

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Diplomová práce

**Datové sklady**

Vypracoval: Bc. Ondřej Matuszczyk

Vedoucí diplomové práce: Doc. Ing. Vojtěch Merunka, Ph.D

2010©

## Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně za použití uvedených zdrojů, svých poznatků a konzultací s vedoucím práce.

V Praze, dne 3. 3. 2010

.....

Ondřej Matuszczyk

## **Poděkování**

Na tomto místě bych rád poděkoval vedoucímu mé diplomové práce váženému panu Doc. Ing. Vojtěchu Merunkovi, Ph.D. za jeho ochotu, trpělivost, cenné připomínky a čas, který mi věnoval.

Dále Danielu Markvartovi, jehož společnost sinne s.r.o. se stala základem této práce a Vojtěchu Kotoučovi, který byl jedním ze spoluvůrců prvního informačního systému FANzone.cz.

*Datové sklady*

*Data warehouse*

## Souhrn

Diplomová práce se zabývá problematikou databází a jejich přechodu od relačních k databázím s objektovou orientací.

Teoretická část poskytuje obecné informace o jednotlivých typech databáze a informace důležité pro jejich tvorbu. Představuje výběr z teoretické základny, na které byla vystavěna tato práce.

Praktická část se zabývá popisem evoluce informačního systému firmy, stručně popisuje analýzy stavů před jednotlivými změnami a na svém konci se pokusí o návrh datového skladu tak, aby vyhovoval předpokládaným budoucím nárokům.

## Klíčová slova

Relační databáze, Objektově orientovaný přístup, Dolování dat, Datový sklad.

## Summary

This graduation thesis is trying to get a sight on issues of database evolution.

In theoretical part it provides general informations about particular database types and rules needed to its creation. It presents theoretical base, basements of this document.

The second part is handling with evolution of information system, shortly summarizes analyses states afore transmutations and at its end is trying to propound the data warehouse so that complies with further requirements

## Keywords

Relational database, Object-oriented approach, Data mining, Data warehouse

## Obsah

1	Úvod .....	11
2	Cíl a metodika práce .....	13
I.	LITERÁRNÍ REŠERŠE .....	14
3	Relační databáze .....	15
3.1	Relace .....	15
3.2	Relační datový model .....	15
3.3	Relační databáze .....	16
3.4	SŘBD .....	17
3.5	SQL .....	18
3.6	MySQL .....	19
3.7	Datová normalizace .....	20
4	Objektově orientovaný přístup .....	21
5	Spravování a analyzování většího množství dat .....	23
5.1	Multidimenzionální databáze .....	23
5.2	Multidimenzionální databázový model .....	23
5.3	Datové sklady .....	24
5.3.1	Integrovanost .....	24
5.3.2	Časová variabilita a neměnnost .....	25
5.3.3	Návrh a koncepce datového skladu .....	25
5.3.3.1	Hardware .....	25
5.3.3.2	Software .....	26
5.3.4	Budování datového skladu .....	26
5.3.4.1	Metoda velkého třesku .....	26
5.3.4.2	Evoluční metoda – metoda přírůstková .....	27
5.3.5	Příprava údajů pro datové sklady – ETL .....	27
5.3.5.1	Extraction – extrakce .....	28
5.3.5.2	Transformation - transformace .....	28
5.3.5.3	Loading – vkládání .....	28
5.4	Datové trhy .....	29
5.5	Analýza OLAP .....	29
6	Ostatní použité pojmy .....	30

II. Projekt.....	32
7 Zkoumaný objekt a jeho počítačící vývoj.....	33
7.1 Představení subjektu.....	33
7.2 Přeclod ze skladových karet .....	34
7.3 Internetový obclod – počítačící řešení .....	34
7.3.1 Struktura databáze obclodu.....	35
7.3.1.1 Zákazník .....	35
7.3.1.2 Zboží.....	36
7.3.1.3 Objednávka.....	36
7.3.2 Přímý prodej .....	37
7.3.3 Skladový systém .....	37
7.3.4 Analýza stavu.....	37
7.3.4.1 Výhody dosavadního systému .....	37
7.3.4.2 Nevýhody dosavadního systému .....	38
7.3.4.3 Shrnutí stavu.....	39
7.4 Nová řešení pro zvýšení funkčnosti.....	39
8 Skladovnícký systém – proces změny .....	41
8.1 Výběr software.....	41
8.1.1 Již vytvořené skladovnícké systémy.....	41
8.1.2 Programování systému na míru .....	42
8.2 Popis vybrané varianty – ekonomický systém POHODA .....	42
8.3 Kódy produktů.....	44
9 Internetový obclod.....	46
9.1 Nové nároky na systém.....	46
9.2 Proces výběru.....	47
9.3 osCommerce .....	47
10 Zhodnocení stavu a návrh vylepšení .....	50
10.1 Zhodnocení aktuálního stavu .....	50
10.1.1 Požadavky a jejich naplnění .....	50
10.1.1.1 Schopnost obsloužit větší množství dat a více provozoven .....	51
10.1.1.2 Zlepšení přehledu o zásobách a přímého prodeje, zavedení registračních pokladen.....	51
10.1.1.3 Centralizace informačního systému, především skladovnictví .....	52
10.1.1.4 Restrukturalizace a zefektivnění internetového obclodu.....	53



10.1.1.5	Shrnutí .....	54
11	Návrh podpory analýzy dat .....	55
11.1	Struktura ukládaných dat .....	55
11.2	Návrh datového skladu .....	56
12	Závěr.....	59
13	Seznam použité literatury .....	60

## Seznam obrázků

obr. 1	- schéma činnosti SŘBD .....	18
obr. 2	- ER diagram původní databáze .....	35
obr. 3	- ER Diagram navržené databáze .....	56
obr. 4	- model datového skladu .....	58

## Seznam tabulek

tab. 1	- fakta.....	56
tab. 2	- dimenze .....	57

# 1 ÚVOD

Nejcennější komoditou současnosti je informace. Informace, která znamená náskok před konkurencí. Informace, která znamená náskok před světem.

Může to být příznivá informace, znamenající zisk, neméně důležité jsou však i informace o kritických stavech či nedostatcích. Ten, kdo chce jakoukoliv složitější činnost, která se skládá z více vstupů a výstupů, udržet v chodu a samu činnost co nejvíce optimalizovat, musí o této činnosti shromáždit maximum informací a veškeré získatelné informace analyzovat. Takovýchto informací se však dá získat nepřeberné množství, musejí tedy být extrahovány ze systému za nějakým účelem a na základě takového předem stanoveného cíle filtrovány a ukládány. Bezúčelné získávání dat by jen těžko vedlo k použitelným výsledkům.

K tomu, aby mohla být provedena užitečná analýza datového souboru, je potřeba, aby takovýto soubor byl co nejvíce rozsáhlý. Při malém rozsahu dat by mohlo dojít k silnému zkreslení, způsobeném vychylujícími se ojedinělými extrémy. A tyto data je také potřeba někde schraňovat a ukládat je tak, aby byla co nejjednodušeji použitelná. Informace ve formě dat jsou shromažďována od nepaměti. Z počátku na zdech jeskyní, hliněných tabulkách, papyru, poté v kartotékách (které se z dost praktických důvodů) používají na důležité věci do dnes. V padesátých letech 20 století však začaly být pomalu vytlačovány za pomoci vynálezu složitějšího počítačů, které se po několika desítkách let evoluce stalo modlou pro většinu světa. Řeč je pochopitelně o osobním počítači.

S tímto vynálezem začal být problém jak uložit data velice aktuální. Už nestačilo je jen zapisovat, vkládat do kartoték a poté při hledání rozložit více papírů vedle sebe.

Zásadním průlomem v tomto směru se ukázal E.F.Codd, jehož pohled na databázi jako na provázanou tabulku určil trend budoucího vývoje. Relační databáze, postavená na jeho pravidlech je efektivní cestou jak uložit velké množství dat.

Avšak pro práci analytika takováto databáze nestačí. Pro analyzování dat je potřeba znát jejich vývoj, stejně jako aktuální statickou hodnotu, kterou uchovává relační databáze, nezbytným je také daleko větší důraz na objekt. Objektové orientaci musejí ustoupit i požadavky na co nejmenší prostorovou náročnost a co nejmenší násobnost dat, o které se snaží databáze relační.

Vklíníme-li tedy nashromážděná data do struktury objektové orientované databáze a zajistíme-li dostatečnou datovou základnu, která bude stále aktualizována o další a další nová data a přidáme k tomu vhodný softwarový produkt, který dokáže provádět analýzy, získáme pomůcku k rozhodování, vhodné pro všechny řídicí pozice současné civilizace.

## 2 CÍL A METODIKA PRÁCE

Cílem této práce je pomocí praktického příkladu ukázat proces tvorby databázového a informačního systému v závislosti na požadavcích a možnostech zadavatele. Tento proces přesvědčivě znázorňuje vývoj firmy sinne s.r.o.. Autor této práce je jejím zaměstnancem na řídicí pozici, dokáže tedy popsat i zhodnotit celý její vývoj. Na základě zhodnocení poté navrhne řešení pro analytické zpracování dat a tvorbu datového skladu.

V první části se práce popisuje teoretickou základnu, na které byla vystavěna. Jsou zde zmíněny všechny důležité údaje a fakta, která se promítala do tvorby.

V části druhé – samotném projektu by práce ráda nabídla jako ukázky důvodů k proměnám informačních systémů stručné analýzy, které byly během vývoje vytvořeny. Pojednává o navržení jednoduché relační databáze, která se stala základem pro internetový obchod, rozepíše se o odlaďování internetového rozhraní e-shopu a optimalizování ekonomického systému.

Dále si projekt vzal za úkol na základě prognózovaného vývoje společnosti navrhnout systém, který by podporoval uložení a následné analyzování velkého množství dat.

V závěru této práce se práce pokusí celou tuto proměnu co nejobektivněji zhodnotit.

# **I. LITERÁRNÍ REŠERŠE**

## 3 RELAČNÍ DATABÁZE

### 3.1 Relace

Wikipedia<sup>[1]</sup> nám říká: „*Jako relaci nebo n-ární relaci nazveme v matematice libovolný vztah mezi skupinou prvků jedné nebo více množin*“

### 3.2 Relační datový model

*„Datovým modelem se označují vlastnosti vazeb mezi údaji v databázi a způsob organizace jejich struktury, který se nám prezentuje na logické úrovni, tedy na rozhraní SŘBD. Výběr datového modelu podstatně ovlivňuje vlastnosti dotazovacího jazyka a především způsob konstrukce jednotlivých dotazů. Datový model představuje definici k uložení a práci s informacemi v paměti počítače.“*

Takto shrnuje pojem relační databáze Dr. Vostrovský<sup>[2]</sup>

Relační datový model tedy popisuje databázi jako množinu dat, která jsou logicky a strukturovaně uspořádána. Data jsou elementárního charakteru a jsou propojena funkčními vazbami. Takovéto uspořádání dat nám umožňuje s nimi dále pracovat. RDM nám umožňuje určit pouze data, která bude databáze obsahovat a výstupy, které potřebujeme získat, vše ostatní jako popisy datových

struktur i procedury pro výstupy vyřeší databázový software. Jde tedy o jednoznačnou metodu pro přesné určení dat a dotazů nad nimi.

Základním předpokladem relačního datového modelu je, že všechna data jsou reprezentována jako matematické  $n$ -ární relace, které jsou podmnožinou kartézského součinu. V matematickém modelu je rozhodování o takovýchto datech uskutečňováno v predikátorové logice, nabývajících dvou hodnot – true a false.

Data jsou zpracovávána na základě relačního kalkulu a relační algebry. Relační datový model tedy umožňuje tvůrci databáze vytvořit konzistentní logickou reprezentaci informací. Konzistence je dosahováno zahrnutím deklarovaných omezení do konstrukce databáze.

### **3.3 Relační databáze**

Duchovním otcem relační databáze byl Edgar F. Codd, který tento termín mezi roky 1969 a 1970 poprvé definoval.

Relační databázi si můžeme představit jako tabulku, obsahující všechna data, jednotlivé záznamy jsou uloženy jako řádky této tabulky, sloupce jsou pak položky těchto záznamů. Každá položka záznamu (potažmo data ve všech sloupcích) by měla být jednorozměrná a dále nedělitelná, aby byla práce s nimi co nejjednodušší.

Ve svých 10ti pravidlech relačních databází shrnul E.F.Codd základní přístup k tvoření relační databáze. Je to souhrn podmínek, které by měly být dodrženy, aby relační databáze byla maximálně efektivní. Tato pravidla, která vznikala

v průběhu 70. let jsou z pohledu relačních databází nadčasová a jejich platnost trvá do dnes. Stavíme-li na nich a dodržujeme-li je, měla by být naše databáze z funkčního hlediska optimální, důležité je pak pouze správně databázi navrhnout dle zadaných požadavků.

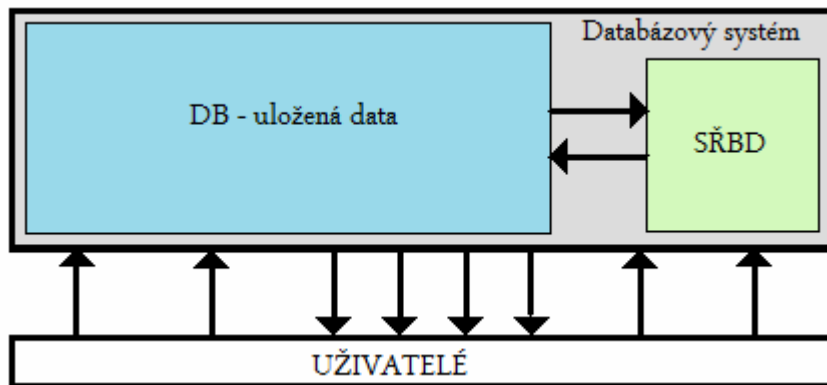
Data obsažená v relacích můžeme rozlišit na dva základní typy. Jsou to data statická a dynamická. Statická data, jak již jejich název napovídá, jsou data neměnná, tedy použijeme-li ně stejný dotaz, výsledek je také stejný. Dynamická data se v čase mění, tudíž výstupy se při opakovaném použití stejné metody mohou lišit.

### **3.4 SŘBD**

Systém řízení báze dat je softwarové zajištění komunikace mezi aplikačními programy a uloženými daty. Je velice důležité, aby jakýkoliv běžný uživatel nebyl připuštěn přímo k uloženým datům.

Není zde řeč o zabezpečení, avšak o bezpečnosti uložení dat. Velké množství havárií nejrůznějších aplikací a operačních systémů je způsobeno obávanou věcí mezi klávesnicí a židlí, jíž se říká běžný uživatel. I u administrátorů databází by přístup přímo k datům měl být co nejopatrnější. Data jsou v zájmu optimalizace databáze uspořádána ve vzájemných závislostech a každé porušení takovéto závislosti by mohlo vést k poruše systému ve velkém měřítku.





obr. 1 - schéma činnosti SŘBD

### 3.5 SQL

SQL je databázový programovací jazyk, jehož počátky datujeme do 70. let 20. století. Byl navržen pro spravování dat v relačních databázích. Spravováním dat myslíme především tvorbu struktury databáze a její obsluhu.

Jazyk je založen na tom, že jeho příkazy specifikují pouze co se má provést, nijak nezohledňují to, jak se toho dosáhne. O vše ostatní se stará relační databázová platforma.

Standardní příkazy SQL lze dle dr. Vostrovského<sup>[3]</sup> rozčlenit do čtyř následujících skupin

DDL (Data definition language)

Příkazy pro definici dat představují množinu příkazů pro vytváření struktury databáze

DML (Data manipulation language)

Příkazy pro manipulaci s daty tvoří jádro jazyka SQL a umožňují vyjádřit v jaké podobě je možno data z databáze získat nebo jakým způsobem existující data upravit

DCL (Data control language)

Příkazy pro řízení dat umožňují rozsáhlým způsobem zpřístupňovat data jednotlivým uživatelům a v případě práce s rozsáhlými databázovými systémy členit činnosti do logických celků – transakcí.

TCC (Translation Control Commands)

### **3.6 MySQL**

Tento multiplatformní databázový systém s dvojitým licencováním stvořila Švédská firma MySQL AB. Jedná se o systém pro spravování relačních databází z prostředí webové aplikace, díky čemuž je velice jednoduše přenositelný. Tento systém si potrpí na jednoduchou administraci, přehledné dotazování a je velice pohodlnou správou. O jeho oblíbenosti svědčí to, že je ho již více než 11 milionů<sup>[4]</sup> instalací.

Za cenu co nejvyšší rychlosti je systém MySQL poněkud zjednodušen. Některé funkce byly doplňovány až v posledních letech (např. pohledy, trigger, ...) Mnohdy se MySQL používá v kombinaci LAMB (Linux, Apache HTTP server, MySQL a PHP) jako softwarový základ webového serveru.

### 3.7 Datová normalizace

Pro optimalizaci relační databáze a pro zamezení výskytu redundantních dat se po navržení databáze provádí normalizace dat. Provádí se na její struktuře tak, aby se způsob rozložení relací co nejvíce optimalizoval a aby databáze řekněme zeštíhlela (zjednodušujeme stávající relace). Provádí se v několika fázích, které na sebe navazují. Ne každá databáze musí být nutně optimalizována na nejvyšší mez (normu), je to však za cenu její menší efektivity.

Cíle datové normalizace dle dr. Vostrovského<sup>[5]</sup> :

- umožnit reprezentaci každé relace v DB
- získat účinné algoritmy vyhledávání založené na jednodušší množině relačních operací
- redukovat potřebu restrukturalizace při zavedení nového typu dat
- zajistit neutrálnost souhrnu relací k četnosti dotazů vykazující tendenci měnit se v čase

## 4 OBJEKTIVĚ ORIENTOVANÝ PŘÍSTUP

Pro OOP, které se rozvíjelo již od 60. let 20. století byly stěžejní týmy Learning research group a Systém Concepts Laboratory, které se zasloužili o jeho větší rozvoj v letech 70. Počítač budoucnosti z jejich snah však nevznikl, proto se OOP musí přizpůsobit směru vývoje počítačů. A tím přizpůsobením byl smíšený přístup, který díky malé ztrátě abstrakce vůči OOP dosáhl jednoduššího a srozumitelnějšího vyjádření v hardwarově-softwarové řeči.

Jak je z názvu patrné, je základem tohoto přístupu objekt. Objektem může být víceméně jakákoliv součást našeho světa včetně nás samých. Objekty jsou vzájemně propojeny nejrůznějšími vazbami a poskytují dohromady pohled na námi zkoumaný, či zaznamenávaný kousek reálného světa. Je očividné, že při modelování se musí jednat pouze o část reálného světa.

Chtěli-li bychom zachytit kompletní strukturu světa, nejen že bychom často v přírodě i naší společnosti naráželi na vztahy a procedury, kterým jsme ještě nestačili porozumět, nebylo by to však možné ani z hlediska úložných kapacit a strojové náročnosti. Proto se vždy snažíme popisovat jen vybraný kousek světa a i ten dle svých potřeb zobecňujeme a zjednodušujeme.

Objekty jsou mezi sebou i s uživatelem schopny komunikovat prostřednictvím zpráv. Zprávu si můžeme představit jako požadavek, komunikaci pak jako zpracovávání požadavků. Další nezanedbatelnou vlastností objektů je, že v sobě mají obsaženy i metody, které se nad jejich daty dají provádět (například jako odpověď na některou ze zpráv uživatele). I modelování je poněkud odlišné od zaběhlých relačních modelů. Nesnažíme se realitu rozsekat do předem připravených mřížek, raději pro každý nám důležitý prvek reálného světa

vytvoříme objekt, který se bude snažit být jeho zjednodušeným a pro náš záměr účelným obrazem.

Hovoříme-li o protokolu objektu, máme na mysli všechny zprávy, které je možno danému objektu poslat, respektive množina všech zpráv, které je objekt schopen zpracovat.

Při seskupení více objektů o nich nehovoříme jako o množině, nýbrž (pro větší přesnost výrazu) jako o kolekci - pouze jedna z mnoha různých druhů kolekcí má vlastnost matematických množin.

Pro základní potřebu OOP vystačíme se třemi druhy kolekcí (jak uvádí doc. Merunka[6]):

Množina (anglicky „Set“) je kolekce, ve které prvky nemají žádné uspořádání. Přidává-li se do této kolekce prvek, který v kolekci již je, objeví se v zůstává v této kolekci jen jednou. Tento druh kolekcí jako jediný přesně odpovídá matematickému pojetí množin.

Ranec (anglicky „Bag“) je kolekce, ve které prvky také nemají žádné uspořádání. Přidává-li se ale do této kolekce prvek, který v kolekci již je, objeví se v této kolekci po takovém přidání více kopií tohoto prvku.

Seznam (anglicky „List“) je kolekce, které se chová jako ranec a navíc zachovává vnitřní uspořádání prvků, které obsahuje. Na rozdíl od rance i množiny proto v této kolekci můžeme najít první, druhý ... a poslední prvek.

## **5 SPRÁVOVÁNÍ A ANALYZOVÁNÍ VĚTŠÍHO MNOŽSTVÍ DAT**

### **5.1 Multidimenzionální databáze**

Multidimenzionální databáze nesplňuje normalizační formy, které byly uvedeny výše, avšak umožňuje mnohem rychlejší a komplexnější způsob pro orientaci ve velkém množství dat. Proto může za cenu větších nároků na hardware, především na diskovou kapacitu, být ideálním základnou pro komplexní analýzy, modelování a prognózování.

### **5.2 Multidimenzionální databázový model**

Vzhledem k tomu, že je potřeba uložit mnoho různých dat a lehce se mezi nimi orientovat, nevystačíme si už pouze s dvojrozměrnou tabulkou, kde sloupce označují atributy a řádky jednotlivé položky databáze. K analyzování potřebujeme mnohem více údajů, které se ve dvojrozměrné relaci nedají zaznamenat, proto nám přichází na pomoc matematika s mnohorozměrným prostorem. První tři rozměry si můžeme představit jako krychli a její souřadnice jako kombinace různých atributů.

U více rozměrů s tím však budeme mít pravděpodobně problém. Nebrání-li však tato skutečnost matematikům při jejich výpočtech, nemusíme se jí nechat omezovat ani my. A ačkoliv si krychli v třinácti-rozměrném prostoru jen těžko

představíme, analyticky jí zapsat ve formě mnoha agregovaných tabulek takový problém není.

### **5.3 Datové sklady**

„Datový sklad je podnikově strukturovaný depozitář objektově orientovaných, časově proměnlivých, historických dat použitých na získávání informací a podporu rozhodování. V datovém skladu jsou uložena atomická a sumární data.“  
[7]

Datový sklad je tedy úložiště určitých dat, nad kterými potřebujeme provádět nejrůznější analýzy. Tyto data musejí být, jak je uvedeno výše, uložena tak, aby byly odpovídající hodnoty ve stejných měrných jednotkách, aby byly uložené informace stejně strukturovány a aby všechna data byla platná a v takovém stavu, aby je nebylo nadále potřeba nijak upravovat (což posléze ani nelze, jinak by nám nemusela dvakrát provedená analýza vracet stejné výsledky).

#### **5.3.1 Integrovanost**

Pro datový sklad je nezbytné, aby všechny odpovídající si údaje byly ukládány ve shodných měrných jednotkách a aby byla jednotná i terminologie názvů. Proto se data před vstupem do datového skladu očišťují a upravují, jejich nekonzistentnost by ohrozila věrohodnost datového skladu.

### **5.3.2 Časová variabilita a neměnnost**

Data jsou schraňována v dávkách, vždy za pravidelný časový úsek. Takto uložená data nemohou již být dále upravována, jsou již jen zpracovávána.

### **5.3.3 Návrh a koncepce datového skladu**

Nežli se začne s budováním datového skladu, je potřeba mít přesnou představu o jeho účelu. Bez předem stanoveného cíle a strategie je jeho budování z největší pravděpodobnosti odsouzeno k neúspěchu. Jedině pokud je přesně známo, co od systému potřebujeme, dá se tento budovat tak, aby výsledek byl optimální.

Musíme si uvědomit, že jakákoliv implementace nových systémů se nikdy neobejde bez nákladů. V případě datového skladu jsou to především náklady na hardware a software. Druhořadými náklady jsou pak zaškolování personálu a údržba samotného systému.

#### **5.3.3.1 Hardware**

Při budování datového skladu musíme počítat s tím, že se budeme snažit uchovávat velké množství informací. A nejen to, budeme je posléze i analyzovat, čili s nimi pracovat. A tomuto faktoru musíme přizpůsobit jak kapacitu našich úložních prostor, tak výpočetní kapacitu strojů. Nesmíme také zapomínat, že datové sklady zpravidla nepoužívá jediný člověk, jsou sdíleným systémem a tudíž musí být tomuto faktu přizpůsobena i přenosová rychlost komunikace.



### **5.3.3.2 Software**

Nedílnou součástí nákladu na budování datového skladu je pořízení nástrojů na jejich vytváření a spravování. Nejedná se o jednoduchou záležitost, tudíž nechávat si dělat tento systém namíru pravděpodobně nepřípadá v úvahu. Musíme tedy sáhnout po již vytvořeném systému – v poslední době se stále více prosazuje trend integrace analytických služeb přímo do databázových serverů.

## **5.3.4 Budování datového skladu**

### **5.3.4.1 Metoda velkého třesku**

Tato velice nepraktická metoda najde své využití pouze při budování malého datového skladu, který bychom mohli spíše nazývat datovým trhem. Při budování velkého, finančně a časově náročného datového skladu je tato metoda nejen nepraktická, ale především neefektivní. Jedinou výhodou této metody je, že projekt může být celý zpracován ještě před samotnou realizací, ale vzhledem k časovým nárokům se může lehce stát, že postavíme datový sklad nejen pro aktuální dobu zastaralý, ale že již nebude odpovídat našim požadavkům.

#### **5.3.4.2 Evoluční metoda – metoda přírůstková**

Jedná se o metodu, kdy datový sklad není budován celý najednou, projekt je rozdělen do jednotlivých částí. Tyto etapy budování pochopitelně zapadají do finálního schématu datového skladu, umožňují však lépe vycházet vstříc požadavkům a pružněji reagují na veškeré změny nároků na systém, ke kterým může v průběhu projektu docházet.

#### **5.3.5 Příprava údajů pro datové sklady – ETL**

ETL je zkratkou - Extraction, Transformation, Loading – a vyjadřuje všechny tři důležité fáze přípravy dat. Jak již bylo zmíněno několikrát výše, pro datový sklad je důležitá konzistence dat. Ta však většinou nevzniká sama od sebe. Údaje, které nám budou sloužit jako datová základna, jsou většinou získávány z více různých zdrojů a styl jejich uložení se různí. K tomu, abychom data sjednotili tedy potřebujeme nějaký univerzální postup, abychom na jeho výstupu získaly data, které pak budeme moct lehce zpracovávat. Je jasné, že tento proces nebude nijak jednoduchý a že bude časově náročný, nicméně pokud bude správně proveden, jeho výsledek bude dostatečnou odměnou.

### **5.3.5.1 Extraction – extrakce**

V této fázi vybíráme data z různých zdrojů – převážně vzájemně nehomogenních. Data bývají uložena na různých platformách a v různých programech.

### **5.3.5.2 Transformation - transformace**

K efektivnímu používání datového skladu potřebujeme mít data co nejvíce kvalitní. Nekvalitní data negativně ovlivňují rozhodování a proto se jich musíme vyvarovat. Nejdříve je potřeba data „očistit“, zbavit všech nevhodných dat i dat, u kterých došlo při vkládání k selhání lidského faktoru. Samotnou transformací poté můžeme rozumět jako úpravě dat do jednotného stylu, čili vypořádat se např. s jejich nejednoznačností (např. v databázi osob může být pohlaví zapsáno jako M, Muž, Mužské, ...), duplicitou dat a chybějícími záznamy.

### **5.3.5.3 Loading – vkládání**

Jak již název naznačuje, jedná se o poslední etapu přípravy údajů – přenos upravených zdrojových dat do datového skladu.

## 5.4 Datové trhy

Nechceme – li budovat celý datový sklad najednou, můžeme zvolit cestu datového trhu. Jedná se vlastně o menší, samostatnou část datového skladu, dalo by se říct podmnožinu, která se dá posléze k datovému skladu připojit, ale je na něm nezávislá a dokáže sama pracovat jako jednoduchý datový sklad. Je možné tento postup samozřejmě i otočit a z datového skladu posléze oddělit jeho část, abychom napomohli lepší flexibilitě a udržitelnosti datového skladu.

## 5.5 Analýza OLAP

Pan Lacko ve své knize<sup>[7]</sup> tuto analýzu popisuje takto:

*OLAP je volně definovaný řád principů, které poskytují dimenzionální rámeček pro podporu rozhodování.*

Každou analýzu OLAP tvoří dva druhy dat – fakta a dimenze. *Fakta* označujeme data, která jsou numerickým záznamem, uloženým v databázi. Fakta mohou být také výsledkem výpočtu na základě jiných faktů. *Dimenzemi* nazýváme slovní vyjádření dat, jsou to vlastně data logicky nebo organizačně hierarchicky uspořádané. Jinými slovy vysvětlují všechny „jak“ a „proč“ pro transakce prvků.

## 6 OSTATNÍ POUŽITÉ POJMY

- *Eform*

Eform označuje dvojici technologií firmy Stormware. Pro tuto práci je důležitá pouze jedna z nich, a to obecně použitelná definice, která usnadňuje elektronický přenos obchodních dokladů, jinými slovy dokáže velice efektivně vytvářet doklady tak, aby se daly importovat do jednotlivých ekonomických systémů. Optimalizováno pochopitelně pro software Pohoda.

- *Log*

V průběhu textu bude použito toto slovo, které znamená funkci počítače, která uchovává historii sledovaných operací systému.

- *PHP*

Český překlad této zkratky je hypertextový preprocesor, jedná se o skriptovací programovací jazyk. Používá se především pro programování dynamických internetových stránek. Jak již bylo zmíněno výše, je obvykle používán v různých kombinacích s ostatními jazyky a programy. Jednou z nejoblíbenějších kombinací je LAMP – linux, apache, MySQL a PHP.

- *PhpMyAdmin*

PhpMyAdmin je velmi oblíbeným nástrojem pro správu MySQL databází s freeware licencí. O jeho oblíbenosti svědčí jeho časté nasazování u providerů webových služeb.

- EAN - European Article Number

Tato zkratka se týká čárových kódů. Čárovým kódem nazýváme standardizovaný prostředek, používaný k automatizovanému sběru dat. Nejpoužívanější varianta – EAN13 se používá k označování produktů po celém světě. Je lokalizován – dá se jeho pomocí zjistit zemi původu.

- PayMUZO

PayMUZO provozuje společnost Globalpayments i.c. Je to on-line platební brána, která umožňuje aplikovat bezpečnou variantu platby pomocí kreditní karty přes internet. Jejím základem je, že při on-line platbě je zákazník přeměrován přímo na stránky PayMUZO, kde provede transakci a poté se vrací do e-shopu. Portál okamžitě informuje, zda platba proběhla úspěšně. Pro internetové je to výborná cesta, jak nabízet možnost platby on-line bez nutnosti budování speciálního zabezpečení a programového aparátu.

- XML – extensible markup language

XML je značkovací jazyk, který se používá pro výměnu dat mezi aplikacemi

## **II. PROJEKT**

## **7 ZKOUMANÝ OBJEKT A JEHO POČÁTEČNÍ VÝVOJ**

### **7.1 Představení subjektu**

*FANzone.cz* je obchodní značkou firmy sinne s.r.o. a zabývá se sportovním merchandisingem – prodejem suvenýrům, poutajícím se k různým sportovním akcím a jednotlivým sportovním klubům.

V současné době je její pole působnosti rozděleno mezi výhradní zastoupení klubu Sk Slavia Praha fotbal, HC Slavia Praha, FK Mladá Boleslav a firmy GSMA a.s. Dále se účastní většiny akcí, pořádaných společností Česká Sportovní a.s.

Počátkem veškerého informačního systému této firmy leží uzavřený v jedné velké kartónové krabici – jsou to skladové karty, papírový model databáze, který vznikl dávno před tím, než E.F Codd určil relačním databázím základní podmínky. A bylo na nás, abychom s přibývajícím množstvím dat a operací s nimi postupně vymýšlely nové způsoby jejich efektivního zpracování.



## **7.2 Přechod ze skladových karet**

Při návrhu relační databáze se postupovalo na základě skladových karet. Na skladové kartě bylo pořadové číslo produktu, jeho název, výrobce a skladované zboží. V malé prodejně s několika kusy zboží to byl naprosto dostatečný a ekonomicky absolutně nenáročný systém. Upgrade hardware probíhal v papírnictví dokupováním dalších skladových karet a propisovacích tužek.

Firma se však začala rozrůstat a systém skladových karet přestal dostačovat. Bylo potřeba pružnějšího přístupu k datům zvýšení jejich dostupnosti. Ke kamennému obchodu se také přidal internetový obchod, bylo tedy potřeba systém poněkud přepracovat, aby mohl začít fungovat on-line.

Byl tedy vyvinut informační systém, který pro potřeby internetového obchodu spojil PHP a SQL a pro skladovnictví a evidenci tržeb využil MS Excel.

## **7.3 Internetový obchod – počáteční řešení**

Internetový obchod a jeho databáze byly spravovány prostřednictvím phpMyAdmin, což ale vyžaduje znalost příkazů SQL pro ovládání základních funkcí v databázi. I pro jednoduché úpravy v databázi produktů, nebo pro drobné úpravy nastavení internetového obchodu je potřeba programovat.

Pro obsluhu objednávek internetového obchodu byly vytvořeny nové relace, které ve formě faktury poskytovaly veškeré potřebné informace – fakturační

adresu, doručovací adresu, způsob platby, objednané zboží s cenami a celkovou cenu objednávky. U objednávky bylo možno měnit její stav – to vše s podporou v PHP naprogramovaném uživatelském prostředí.

### 7.3.1 Struktura databáze obchodu

Jelikož jde o databázi internetového obchodu, atributy jsou víceméně standardní a jak je patrné, jedná se o relativně jednoduchou databázi



obr. 2 - ER diagram původní databáze

#### 7.3.1.1 Zákazník

Zákazník je bezesporu nejdůležitějším objektem, který v každém internetovém obchodě vystupuje.

U zákazníka je na FANzone.cz evidováno jeho celé jméno, adresa a kontakty. Důležitým atributem je země, do které je zboží posíláno.

### **7.3.1.2 Zboží**

Krom názvu, ceny a DPH jsou důležitými aspekty zboží jeho váha a dostupnost.

Váha je nutná k výpočtu poštovného, dostupnost má informativní charakter a slouží jak zákazníkům, tak zaměstnancům. Jde o informaci, zdali je zboží skladem, nebo kdy je očekávána jeho další dodávka.

### **7.3.1.3 Objednávka**

Objednávka v námi zkoumané databázi propojuje databázi zákazníků s databází zboží.

Velmi důležitou informací u objednávky je, zda byla placena online při nákupu, bude zaplácena dobírkou, či zda se čeká na platbu při převodu z účtu na účet. Stav objednávky je nepostradatelný jak pro zaměstnance, tak pro zákazníky. Pro zaměstnance je velice důležitý – dle stavu poznají, zdali se objednávce nevěnuje již někdo jiný, nebo jestli již není vyřízena.

### **7.3.2 Přímý prodej**

Přímý prodej stále probíhal bez jakékoliv podpory výpočetní techniky. Paragony jsou vypisovány ručně, údaje o prodejkách v podobě tržeb jsou zapisovány do tabulky (MS Excel) a archivovány.

### **7.3.3 Skladový systém**

Podobně jako přímý prodej je i sklad evidován pouze ve formě tabulek v msExcel, ve kterém se provádějí veškeré nezbytné analýzy, které tabulkový kalkulátor podporuje.

### **7.3.4 Analýza stavu**

#### **7.3.4.1 Výhody dosavadního systému**

Nespornou výhodou tohoto řešení jsou velice nízké pořizovací náklady. Pro objem dat, které jsou v tomto případě používány, je to i optimálně rychlý systém. Dovoluje automaticky vytvářet doklady pro internetový obchod a poskytuje možnost administrace celého internetového obchodu. Je nenáročný na výpočetní

techniku a umožňuje tedy uchovávat základní informace, potřebné k vedení firmy, za velice minimální náklady.

#### **7.3.4.2 Nevýhody dosavadního systému**

Vezmeme-li v potaz výhodu možnosti administrace internetového obchodu, nevýhodou v tomto bodě je, že veškerá obsluha tohoto systému vyžaduje znalosti SQL dotazů, pro modifikaci funkcí a dat v internetovém obchodě je nutná znalost programovacího jazyka PHP.

Další nevýhodou, kterou je potřeba zmínit, jsou složité úpravy systému a celková nepružnost. Vzhledem k tomu, že firma má tendenci se rozrůstat, je tato nevýhoda poměrně výrazná. K jakékoliv změně je potřeba celý systém přeprogramovávat, má necentralizovanou architekturu a používá více systému uložení dat, kdy hrozí jejich redundance a tudíž mnoho potenciálních chyb.

Vrátíme-li se k obslužnosti internetového obchodu, dalším problémem je, že není schopen uchovávat logy, ani přidávat poznámky k jednotlivým objednávkám. To je z hlediska použitelnosti internetového obchodu velice důležitá věc. V praxi je velice těžké, pokud má sklad velké množství položek, udržovat dostatečné skladové zásoby. Pracujeme-li ještě k tomu s faktorem minimálních nákladů, je to skoro neřešitelná věc.

Je tedy nutné zákazníky informovat o nedostupnosti zboží, případně o alternativách, které jsme schopni nabídnout. Je nutné, aby veškerá takováto komunikace se zákazníkem byla evidována na nějakém centrálním místě. V tomto bodě je totiž redundance dat pro on-line obchod nejnebezpečnější –

vícenásobně odeslaná zásilka, neposlané zásilky, jelikož vytištěná objednávka se špatně zpracovala, atd.

### **7.3.4.3 Shrnutí stavu**

Tento systém byl ideální cestou, jak za minimální cenu vytvořit fungující informační systém. Systém takový, který zvládl obsloužit jak přímý prodej, tak internetový obchod. Řešení to nebylo určitě ideální (jak je vidět z řady nevýhod), ale v rámci mezí dokázal bezchybně fungovat 2 roky a dovolil tak firmě vyřídit více jak 5000 internetových objednávek. Firma se však po této době začala výrazně rozrůstat a pod svá křídla si vzala i další činnosti a otevřela další obchody. Na konci svého životního cyklu začal být tedy i tento systém nedostatečný.

## **7.4 Nová řešení pro zvýšení funkčnosti**

Jak již bylo zmíněno výše, začala se firma zvětšovat. Zvýšil se počet kamenných obchodů z jednoho na 3. Přibily také mobilní obchody, stánky, které jsou otevírány pouze v době konání různých akcí. Vytíženost internetového obchodu se také výrazně zvýšila – množství objednávek, chodících za den se vyrovná s množstvím, které bylo zpočátku přijímáno za měsíc. I množství zaměstnanců se zvýšilo, nebylo tedy jiné možnosti, než inovovat již nedostatečný systém.

V první řadě bylo potřeba sjednotit a pokud možno centralizovat skladovou evidenci, systém excelovských tabulek, sbíraných jednou měsíčně, již naprosto nedostačoval potřebám. Také způsob prodeje bylo potřeba zefektivnit, nemluvě o nutnosti používat od 1.1.2007 používat při maloobchodním prodeji registračních pokladen.

Změny, které nastaly, je možno rozdělit do dvou skupin, změny skladovnického systému a změny na poli internetového obchodu.

## **8 SKLADOVNICKÝ SYSTÉM – PROCES ZMĚNY**

System spravování dat prošel výraznou změnou. Je-li potřeba uchovávat velké množství dat a ještě s nimi dále pracovat, bylo by velice neefektivní, až nemožné používat stávající systém. Bylo potřeba nalézt řešení, které by zefektivnilo uchovávání dat i jejich přenosnost.

### **8.1 Výběr software**

Jak bylo zmíněno několikrát výše, systém excelovských tabulek bylo potřeba nahradit, respektive rozšířit efektivnějším skladovnickým systémem. Bylo vybíráno z několika možností, které nám poskytl český trh.

#### **8.1.1 Již vytvořené skladovnické systémy**

jedním z možných řešení je pořídit již vytvořený skladovnický systém. Ze systémů, které se vyskytují na našem trhu byly k porovnávání vybrány SW Pohoda a skladovnický systém MAtis. U těchto systémů je výhodou jednoznačný rozpočet. Je předem dáno, kolik nás bude takový program, včetně pro nás důležitých modulů stát. Na druhou stranu je to cesta ústupků a kompromisů, jelikož musíme náš informační systém vměstnat do předem vytvořené kostry. A i jednoduché rozšiřování a úpravy jsou možné pouze na základě dokupování doplňků, vydaných k tomuto systému.



### **8.1.2 Programování systému na míru.**

Na rozdíl od již vytvořených systémů nám programování na míru umožňuje dostat ve výsledném produktu pouze to, co opravdu potřebujeme a potřebovat budeme. Zadá – li se projekt profesionální firmě, máme jistotu, že software pro nás bude z provozní stránky optimalizován, bude kladen důraz na pro nás důležité faktory a procesy. Negativní stránkou programování na míru jsou ekonomická a časová náročnost. Oproti již vytvořeným systémům nemáme jistotu, že systém nepřekročí v průběhu svého vývoje rozpočet a co se časového hlediska týče, je tento způsob jednoznačně náročnější – je potřeba analýza stavu a zdrojů, čas na vytvoření samotné aplikace a čas na její implementaci a následnou optimalizaci. U již vytvořených systémů se tento čas omezuje pouze na implementaci a optimalizaci, ostatní již kupujeme jako funkční celek.

Programování na míru je pro malou firmu ekonomicky velice neefektivní, je vhodnější sáhnout po již hotovém produktu. Z výše popsaných systémů se pro potřeby společnosti jevil Pohoda software jako ideální – nebylo potřeba dokupovat další moduly, vystačíme si s variantou Standard, která obsahuje veškeré funkce, které bude firma potřebovat, včetně on-line pokladny a podpory propojení s internetovým obchodem.

## **8.2 Popis vybrané varianty – ekonomický systém POHODA**

Pohoda je velice flexibilní ekonomický systém, který firmě umožňuje vést skladovou evidenci, účetnictví i systém pokladen. Základní funkce v systému

jsou jednoduché k ovládní a celkově je systém dobře vybaven na omezování funkcí na provozní minimum.

Administrátor má pochopitelně práva k provádění veškerých operací, která program nabízí. Má také přístup k vcelku jednoduchému omezování práv ostatních uživatelů. V přehledném menu se dá přesně určit akce, které může daný uživatel provádět. Jak bylo psáno v rešerši, nejnebezpečnějším subjektem pro jakoukoliv databázi je „obyčejný uživatel“, proto je tato funkce velice důležitá.

Architektura programu je centralizovaná, aplikační základna spolu s polem dat je uloženo na serveru, ostatní jednotlivé stanice používají síťovou verzi aplikace, chovající se jako terminál – akci je možno zadat na jakékoliv stanici, samotný proces je prováděn na serveru.

Další nespornou výhodou je možnost rozdělení skladů. Na jednom serveru je možno vést evidenci více na sobě nezávislých skladů bez nutnosti dokupování další licence programu. Tato vlastnost je velice užitečná, vezmeme-li v potaz, že je potřeba od sebe oddělit např. evidenci centrálního skladu od skladu obchodu, když obé běží na stejném serveru.

S okolím komunikuje Pohoda pomocí tzv. eform. Jedná se o technologii tvorby dokladů pro skladovnictví (dodací listy, výdejky, příjemky, atp.) tak, aby byla byly jednoduše přenositelné. Takto vytvořené soubory ve formátu xml lze poté jednoduše odesílat, přijímat i importovat do ekonomického systému. Při obsluze více poboček a skladů je to neocenitelná pomůcka, která má praktické využití i při obsluze internetového obchodu.

### 8.3 Kódy produktů

Jelikož se sklad výrazně rozrostl, musel doznat výraznějšího systému rozdělení. Jednotlivé položky skladu byly do této chvíle evidovány na základě čtyřciferného kódu. Tento kód byl pouze pořadovým číslem produktu. Sortiment se však rozrostl. Když firma začínala, věnovala se pouze prodeji suvenýrů z hokejového Mistrovství Světa, posléze se rozrostla o suvenýry HC Slavia Praha, dále o suvenýry SK Slavia Praha fotbal, FK Mladá Boleslav, o suvenýry k akcím GSMA Freestyle Motocross a suvenýry k akcím České sportovní, a.s.

Tyto jednotlivé části sortimentu bylo potřeba od sebe odlišit tak, aby se ve skladu dalo efektivněji orientovat, aby na základě kódu produktu bylo patrné, do kterého sortimentu patří. Zároveň však bylo důležité zajistit unikátnost každého kódu.

Vznikl tedy nový systém evidence. Bylo zachováno tělo kódu, tedy pořadové číslo. Z důvodu předpokladu, že se sortiment bude stále rozvíjet, byl tento kód rozšířen o jednu cifru. O další litery jej rozšířilo o rozlišování na jednotlivé druhy sortimentu.

Příklady kódu:

HC12231, MB02621, U03931, UMB03539, ON03342, SK03531, 903531, GSMA03452

První až 4 litery (jedná-li se o písmena) reprezentují příslušnost k části sortimentu – pro větší možnosti kódů jsou realizovány i textovým řetězcem, který dává větší prostor kombinačním možnostem a lepšímu rozlišení. MB na začátku kódu produktu tedy znamená, že zboží je součástí sortimentu FK Mladá Boleslav, U vypovídá o tom, že patří do sortimentu Umbro. Dalších 5 cifer reprezentuje samotné pořadové číslo produktu.

Při tvorbě takového systému, především při jeho aplikaci je zásadní unikátnost každého kódu a to nejen celého, ale i jeho části, která vyjadřuje pořadové číslo - ne všechny technické prostředky dokáží pracovat s písmeny v kódu (jednodušší čtečky EAN kódů a pod.)

## **9 INTERNETOVÝ OBCHOD**

### **9.1 Nové nároky na systém**

U on-line obchodu, jehož návštěvnost se zvýšila geometrickou řadou, bylo potřeba zavést takový systém, který by pružně reagoval na změny struktury a umožňoval rychlé ovládání funkcí a dat v něm obsažených.

Přibyly požadavky na systém, které nebylo možno vyřešit bez přeorganizování celého systému a následné přeprogramování jeho větší části. Abychom udrželi krok s dobou, bylo nezbytné vyřešit on-line platby za objednávky. E-shop a jeho administrační prostředí postrádalo také možnost komunikovat se zákazníky a tuto komunikaci zaznamenávat. Stejně tak zaznamenávat jednotlivé změny stavů.

Tato evidence historie je velmi důležitá. Představme si vyřízenou objednávku – v systému je evidována pod stavem „odesláno a vyskladněno“. Představme si dotaz od nespokojeného zákazníka, který se dotazuje, kde je jeho objednávka, že stále nic neobdržel. Pokud není možné zobrazit historii změny stavu a připojovat poznámky k jednotlivým stavům, je dohledání takovéto kolize faktů velmi náročnou operací – úspěšnou pouze pokud dostatečně evidujeme průběh objednávek alespoň v papírové podobě. Pro běh internetového obchodu jsou však takovéto informace velice podstatné – tvoří cca 70% dotazů.

Bylo tedy potřeba hledat nějaké efektivnější řešení e-shopu tak, aby vyhovoval novým potřebám.

## 9.2 Proces výběru

Opět se dostáváme ke stejnému dilematu, jako u výběru ekonomického systému - nechat si obchod naprogramovat na míru, či využít nějaký z předprogramovaných a jen jej nakonfigurovat.

Je-li celý systém budován na míru, vystavujeme se riziku nesplnitelných nároků, nedodržení časových termínů, či překročení rozpočtu. Hotový produkt je tedy cesta jednodušší, bezpečnější, ne vždy však dostatečná. Jelikož v tomto případě není potřeba hledat složitá řešení, množství dat není závratné, ani nemáme nestandardní požadavky, byla zvolena cesta již hotového obchodu.

Při hledání optimální platformy, na které hodláme vystavět náš e-shop máme v dnešní době na českém trhu nepřeberné množství možností. V našem případě bylo přistoupeno na variantu, kterou firmě navrhla společnost ACE&S, se kterou byla navázána spolupráce za účelem programového zaštitění restrukturalizace systému. osCommerce se ukázala jako velice zdařilou variantou velmi dobře konfigurovatelného a programovatelného obchodu, který je ovládán jednoduše přes webové rozhraní.

## 9.3 osCommerce

Celý systém administrace je na základě požadavků rozdělen do mnoha částí. K odbavování objednávek však stačí několik základních agend – Zákazníci, Objednávky a Zboží.

Agenda *Zboží* nám dává možnost jednoduše přidávat produkty, včetně obrázků, popisů, i kategorií. Kategorie určují příslušnost k části sortimentu a i na jejich základě může být produkt v obchodě vyhledán. Lze také nastavit množství, které je skladem a poté nastavit podmínky, při kterých bude zákazník informován o zbývajícím počtu kusů předmětu, eventuálně o jeho dočasné nedostupnosti a datu jeho z dostupnění.

*Zákazníci* nám nabízejí pohodlně ovladatelnou databázi všech registrovaných uživatelů. Při registraci do e-shopu dovoluje ověřit příslušnost k fan klubu jednotlivých klubů ( tyto skupiny zákazníků jsou oproti ostatním zvýhodněny). Lze také např. omezit možnosti platby a znedostupnit pro zákazníka možnost objednat si zboží na dobírku – toto omezení je používáno v případě, že zákazník bez udání důvodu nevyzvedne svou doručenou objednávku, stejně jako na požádání majitele účtu zaslat nové heslo. Jednoduchá je i obsluha tzv. newsletteru, e-mailu, rozesílaného všem registrovaným uživatelům, pomocí kterého jsou informováni o novinkách na stránkách.

Nejpotřebnějších změn doznala část *Objednávky*. Oproti předchozímu systému je ovládání objednávek mnohem pružnější a efektivnější. Optimalizace tohoto systému sice nebyla úplně rychlým procesem - odlad'ovalo se za běhu, kdy se teprve ukázaly praktické nedostatky.

Jednou z nejdůležitějších změn je možnost sledovat agendu nazvanou *stav objednávky*, ve které je záznam všech akcí, které byly se záznamem prováděny. Tato transakční historie zaznamenává veškeré změny stavu objednávky, datum i čas těchto úprav i poznámky, které lze pro potřeby administrace objednávek vkládat. Velice užitečná funkce je možnost informovat zákazníky o jednotlivých

změnách pomocí automatizovaného rozesílání e-mailů. Nesmíme totiž zapomínat, že dotazy na stav objednávky tvoří 70% všech dotazů.

Je také možno upravovat jednotlivé objednávky pomocí přidávání a odebrání zboží. Krom přesné evidence odesílaného zboží je tato aktualizace objednávek nepostradatelnou součástí, bez které by komunikace se systémem Pohoda nemohla probíhat korektně.

Výměnu dat s Pohodou zajišťuje možnost exportu dat z internetového obchodu do elektronického dokladu, který je upraven tak, aby bylo možno jej pomocí funkce eform jednoduše importovat.

Problém s on-line bankovníctvím byl vyřešen díky Pay MUZO. Praxí jsme ověřili, že je to funkční, bezpečný a rychlý způsob přijímání on-line plateb kartou.

Díky dobré strukturovanosti je možno systém jednoduše upravovat a „nalepovat“ další a další modifikace, úpravy a funkce na základě provozních požadavků bez nutnosti celkové restrukturalizace při jejich aplikování.



## **10 ZHODNOCENÍ STAVU A NÁVRH VYLEPŠENÍ**

### **10.1 Zhodnocení aktuálního stavu**

Maximální efektivitu se nedá docílit bez dostatečného nasazení. Nebudou-li lidské zdroje pracovat s maximálním úsilím, sebelepší systém nedosáhne výsledků, kterých by mohl. To však není tématem této práce, zhodnotíme tedy pouze kvalitu aktuálního stavu informačního systému.

Současný informační systém již 2 roky prověřuje ostrý provoz. V průběhu této doby bylo potřeba reagovat na celou řadu situací, se kterými nebylo původně počítáno. Byly vyřízeny desítky tisíc objednávek, množství položek skladu se vyšplhalo skoro na deset tisíc, zvýšilo se množství poboček i akcí, pro které bylo potřeba vytvořit a přenést skladové doklady a vést skladovou evidenci.

Na základě praktického používání systému můžeme vyvodit mnoho závěrů a tudíž implementaci celkem objektivně zhodnotit.

#### **10.1.1 Požadavky a jejich naplnění**

Pro zhodnocení projektu je zásadní, zda bylo dosaženo vytýčených cílů. Proces přeměny informačního systému jich měl přinést celou řadu a v následující části se tato práce pokusí tyto požadavky shrnout, specifikovat aplikované řešení a

posoudit jeho efektivnost a optimalitu. Pokusí se nastítnit klady a zápory zvoleného řešení a v dalších částech nabídnout alternativy pro odstranění záporů.

#### **10.1.1.1 Schopnost obsloužit větší množství dat a více provozoven**

V tomto bodě bylo dosaženo kompromisu mezi přenosností a zpracovatelností dat. Zatímco skladové pohyby jsou řešeny efektivním přenosem dokladů v elektronické podobě a jejich následné zpracování a uchování probíhá v rámci Pohody, údaje o prodeji je potřeba exportovat do excelovské tabulky a teprve potom je přenášet a dále s nimi pracovat. Pro inventury a evidování velkých objemů údajů o pohybech ve skladech je to systém optimální. Pro analýzu prodeje a tržeb tento systém neposkytuje dostatek informací. Dle mého názoru by bylo potřeba zlepšit systém exportu dat.

#### **10.1.1.2 Zlepšení přehledu o zásobách a přímého prodeje, zavedení registračních pokladen**

Pro prodejny bylo zvoleno řešení, kdy v zázemí prodejny je počítač, sloužící jako server pro software Pohoda. Jako pokladna slouží počítačová sestava s dotykovým displejem, přidanou čtečkou čárového kódu, tiskárnou na paragony a je vybavena síťovým klientem Pohody. Pracuje se v agendě prodejek, která umí uloženému záznamu vytvořit tiskovou sestavu ve formátu paragonu, splňující všechna kritéria daňového dokladu.

Tento systém je poměrně pružný při editaci záznamů, obsluhování jednotlivých skupin s finančním zvýhodněním – je potřeba přiznávat slevy členům fanklubů, věrným zákazníkům, majitelům permanentních vstupenek, realizačním týmům a pod. Jednoduše je také realizovatelná výměna dříve zakoupeného zboží za jiné společně se zaúčtováním eventuálního finančního rozdílu.

Pro pravidelný denní prodej, kdy je důraz kladen na kvalitu obsluhy a nabízených služeb, je rychlost účtování dostatečná, je to tedy řešení optimální, které nám umožňuje mít v prodejně přesné informace o stavu skladu.

Přestává stačit v momentě velkého množství operací. Musíme mít na paměti, že se jedná o sportovní merchandising, kdy jednou z nejdůležitějších fází prodeje jsou jednotlivé zápasy. A při nich je potřeba obsloužit velké množství zákazníků za co nejkratší dobu. V tomto směru je účtování v Pohodě dosti neefektivní, k zaúčtování prodeje předmětu je potřeba až 4 kroků, což je velmi pomalé. Jelikož základem systému pokladny je operační systém Windows, nevyhneme se také občasným nepředpokládatelným chybám.

### **10.1.1.3 Centralizace informačního systému, především skladovnictví**

Tento požadavek dosti souvisí s předchozím bodem. Je důležité mít aktuální přehled o stavech jednotlivých skladů. Zvláště důležité je to při obsluhování internetového obchodu pouze v jedné z více provozoven – aby bylo možno nabízet zboží v obchodech i e-shopu se stejnou šířkou sortimentu a nebyl nabízen produkt, který není dlouhodobě skladem.

Stejně tak je velice podstatné, aby se dalo pružně reagovat na potřeby trhu. Je-li nějaký produkt v oblibě a je často prodáván, je potřeba mít přehled o aktuálním stavu zásob - dodací doba dodavatelů se pohybuje od týdne do 3 týdnů, s čímž je potřeba při objednávání počítat. Také je důležité sledovat, jaké množství alternativních produktů evidujeme, aby nedošlo k přezásobení. V rámci každé prodejny zvlášť je evidence zboží a jeho pohybů dostatečná (jak bylo zmíněno výše).

Centralizované řešení to však není, systém bohužel neumožňuje sdílet data o prodeích natolik obsažná, aby se daly na jejich základě provádět analýzy. Neexistuje centralizované místo, které by se dalo jednoduše aktualizovat a dokumentovalo by stavy všech našich skladů. Budeme-li předpokládat že se tendence rozrůstání udrží na stejné úrovni, bude potřeba zapojit efektivnější nástroje na získávání a analyzování dat.

#### **10.1.1.4 Restrukturalizace a zefektivnění internetového obchodu**

Internetový obchod naplnil mnoho očekávání. Cesta k naplnění však nebyla jednoduchá. Optimalizace systému do podoby, ve které by pracoval natolik dobře, aby byly potřeba pouze občasné aktualizace, trvala rok. Bylo potřeba reagovat na nové a nové úkoly, které za provozu vznikaly. Zefektivňování a zvyšování použitelnosti e-shopu probíhal na základě názorů, dotazů a přání jednotlivých zákazníků, které byly analyzovány a byly z nich činěny závěry. Dalšími, kdo velmi ovlivňoval podobu systému, byli ti, kteří se starají o provoz systému objednávek a internetový prodej.

System splnil požadavky na jednoduchost ovládání a omezil nutnost počítačové gramotnosti pro spravování objednávkového systému. Je efektivním nástrojem pro zaznamenávání jednotlivých kroků při zpracování objednávky a komunikaci se zákazníky i pro vnitropodnikové sdílení dat.

#### **10.1.1.5 Shrnutí**

Celkově se dá současný stav informačního systému FANzone.cz označit jako efektivní. Pro obsluhu stávajícího množství prodejen a dat je dostačující, má i značné rezervy. Tyto rezervy můžeme uvažovat pouze v obslužnosti, nikoliv z analytického hlediska. Zohledníme-li růstové tendence společnosti, můžeme i nadále očekávat progresivní vývoj, zvyšování množství poboček a tudíž zvýšené nároky na analytickou činnost. Procedury, které jsou používány pro malé objemy dat, nejsou použitelné při takovéto expanzi.

Vezmeme-li také v potaz že při akcích, neprobíhajících na stadionu v Edenu, tudíž v pojetí firmy na prodejně, která se zde nalézá, jsou záznamy prodejů stále dosti omezené. Budeme-li uvažovat o pořízení takových mobilních zařízení, která by dokázala prodeje na těchto prodejnách efektivněji zaznamenat, je již v tuto chvíli jasné, že budeme potřebovat nástroj pro analyzování velkého množství dat.

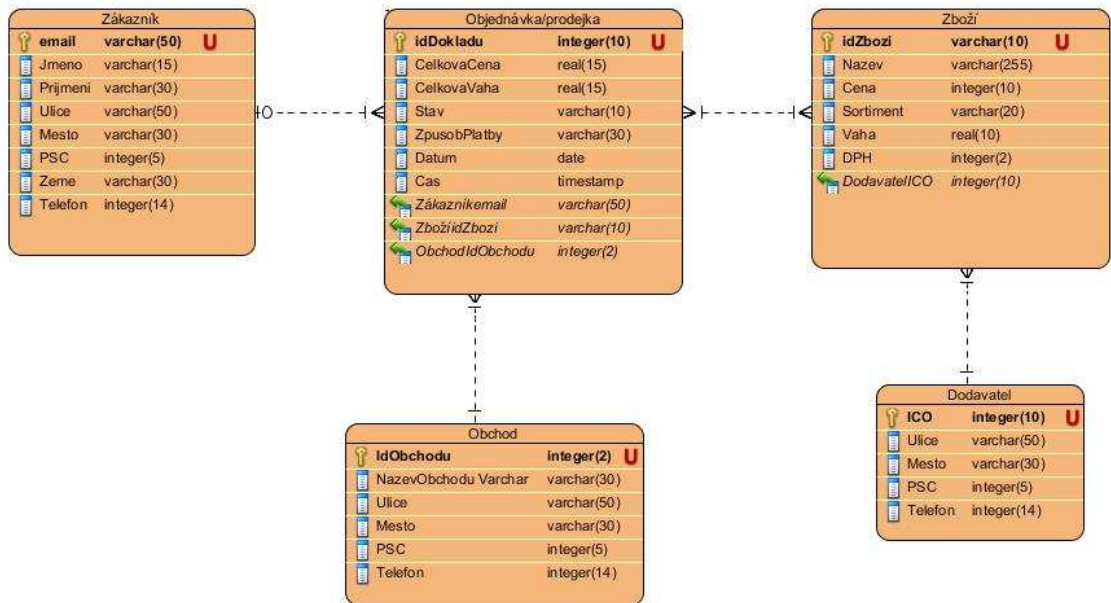
## **11 NÁVRH PODPORY ANALÝZY DAT**

Uvažujeme-li o zpracovávání většího množství dat, měli bychom vyslyšet trend současnosti a využít analytických možností datového skladu. Jelikož se v našem případě jedná o relativně jednoduchou databázi a malý objem dat, hodil by se nám spíše datový trh – datový sklad v malém.

### **11.1 Struktura ukládaných dat**

Aby bylo možno uchovávat všechny údaje, které budeme potřebovat pro následnou analýzu prodeje a pohybu zboží, je potřeba poněkud přeorganizovat stávající databázi uchovávaných záznamů. Schéma vychází z původního návrhu databáze, avšak zohledňuje větší nároky na množství informací, které potřebujeme evidovat, uchovávat a analyzovat.

Rozšíření se týká především získávání a uchovávání informací o prodeji na jednotlivých prodejních. Aby byl jejich provoz optimální, je potřeba tyto informace dále analyzovat. Internetový obchod se zařadil mezi ostatní obchody, dostal své id, proto mohly být sloučeny záznamy z prodejek i internetových objednávek. Dalším doplněním databáze je evidování data a času prodeje zboží – na základě frekvence prodeje lze určit rizikové časy a posílit v nich lidské zdroje. U zboží je evidován i dodavatel, jelikož ne všichni dodavatelé nabízejí stejné ceny ekvivalentních produktů.



obr. 3 - ER Diagram navržené databáze

Bylo by tedy potřeba modifikovat aktuální ekonomický systém a jeho výstupy odpovídaly výše uvedenému diagramu. Z této databáze můžeme následně vztyčit základy datového skladu.

## 11.2 Návrh datového skladu

Největší důraz je kladen na prodej a jeho analýzu, tabulka faktů se jej tedy bude bezprostředně týkat.

název	typ
idZbozi	varchar(10)
idObchodu	integer(2)
idCas	date
prodanoKusu	number(3)
celkovaCena	number(3.2)

tab. 1 - Fakta

Tabulky dimenzí odpovídají námi zvolené kategorizaci údajů, neboli kritériím, podle kterých chceme prodeje posuzovat. Jsou to Produktová dimenze, Časová dimenze a dimenze Obchodu.

produktová dimenze		dimenze obchodů		časová dimenze	
název	Typ	název	typ	název	typ
idZbozi	varchar(10)	idObchodu	number(2)	idCas	date
nazev	varchar(30)	oznaceni	varchar(15)	mesic	verchar(8)
popis	varchar(255)	ulice	varchar(50)	den	varchar(7)
sortiment	varchar(15)	mesto	varchar(30)	cisloTydne	number(2)
vaha	number(7,3)	PSC	varchar(5)	rok	number(4)
nakupniCena	number(10,2)	telefon	number(14)		
prodejniCena	number(10,2)	akce	varchar(20)		

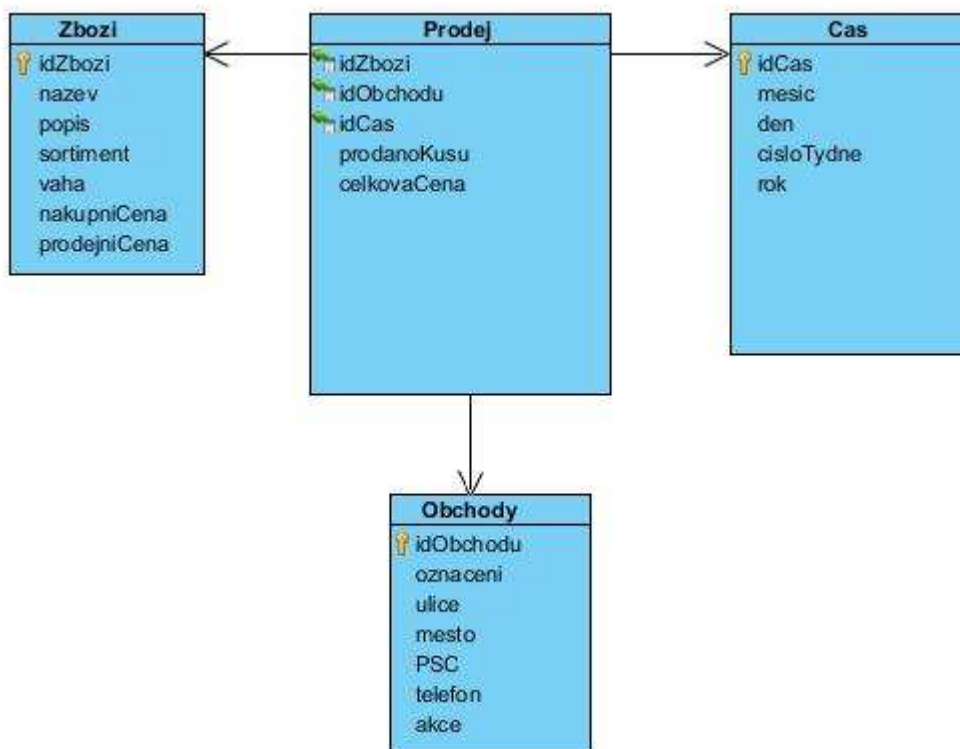
**tab. 2 - dimenze**

Je tedy patrné, že pro naši analytickou činnost je důležité časové určení prodeje. Časová dimenze má poněkud denormalizovanou strukturu, která však umožňuje vytvářet dimenze podle kalendářních zvyklostí.

U zboží je pro nás důležité sledovat nákupní a prodejní ceny, stejně tak sortiment. Ten, jak již bylo zmíněno výše, označuje jednotlivé části merchandisingové činnosti FANzone.cz.

Dimenze obchodů sleduje rozložení prodeje podle jednotlivých poboček. Na každé pobočce může být dominantnější jiná cílová skupina zákazníků a my bychom měli být schopni na tyto věci pružně reagovat.





**obr. 4 - model datového skladu**

Architektura datového skladu je postavena na hvězdicovitém schématu a odráží náš důraz na prodej. Takto orientovaný datový sklad bude přinášet možnosti analyzování prodeje ze všech pro naše účely potřebných úhlů pohledu. Proč právě důraz na prodej? Protože je alfou i omegou veškerého obchodování. A ze zkušeností, které jsem nabyl během let provozu FANzone.cz vím, že úspěšnost prodeje záleží na všech faktorech i detailech. A každá pomoc při odhadování chování spotřebitelů a trhu je významná.

## 12 ZÁVĚR

Věřím, že práce poskytla zajímavý pohled na budování informačního systému za ne zrovna ideálních podmínek. Ukázala, že řešení se dá nalézt i za velice omezeného rozpočtu a jak informační systém reaguje na změnu požadavků. Sledovala vývoj požadavků při zvětšování objemu dat a pole působnosti a já věřím, že najde praktické využití pro čerpání informací

Cíle byly naplněny, popis se věnuje celému vývoji a na závěr byl vytvořen návrh datového skladu tak, aby sledoval prognózovaný vývoj společnosti. Je však třeba mít na paměti, že datový sklad je věc ekonomicky dosti náročná, proto bych jeho nasazení doporučoval až při opravdu velkém objemu dat.

Věřím, že tato práce, která mi pomohla v pochopení fungování databázových systémů, pomůže i v rozhodování o budoucím vývoji firmy, o jejíž analýzu se pokusila.

## 13 SEZNAM POUŽITÉ LITERATURY

- [1] Relace [online], poslední aktualizace 1. 4. 2009 20:34 [cit. 1. 4. 2010], Wikipedie  
**[http://cs.wikipedia.org/wiki/Relace\\_\(matematika\)](http://cs.wikipedia.org/wiki/Relace_(matematika))**
- [2] VOSTROVSKÝ, Václav. Vytváření databázi v Oracle. Praha: Česká Zemědělská Univerzita v Praze Provozně ekonomická fakulta, 2004. str. 14
- [3] VOSTROVSKÝ, Václav. Vytváření databázi v Oracle. Praha: Česká Zemědělská Univerzita v Praze Provozně ekonomická fakulta, 2004. str. 61, 62
- [4] BABCOCK, Charles. *Sun Locks Up MySQL, Looks To Future Web Development*. InformationWeek. Retrieved on 2008-02-23
- [5] VOSTROVSKÝ, Václav. Vytváření databázi v Oracle. Praha: Česká Zemědělská Univerzita v Praze Provozně ekonomická fakulta, 2004. str. 111, 112
- [6] LACKO, Luboslav. Datové sklady, analýza OLAP a dolování dat, Brno: Computer Press, 2003, str. 111
- [7] MERUNKA, Vojtěch. Datové modelování. Praha: Alfa Publishing, 2006
- [8] LACKO, Luboslav. Datové sklady, analýza OLAP a dolování dat, Brno: Computer Press, 2003

- [9] HUMPHRIES, Mark – HAWKINGS, Michael – Dy, Michelle. Data warehousing – návrh a implementace. Praha: Computer Press, 2002
- [10] MERUNKA, Vojtěch – PERGL, Robert – PÍČKA, Marek. Objektově orientovaná tvorba softwaru. Praha: Česká Zemědělská Univerzita v Praze Provozně ekonomická fakulta, 2004