5.1. Teorie

Zkratka SEM znamená **Search Engine Marketing**, v prekladu toto znamená **marketing ve vyhledávacích** (search engine = angl. vyhledávací stroj). Vetšinou se jedná o textovou reklamu. Oproti SEO je SEM službou placenou, jedná se o výhodný nákup pozic ve vyhledávacích.

Zamerení na cílený prodej

Oproti jiným reklamním kampaním, napríklad **banerovým,** nebo kampaníí v rádiích ci televizi, se SEO zameruje na klienty hledající urcitý typ produktu ci služeb. U obecnejších reklamních kampaní, nás nutí k urcité akci a nakonec si koupíme i to, co jsme nechteli. Toto u SEM není možné.

5.2. Konkrétní rešení pro náš e-shop

- je treba nakoupit odkazy na klícová slova, cena za klícové slovo je ruzná, v závislosti na atraktivite slova, vetšinou se pohybuje kolem 5 – 7 korun/proklik.

- Rozmyslíme si, jaká klícová slova a spojení použijeme, napríklad první cást bude zamerena na lécbu akné, navrhneme tedy slova a slovní spojení:
    - Lécba akné
    - Akné
    - Prípravky proti akné

Slova a slovní spojení, volíme tak, že premýšlíme co bychom sami zadali, kdybychom danou vec vyhledávali.

- Vytvoríme reklamní kampan.

- Nyní mužeme zjistit, kolik lidísi daný výrobek objedná za použití reklamy (napríklad vložením malého dotazníku u objednávky, s dotazem, jak naše stránky našli).

- Napríklad za 100 prokliku zaplatíme 500Kc, a objednávku uciní 10 lidí, máme tedy náklady na klienta 50Kc, i když prvotní cistý zisk na klienta bude nižší, máme šanci,že postupem casu, tento klient, který nás stál hodne penez na reklame, prijde sám.

Kampan doporucuji založit na

- https://adwords.google.com/
- http://www.etarget.cz/
- www.adfox.cz
- www.bbkontext.cz

Ukázka z nastavení kampane bbkontext, kterou jsem volil pro webhostingovou firmu banan.cz.

Jako klícová slova jsem zvolil: php, mysql, internet, doména, webhosting. Je možné zadávat presná slovní spojení jako napríklad „kvalitní webhosting“, nebo volná slovní spojení „php,webhosting“. Vetšina systému poskytuje i dobrou nápovedu.

Reklamy se poté budou zobrazovat na cizích webstránkách. První ukázkou je reklama na centrum.cz s využitím reklamního nástroje adfox:

Druhou ukázkou je adwords od google.com:

V nekterých prípadech se podarí zakoupit i výhodné pozice v katalogu **seznam.cz** . Seznam.cz nabízí vetšinou sponzorované odkazy v pravém rohu, garance první stránky, a klasicky sponzorované pozice pri vyhledávaní.

Pravdepodobne se již nevyplatí investovat do reklamy na centrum.cz (využíváme-li systém adfox, tak tam již reklamu máme) a na atlas.cz. Ceny bývají podobné jako u seznam.cz, a reklamy získáme méne.

Po registraci do katalogu jako je seznam.cz nebo centrum.cz mužeme ocekávat, že nám brzy bude volat jejich obchodník a presvedcovat nás o výhodnosti koupe reklamy na jejích katalogu. Z mé vlastní zkušenosti tak ucinili centrum.cz a seznam.cz. Pokud budeme dobrými obchodníky a budeme znalostmi o možnostech reklamy a poprípade i SEO prevyšovat obchodníka, je možné že nám brzy zavolá a nabídne velice výhodnou cenu.

Jako další zpusob reklamy mužeme použít banerovou reklamu, poprípade jinou alternativní reklamu, nebo prímo kontaktovat daný server s žádostí, že máme zájem o reklamu. Ale tento zpusob již nepatrí do kategorie SEO ani SEM.

POUŽITÁ LITERATURA:

Úvod do databází : Ing. Zdenka Telnarová, Ph.D.

www.mysql.com
www.apache.org
www.php.cz

www.root.cz
www.abclinuxu.cz
www.interval.cz
www.lupa.cz

PRÍLOHY:

CD se zdrojovými kódy eshopu