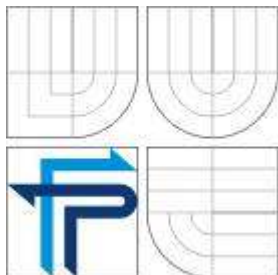




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUT OF INFORMATICS

NÁVRH FIREMNÍ DATABÁZE POMOCÍ SQL SERVERU A JEJÍ NÁSLEDNÉ VYUŽITÍ V INTRANETU

DESIGN OF COMPANY SQL DATABASE AND HER USING IN INTRANET

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR ŠÍSTEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JIŘÍ KRÍŽ, Ph.D.

BRNO 2008

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Šístek Petr

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

Návrh firemní databáze pomocí SQL Serveru a její následné využití v intranetu

v anglickém jazyce:

Design of company SQL database and her using in intranet

Pokyny pro vypracování:

Úvod
Vymezení problému a cíle práce
Teoretická východiska práce
Analýza problému a současné situace
Vlastní návrhy řešení, přínos návrhů řešení
Závěr
Seznam použité literatury
Přílohy

Abstrakt

Cílem bakalářské práce je návrh funkčního a jednoduchého, ale přitom mocného, databázového firemního systému pro evidenci potřebných dat pro firmu. Návrh databázových struktur a logických schémat. Hlavním cílem je zrychlení běžné práce a především možnost rychlého získání dat pro různé operační procesy.

Abstract

The purpose of this bachelor's thesis is design of functional and single company's database system for evidence needed data in company. Design of database structure and logical diagrams. Main purpose is to get faster work and faster data mining for different operations.

Klíčová slova

databáze, firemní databáze, SQL, intranet, entity

Keywords

database, company database, SQL, intranet, entities

Bibliografická citace mé práce:

ŠÍSTEK, P. *Návrh firemní databáze pomocí SQL serveru a její následné využití v intranetu*.
Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2008. 59 s. Vedoucí bakalářské práce Ing. Jiří Kříž, Ph.D.

ČESTNÉ PROHLÁŠENÍ:

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně.

Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským, ve znění pozdějších předpisů).

V Brně dne 30.5.2008

podpis

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Jiřímu Křížovi, Ph.D. za odbornou pomoc při řešení problémů během jejího zpracování.

Dále firmě ADAPTA, spol. s.r.o. za vzájemnou spolupráci a ochotu.

Obsah

ÚVOD	10
1 VYMEZENÍ PROBLÉMU A CÍLE PRÁCE	11
2 ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE	12
2.1 Profil firmy	12
2.2 SWOT analýza.....	13
2.2.1 Silné stránky:.....	13
2.2.2 Slabé stránky:	13
2.2.3 Příležitosti:	13
2.2.4 Hrozby:.....	13
2.3 Informační technologie	14
3 TEORETICKÁ VÝCHODISKA PRÁCE	15
3.1 Co je to databáze	15
3.2 Historie databází	16
3.3 Databázový systém	18
3.4 SŘBD	19
3.5 SQL servery	22
3.6 Historie databázových serverů a jazyka SQL	23
3.7 Podmnožiny jazyka SQL	24
3.8 Datové a funkční modelování.....	25
3.8.1 Datové typy	25
3.8.2 Integrita relačního modelu	25

3.8.3	Vztahy mezi tabulkami	26
3.8.4	Normalizace	26
4	VLASTNÍ NÁVRHY ŘEŠENÍ.....	28
4.1	Konceptuální návrh	28
4.2	Schéma databáze na logické úrovni.....	32
4.3	Datový slovník	33
4.3.1	Zaměstnanci	33
4.3.2	Pozice Zaměstnanci.....	34
4.3.3	Umístění Zaměstnanců.....	34
4.3.4	PSČ.....	34
4.3.5	Brigádníci	35
4.3.6	Umístění Brigádníků	35
4.3.7	Dodavatelé.....	36
4.3.8	Materiál Dodavatelé	37
4.3.9	Smlouvy Dodavatelé	37
4.3.10	Dodavatelé2.....	38
4.3.11	Zákazníci	38
4.3.12	Typ Zákazníci.....	39
4.3.13	Zákazníci2	40
4.3.14	Faktury.....	40
4.3.15	Zakázky	41
4.3.16	Typ Zakázky.....	42
4.4	Výběr platformy	42
5	PŘÍNOS (EFEKTIVNOST) NÁVRHŮ ŘEŠENÍ.....	44

ZÁVĚR	45
SEZNAM POUŽITÉ LITERATURY	47
Písemné zdroje publikované	47
Internetové adresy	48
SEZNAM POUŽITÝCH ZKRATEK	49
PŘÍLOHY	50

ÚVOD

Databáze a databázové systémy v dnešní době tvoří zřejmě již nepředstavitelnou součást běhu firem. Pomáhají uchovávat informace například o zákaznících, skladových zásobách, zakázkách, zaměstnancích, dodavatelích, odběratelích, účetnictví a v podstatě všeho dalšího co společnosti potřebují k běžné činnosti. Proto je na databáze upřen velký zájem nejenom odborné veřejnosti, ale i běžných uživatelů PC. Vzniká nespočet projektů, jež mají práci usnadnit, urychlit, zefektivnit či nějakým způsobem inovovat.

Důkazem velké popularity databází je i nepřehledné množství literatury pomáhající nejen začátečníkům a pokročilejším uživatelům, ale i profesionálům k objevení nových možností těchto systémů.

Ve chvíli realizace projektu se budeme muset rozhodovat mezi komerčními SQL servery, mezi které patří Microsoft SQL Server, Oracle, InterBase a mnoho dalších či tzv. open source programy jako je například MySQL. Přestože se jedná o oblast informatiky, nesmíme opomenout při výběru další hledisko, a to hledisko ekonomické. Neustále se hledá ideální spojení pro nejefektivnější práci za přijatelné finanční prostředky. Navíc obor ekonomie je definován jako „věda o rozhodování“, kde nám k tomuto nezbytnému rozhodování mohou posloužit i samotné databáze využívající principy datových skladů a Business Intelligence.

Při zpracování bakalářské práce jsem se snažil o co nejnižší náklady z důvodu minimalizace nákladů firmy a současně o její ekonomický přínos. Jak se říká čas jsou peníze, proto by práce s navrženým systémem měla být rozhodně rychlejší, přehlednější a snad i bezpečnější než doposud. Jelikož žádný z majitelů netuší co je jazyk SQL natož umět s ním pracovat, mělo by tuto efektivitu podpořit i mnou vytvořené grafické prostředí intranetu prostřednictvím jazyka PHP. Zachyceny jsou všechny prvky databáze s důrazem na jednoduché, intuitivní ovládání a oku barevně „příjemné“ prostředí.

1 VYMEZENÍ PROBLÉMU A CÍLE PRÁCE

Firma ADAPTA, spol. s.r.o. nemá žádnou aplikaci, která by alespoň trochu připomínala intranet a ani nepracuje s daty prostřednictvím databází. Veškeré informace, jež potřebuje ke své činnosti, jsou pouze na papírech umístěných v archivu či aktuálních skladových kancelářských prostor.

Proto cílem práce je návrh datových struktur, vytvoření dotazů pro nejčastěji používané žádoucí výsledky a případné naprogramování procedur či funkcí pro automatizaci některých činností. Tedy kompletní databáze zahrnující data pro potřeby běhu firmy jako jsou evidence zaměstnanců, brigádníků, dodavatelů, zákazníků a především zakázek. Kromě evidence je také důležité vytvoření dotazů, které se používají i několikrát denně. Nezbytný je systém vyhledávání pro rychlé získání potřebných informací. Všechny tyto procesy jsou umožněny ovládat skrze intranet. Tedy jednoduché formuláře a přehledné prostředí. Rád bych zmínil, že ve firmě došlo k jisté domluvě v případě nasazení, úspěšném zaběhnutí a zavládnutí spokojenosti s aplikací. Jednalo by se o rozšíření na část pro účetnictví. Ta by tvořila samostatný modul v rámci intranetu pouze pro účely daňové poradkyně.

Dalo by se říci, že takovým podúkolem je vytvoření prostředí intranetu. Jelikož firma nemá žádnou takovou aplikaci, byl jsem, za mne zodpovědnou osobou, pověřen jejím vytvořením. Celá tato aplikace běží v programovacím jazyce PHP, který sice není náplní této bakalářské práce nicméně v příloze uvádím několik obrázků intranetu.

Pro upřesnění bych dodal, že navržená výsledná aplikace intranetu není dokonalá, ale postupem času se bude rozšiřovat a ladit na základě flexibilních požadavků společnosti ADAPTA, s.r.o. k co nejkvalitnějším službám pro firmu.

2 ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE

2.1 Profil firmy

Stavební firma vznikla v roce 1990 jako družstvo pěti, později čtyř vlastníků. Roku 1992 se družstvo přetransformovalo na společnost s ručením omezeným tvořenou pěti společníky a přibližně deseti zaměstnanci. V roce 1996 se počet společníků snížil na tři odchodem dvou zakládajících společníků z firmy. Zbylí společníci se stále drží původního předmětu podnikání, čímž je stavební činnost. Počet zaměstnanců se v posledních letech stabilizoval na 25. Společnost pracovala i na zahraničních zakázkách, například pro Německou či Rakouskou firmu a podílela se i na stavebních pracích v Londýně a Francii.

Předmětem podnikání je provádění staveb včetně jejich změn, udržovacích prací na nich a jejich odstraňování. Zateplování budov a fasády. Firma není výrobním podnikem, ale pouze poskytující určitý sortiment služeb.

Sortimentem služeb je provádění obvyklé zednické práce jako je zdění, omítání, potěry, montáž některých zabudovávaných předmětů, provádění obkladů a dlažeb, provádění fasádních nátěrů a maleb.

Ostatní řemeslné práce je schopna zajistit formou dodávky na klíč pomocí spolupracujících specializovaných firem. Specializací je zejména provádění zateplovacích prací pomocí kontaktních zateplovacích systémů různých firem namátkou od firmy ALSECCO, TERRANOVA, BAYOSAN, HASIT, LAFARGE a dalších.

Součástí regenerace panelových domů je i sanace betonových ploch, jejíž realizaci firma provádí ze systémových produktů. K sanaci využívá produktů firem ALSECCO, MUREXIN a ARDEX.

2.2 SWOT analýza

2.2.1 Silné stránky:

Společnost se snaží nebrat zákazníka jako položku v seznamu zákazníků, ale ke každému má individuální přístup. Jednatelé a společník se snaží se svými klienty vycházet tak nějak lidsky a ne ve vztahu firma – klient. Možná se to zdá jako samozřejmost, ale z vlastní zkušenosti vím, že tomu tak není vždy.

Další silnou stránkou je jistě kvalifikace dělníků. Každý je proškolen zvlášť a velmi kvalitně. Samozřejmě zadarmo. Asi jako největší silnou stránkou firmy je, alespoň z mého úhlu pohledu, že to ty pány baví, zajímají se o nové stavební trendy, tudíž koníček se jim stal prací. A takříkajíc mají zájem o klienta a jeho požadavky a taky budoucnost firmy jim není lhostejná.

2.2.2 Slabé stránky:

Jistě je velkou slabou stránkou to, že se jedná se o malou firmu. Z toho plyne v podstatě silný konkurenční boj o zákazníka. Asi bych měl zmínit i nepříliš strategické umístění sídla, jež může být pro neznalé regionu tvrdým oříškem nalezení správné cesty.

2.2.3 Příležitosti:

Úzká spolupráce s dodavatelskými subjekty, jež na trhu mají již zavedené jméno. Přestože se jedná o firmu malou, každý má možnost dostat se na zahraniční trh. Další příležitostí bych uvedl možnosti velkého množství kontaktů.

2.2.4 Hrozby:

Jistě hrozbou je zdražení materiálů a tím i konečných částek pro zákazníky. Konkurenční boje a udržení alespoň minimálního podílu na trhu jsou zřejmě největšími hrozbami pro firmu s přídomek malá.

2.3 Informační technologie

Společnost disponuje třemi počítači. A to jedním nejvýkonnějším, který plní funkci serveru. Na něm jsou čteny potřebné výkresy k daným stavbám, různé důležité dokumenty a především komunikace pomocí elektronické pošty. Slouží i k propojení s digitálním fotoaparátem a následnou prací s fotografiemi, propojení s mobilním telefonem, faxem, multifunkční tiskárnou a dalším příslušenstvím.

Druhý je svým výkonem na střední úrovni v porovnání s ostatními. Také se od něj žádný výkon neočekává, jelikož slouží pouze pro kancelářské využití, konkrétně účetnictví, knihu jízd apod.

Třetí bych snad ani neměl zmiňovat, protože je velmi starý, hardwarově pomalý a čeká se pouze na jeho dosloužení. Každopádně nějaké minimální práce na něm stále probíhají.

Počítače jsou vzájemně propojeny jako jedna LAN síť, kde serverem je již zmiňovaný nejvýkonnější počítač značky ACER. Všechny mají samozřejmě připojení na internet o rychlosti 2 Mbit/s s typem připojení ADSL.

Co se týče softwaru, jsou počítače vybaveny antivirovým systémem Norton Internet Security 2007, Microsoft Office 2000 Premium, SQL Serverem (z důvodu elektronického bankovníctví a spojení s bankou). Samozřejmě vše jsou legálně zakoupené programy. Disponuje i s pár volně šiřitelnými programy, tzv. freeware, které jsou využívány pro usnadnění práce některých úkonů.

3 TEORETICKÁ VÝCHODISKA PRÁCE

3.1 Co je to databáze

Databázi si lze představit jako soubor informací sloužící pro popis reálného světa, kupříkladu evidence studentů a jejich známek, evidence zboží na skladě či evidence obyvatelstva. Tyto fakta jsou uložena na jediném centrálním místě.

Každý databázový systém je složen z následujících třech částí:

- 1) *Systém řízení báze dat (SŘBD)*, což je program pro organizaci a údržbu dat
- 2) *Databázová aplikace*, program umožňující výběr, prohlížení a aktualizaci informací
- 3) *Databáze*, tedy báze (základna) dat

Prvek ze skutečného světa převedený do terminologie databází je označován pojmem entita. Může jí být třeba člověk, vyučovaný předmět, auto, sklad apod. Vlastnosti tohoto prvku jsou atributy (jméno, příjmení, bydliště, plat, rodinný stav,...).

Dále je třeba mít na paměti existenci jistého databázového modelu, což je vlastně matematický popis databáze. Nejprve se užívaly modely dvojího typu a to model hierarchický, který je založen na modelování principu hierarchie mezi entitami, tzn. vztahy podřízenosti a nadřízenosti a síťový model vycházející z teorie grafů. Uzly v grafu odpovídají entitám a orientované hrany definují vztahy mezi entitami.

V 70. letech se tyto modely ukázaly býti nedostatečnými z důvodu problémů realizace a implementace vazby N:M, a proto vznikl relační model, který se stal standardem pro databáze a je používán dodnes.

3.2 Historie databází

konec 19. století – počátek databází

1890 – na žádost vlády USA vytvořil Herman Hollerith první automat, který pracoval s děrnými štítky

1911 – Herman Hollerith spolu s dalšími společníky založil firmu IBM existující dodnes

1935 – vytvoření prvního digitálního počítače UNIVAC I pro komerční využití na žádost vlády USA za účelem zavedení SSA (každý občan má své číslo, obdoba rodného čísla v ČR), stroj pracoval s děrnými štítky

60. léta – vznikají pojmy jako databáze, entita, atribut entity a vazba mezi entitami využívané dodnes

1961 – Charles Bachman představuje první integrovaný datový sklad s náznakem databázového managementu

1970 – Ted Codd (nespokojený zaměstnanec IBM) publikuje článek „A Relation Model of Data for Large Shared Data Banks“ (Relační model dat pro rozsáhlé sdílené banky dat) týkající se návrhu implementace nového datového modelu nazvaného jako relační

1974-1975 – první fáze vývoje u IBM, praktické ověření návrhu relačního modelu

1978-1979 – implementace plně funkčního víceuživatelského systému

1980 – první SQL databáze ORACLE na trhu

80. léta – lze považovat za počátek zlatého věku relačních databází

Relační databáze

Codd ukázal možnost, jak použít relační kalkul a algebru i pro netechnické uživatele při ukládání a manipulaci s daty. Podle něj k tomu mělo být používáno srozumitelných příkazů, které vycházejí z běžné angličtiny. Již tato koncepce počítala s ukládáním dat do tabulek. Návrh byl založen na nezávislosti dat na použitém hardware, na způsobu jejich fyzického uložení, dále pak na přístupu k datům pomocí ne-procedurálního jazyka. Uživatel měl mít možnost specifikovat určitou operaci nad množinou dat jím definovanou místo pouhé manipulace s jedním záznamem.

Dle relační teorie lze pomocí základních operací (sjednocení, kartézský součin, rozdíl, selekce a projekce) uskutečnit veškeré operace s daty a ostatní jsou pouze jejich kombinacemi.

Jen pro zajímavost prvotní databáze byly testovány s 8 MB dat, v dnešní době databáze obsahují terabyty dat.

Codd definoval pro relační model **13 pravidel**:

- 1) **Relační databázový řídicí systém (RDBŘS)** musí být schopen řídit databáze prostřednictvím jejich relační schopností
- 2) **Informace** – všechny informace v relační databázi (včetně názvů tabulek a sloupců) jsou představovány výlučně hodnotou v tabulkovém formátu
- 3) **Zaručený přístup** – každá hodnota v relační databázi musí být přístupná prostřednictvím kombinace názvu tabulky, hodnoty primárního klíče a názvu sloupce
- 4) **Systematická podpora prázdných hodnot** – RDBŘS poskytuje systematickou podporu pro zpracování prázdných hodnot (neznámá nebo neupotřebitelná data), které se musí lišit od výchozích hodnot a být nezávislé na jakékoli doměně
- 5) **Aktivní, vždy dostupný relační katalog** – popis databáze a jejího obsahu je na logické úrovni reprezentován rovněž formou tabulky a informace lze z něho získávat pomocí databázového jazyka
- 6) **Všeobecný datový podjazyk** – musí existovat alespoň jeden jazyk s vhodně definovanou syntaxí a musí být všeobecný, musí umožňovat definici dat, manipulaci s daty, práci s pravidly integrity, ověřování totožnosti a realizaci transakcí.
- 7) **Aktualizace pohledů** – Systém musí umožňovat aktualizaci dat ve všech pohledech, které jsou teoreticky aktualizovatelné
- 8) **Vkládání, aktualizace a odstraňování na úrovni množin** – relační databázový systém musí umožňovat nejen výběr, ale i vkládání, aktualizaci a odstraňování dat na úrovni množin

- 9) **Fyzická nezávislost dat** – aplikační programy a programy „ad hoc“ nesmí vyžadovat modifikaci, dojde-li ke změně interní struktury úložiště nebo metody fyzického přístupu k němu
- 10) **Logická nezávislost dat** – aplikační programy a programy „ad hoc“ nesmí být při provedení změn v tabulkových strukturách na logické úrovni do maximální možné ovlivněny
- 11) **Nezávislost integrity** – databázový jazyk musí umožňovat definici pravidel integrity, tato pravidla musí být uložena v neustále přístupném katalogu a nesmí být možné je obejít
- 12) **Nezávislost distribuce** – aplikační programy a požadavky „ad hoc“ nesmí vyžadovat modifikaci při prvotní ani opakované distribuci dat na jiné počítače
- 13) **Nenarušitelnost** – má-li databázový systém nízkourovňový (procedurální) programovací jazyk, tomuto jazyku nesmí být umožněno obcházet pravidla integrity definovaná databázovým jazykem¹

3.3 Databázový systém

Pojmem databázový systém se rozumí jednak údaje uložené a spravované v databázi, ale i program neboli software pro přístup k těmto údajům. Souhrnem nástrojů, postupů a technik, využívaných v souvislosti s databázemi, je nazývána databázová technologie. Databázové systémy rozdělujeme na několik základních typů:

- a) **hierarchické a síťové** – aplikační programy jsou závislé na databázi, z toho plyne například problematická údržba,
- b) **relační** – u těchto databází je typická neprocedurální manipulace s daty, ukládání dat s pevnou strukturou v tabulkové formě,
- c) **objektové** – používají složité datové struktury a složitá pravidla založená na obchodní logice známe pod názvem Business rules.

¹ STEPHENS, R. a PLEW, R. *Naučte se SQL za 21 dní*. 1. vyd. Brno:Computer Press, 2004, s. 34, ISBN 80-7226-870-8.

3.4 SŘBD

Neboli systém řízení báze dat je objekt obstarávající přístup k uloženým údajům v databázi. Díky tomuto softwaru nemusí uživatel znát fyzickou strukturu uložených údajů, jelikož právě k těmto datům přistupuje pomocí SŘBD.

Prevažná většina SŘBD při uspořádání dat používá relační model. V tomto modelu jsou data uspořádaná do tabulek. Tabulka (entita) většinou shromažďuje informace o jednom druhu objektu. Sloupce tabulky se nazývají položky nebo atributy, řádkům říkáme záznamy. Každá tabulka má své jedinečné jméno.

Komunikace klienta či aplikačního programu SŘBD probíhá za pomoci jazyka SQL. Většina SŘBD v současných programech má charakteristické standardní rysy, jimiž jsou:

Transakční zpracování

Při složitější manipulaci s údaji složených z posloupnosti určitých kroků, je vykonávána jako transakce, která databázi převede z jednoho konzistentního stavu do druhého. Čili buď všechny operace v transakci proběhnou úspěšně a nebo neproběhnou vůbec. Tzn. že databáze bude uvedena do stavu v jakém byla před transakcí.

Zotavení se z chyb a nehod

Pokud při zpracování údajů dojde k chybě, musí být systém schopen se z této chyby zotavit a pokračovat dále v práci. Například obnovením databáze ze záložních souborů.

Víceuživatelský přístup

Předpokládá se přístup k databázi a příslušným údajům více uživatelů. Proto musí systém umět definovat různé uživatelské role pro efektivní řízení přístupu k údajům.

Ochrana údajů

Myšleno například při havárii počítače či fyzické krádeži. Musí být zabráněn přístup k těmto datům a eliminace možnosti zneužití dat.

Architektura

V závislosti na velikosti, objemu a místu vlastního zpracování lze architekturu rozdělit do čtyř základních kategorií neboli platforem:

Centralizované

Na hlavním (hostitelském) počítači jsou zpracovávány všechny programy potřebné pro práci s databází a komunikaci s ní. Uživatelé k ní přistupují z lokálních či vzdálených terminálů. Aplikace i SŘBD běží na témže počítači a vzájemná komunikace probíhá přes sdílené oblasti paměti či paměťové oblasti přidělené aplikaci, spravované operačním systémem hostitelského počítače. SŘBD je zodpovědné za přenos dat na a z disku s využitím služeb operačního systému.

Výhody:

- centrální zabezpečení
- schopnost uložení velkého množství dat
- současná práce více uživatelů

Nevýhody:

- náklady na pořízení a údržbu
- velká kvalifikovanost operátorů a systémových programátorů (mzdy)

Systémy ne osobních počítačích (PC)

Pokud běží SŘBD na PC, pak toto PC pracuje nejen jako hostitelský počítač, ale i jako terminál. Funkce SŘBD a databázových aplikací jsou spojeny do jednoho programu, čímž se dosahuje značné mohutnosti, flexibility a rychlosti ovšem na úkor snížení bezpečnosti a integrity dat.

Výhody:

- víceuživatelský přístup ke sdíleným datům

Nevýhody:

- výkonnost systému limitována výkonem PC
- možnost měnit data i mimo SŘBD

Databázové systémy klient/server (C/S)

Tento typ rozděluje zpracování mezi dva systémy, a to PC v roli klienta, na němž běží databázová aplikace a databázový server, na kterém běží vlastní SŘBD nebo jen jeho část. Tedy uživatel vytvoří požadavek na data z databázového serveru a aplikace vyše požadavek na server. Ten provede vlastní vyhledání a pošle zpět pouze data odpovídající uživatelova dotazu.

Výhody:

- redukce množství dat posílaných po síťovém médiu
- přenos pouze konkrétních dotazů a výsledků místo celých souborů
- některé procedury a dotazy uloženy přímo na serveru = další redukce
- aplikační nezávislost
- udržení integrity dat
- transakční zpracování

Nevýhody:

- zvýšené personální náklady na obsluhu a údržbu databázového serveru
- náklady na hardware

Největší nevýhodou výše popsaných systémů je požadavek na uložení dat na jediném počítači. Kupříkladu velké společnosti mající své pobočky v různých městech nemohou propojit jednotlivá oddělení s hlavním hostitelským počítačem. Tento problém řeší:

Systémy distribuovaného zpracování

Data jsou sdílena mezi různými hostitelskými systémy tak, že se mezi nimi posílají veškeré změny přímým spojením (ve stejné síti) a nebo dálkově po vyhrazených linkách. V případě změny dat v jedné pobočce dojde k aktualizaci databází na všech ostatních pobočkách. Data jsou posílána do centrálního počítače po skončení pracovní doby. Uživatel nejdříve vytvoří požadavek, který odešle databázovému lokálnímu serveru. Ten vyše požadavek na data, která sám neobsahuje, po síti na střediskový počítač. Po odpovědi na dotaz ji zkombinuje s údaji na vlastních discích a vše pošle zpět uživateli. Další možností je, že uživatelé u terminálů připojených přímo ke

střediskovému počítači mohou přistupovat k datům, jež jsou uloženy na vzdálených databázových serverech.

Výhody:

- aplikační program nepotřebuje znát místo uložení dat
- uživatel vidí pouze svá data nikoliv cestu k nim
- systém sám automaticky zajistí dodání dat do místa, kde je uživatel
- zvýšená spolehlivost

Nevýhody:

- vyšší provozní náklady
- obtížnější kontrola datové integrity

3.5 SQL servery

Předtím než započne vlastní návrh datových struktur, vztahů apod. je velice důležitý výběr databázového serveru. Je mnoho faktorů a kritérií pro jejich správný výběr. Základním rozdělením jsou programy **komerční**, za které se platí nemalé částky, ale za to s velkou spolehlivostí, vyčerpávající dokumentací a kvalitním zákaznickým servisem a tzv. **open source** programy. Ty jak název napovídá mají otevřený zdrojový kód a jsou tudíž volně k dispozici, ale příliš velká spolehlivost se očekávat nedá, dokumentaci pokud najdeme někde na internetu, v lepším případě přímo od výrobce a servis se v drtivé většině neposkytuje. Proto velké instituce jako jsou banky, úřady, pojišťovny a jiné jim podobné raději vynaloží vyšší náklady a sáhnou po komerčních produktech. Kdežto malé firmy či pro účely rodinné databáze zvolí uživatel z důvodu minimálních nákladů open source.

Dalšími důležitými kritérii jsou zcela jistě kompatibilita s ostatními programy, výkon databázového serveru, jakým způsobem jsou data interpretována (zda-li je potřeba vytvořit uživatelské prostředí, podporuje-li databáze komunikaci s jazykem v němž je prostředí vytvořeno, či uživatelský interface už obsahuje apod.), ale i lokalizace pro území na němž bude provozován. Česká republika je velice specifická z důvodu diakritiky a mnoho open source programů tuto problematiku nebere v potaz.

Komerční SQL servery

V současné době je na výběr celá řada produktů databázových serverů. Zmíním se jen o těch nejrozšířenějších. Nejvíce známý je **SQL Server** od firmy Microsoft, nyní ve verzi 2005 a v blízké době bude k dispozici i 2008. Společnost nabízí na svých internetových stránkách možnost stažení SQL Serveru 2005 v edici Express Edition, která je zcela zdarma, obsahuje pokročilé funkce, ale je samozřejmě „osekaná“ o mnoho důležitých nástrojů. O nic méně je úspěšný současný produkt **Oracle 11g**, jež je k dispozici také zdarma, od firmy Oracle. Neměli bychom opomenout další, širší veřejnosti jistě známější, software od Microsoft a to **MS ACCESS**, který je součástí kancelářského balíku MS Office. Ovšem na vedení složitých a náročných firemních databází není příliš vhodný.

Open source programy

Jsou programy distribuované zcela zdarma a volně ke stažení. Většinou nemívají žádnou technickou podporu a v lepším případě nalezneme na stránkách vydavatele referenční příručku. Svou silou je nejznámější a velmi hojně používaný **MySQL**. Není zde žádné grafické prostředí, nýbrž pouze příkazový řádek. Některé nástroje ani neobsahuje oproti komerčním serverům. Dále bych neměl opomenout **PostgreSQL**. Ten vyniká stabilitou, rychlostí a dobrou podporou. Z počátku byl určen pro operační systém UNIX, ale od verze PostgreSQL 7.5 nastala v tomto přístupu změna.

3.6 Historie databázových serverů a jazyka SQL

První produkty databází se zabudovaným SQL jazykem byly od firmy IBM (produkt DB2), SQL/DS. S nástupem osobních počítačů začali i ostatní výrobci stále více uvažovat o SQL. Zřejmě nejznámější je dBASE IV od společnosti Ashton-Tate. Ovšem zde nebyla implementace SQL úplná. Dalšími byly INFORMIX-SQL, INGRES pro PC, ORACLE, SQLBase, XDB II, XQL. S rozmachem jazyka se začal zabudovávat i na systém UNIX. Jmenujme aspoň SŘBD ORACLE, INFORMIX.

Historie SQL (Structured Query Language – strukturovaný dotazovací jazyk) se začíná psát roku 1974 uveřejněním první práce Dr. Codda. První použitelnou verzi vyvinula IBM pod názvem SEQUEL (Structured English Query Language). Už od

počátku bylo zřejmé vytvoření jazyka blízkého angličtině. K tomuto standardu se postupně přidávaly i jiné firmy (Oracle, Sybase, Informix). Původně byl jazyk určen pro produkt DB2, který lze sehnat i dnes. Postupně byly přijaty upravené a vylepšené standardy jazyka SQL s názvem SQL-86, SQL-92 (vžil se zkrácený název SQL-2). V současné době je dostupný standard SQL3 (SQL-99).

3.7 Podmnožiny jazyka SQL

DDL (Data Definition Language) – Jazyk pro definici dat

Slouží k definici struktur a vytváření objektů v databázi, jako jsou tabulky, pohledy apod., či k změně jejich struktury anebo zrušení (odstranění). Patří sem například tyto příkazy:

CREATE DATABASE, CREATE TABLE, CREATE INDEX, CREATE VIEW,
ALTER TABLE, ALTER VIEW, DROP TABLE, DROP INDEX, DROP VIEW

DML (Data Manipulation Language) – Jazyk pro manipulaci dat

Je určen k manipulaci s daty, především jejich výběr, vkládání a aktualizace. Zahrnuje tyto hlavní příkazy:

SELECT, INSERT, UPDATE, DELETE

DCL (Data Control Language) – Jazyk pro řízení dat

Jedná se o speciální podmnožinu příkazů k provozu a údržbě databáze a přidělování práv.

CREATE USER, ALTER USER, DROP USER, GRANT, REVOKE

TCC (Transaction Control Commands) – Jazyk pro řízení transakcí

SET TRANSACTION, SAVEPOINT, ROLLBACK, COMMIT

3.8 Datové a funkční modelování

3.8.1 Datové typy

Textové údaje

CHAR (n) – libovolný řetězec o délce 1 – 255

VARCHAR (n) – viz. CHAR, rozdíl je ve velikosti (pokud u CHAR napíšeme například délku 5 a vložíme sem hodnotu o délce 3 zbývající znaky doplní mezerou, kdežto VARCHAR prázdná místa mezerou nevyplňuje)

Číselné údaje

NUMERIC (x,y) – reálné číslo

INTEGER – celé číslo včetně znaménka

DECIMAL (x,y) – desetinné číslo s pevnou desetinnou čárkou

REAL – číslo v pohyblivé řádové čárce

Datové a časové údaje

DATE – kalendářní datum

TIME – denní čas

TIMESTAMP – „časové razítko“

INTERVAL – časový interval

Ostatní

Boolean – logické hodnoty true / false

Datových typů je mnohem více, zde je pouze výčet těch nejpoužívanějších.

3.8.2 Integrita relačního modelu

a) doménová integrita

Hodnota každého z atributů musí být z množiny (domény) pro daný atribut přípustných.

b) entitní integrita

primární klíč – jednoznačná identifikace každého řádku

kandidátní klíč – to stejné jako primární, akorát není vybrán, je spíše alternativní volbou

c) referenční integrita

cizí klíč – umožňuje propojení tabulky s jinou tabulkou např. číselníkem

3.8.3 Vztahy mezi tabulkami

1:1 – jednomu záznamu z tabulky odpovídá právě jeden záznam tabulky jiné (jeden člověk může mít pouze jeden řidičský průkaz a jeden konkrétní průkaz vlastní pouze jeden konkrétní člověk)

1:N – jednomu záznamu z tabulky odpovídá více záznamů tabulky jiné (jeden člověk vlastní více kreditních karet ale jedna konkrétní karta přísluší pouze jednomu konkrétnímu člověku)

N:1 – to stejné jako 1:N akorát se na ni díváme z opačné strany

N:M – několika záznamům jedné tabulky odpovídá více záznamů jiné tabulky (jeden student studuje více předmětů, ale jeden konkrétní předmět navštěvuje také více studentů)

3.8.4 Normalizace

1. normální forma – multizávislost

Relace je v první normální formě, pokud jsou všechny její atributy definovány nad skalárními obory hodnot (doménami).² Všechny atributy musí být jednoduché a ne složené či vícehodnotové.

² KOCH, M. *Datové a funkční modelování*. 1. vyd. Brno: Akademické nakladatelství CERM, s.r.o, 2004. s. 56. ISBN 80-214-2724-8.

2. normální forma – funkční závislost

Relace je ve druhé normální formě, pokud je v první normální formě a navíc všechny její atributy jsou závislé na celém kandidátním (primárním) klíči.³

3. normální forma – tranzitivní závislost (Boyce-Coddova normální forma)

Relace je ve třetí normální formě, pokud je ve druhé normální formě a navíc všechny její neklíčové atributy jsou vzájemně nezávislé⁴. Boyce – Coddova normální forma je pouze variací třetí normální formy a platí, je-li relace v Boyce – Coddově normální formě je i ve třetí normální formě. Ovšem obráceně to neplatí. Dá se používat jen pro specifické případy a ne vždy se jich dá dosáhnout.

4. normální forma

Relace je čtvrté normální formě, pokud je v Boyce – Coddově normální formě, a navíc všechny vícehodnotové závislosti jsou zároveň funkčními závislostmi z kandidátních klíčů (v jedné relaci se nesmí spojovat nezávislé opakované skupiny).⁵

5. normální forma

Týká se případu spojené závislosti, která vyjadřuje cyklické omezení, pokud je relace₁ spojena s relací₂, relace₂ je spojena s relací₃ a relace₃ je spojena zpětně s relací₁, pak všechny tři entity musí být součástí stejného vektoru hodnot.⁶

³ KOCH, M. *Datové a funkční modelování*. 1. vyd. Brno: Akademické nakladatelství CERM, s.r.o, 2004. s. 58. ISBN 80-214-2724-8.

⁴ KOCH, M. *Datové a funkční modelování*. 1. vyd. Brno: Akademické nakladatelství CERM, s.r.o, 2004. s. 60. ISBN 80-214-2724-8.

⁵ KOCH, M. *Datové a funkční modelování*. 1. vyd. Brno: Akademické nakladatelství CERM, s.r.o, 2004. s. 63. ISBN 80-214-2724-8.

⁶ KOCH, M. *Datové a funkční modelování*. 1. vyd. Brno: Akademické nakladatelství CERM, s.r.o, 2004. s. 64. ISBN 80-214-2724-8.

4 VLASTNÍ NÁVRHY ŘEŠENÍ

V této části se budu zabývat možnostmi vlastního návrhu pro vzniklý problém popsaný výše. Veškeré datové části mají pouze charakter evidenčního typu. To znamená pouze zapisování, úprava, mazání a operace s daty. Snažil jsem se vyhnout účetním položkám, jelikož ve firmě je již používán a zaběhnutý ekonomický software. Ten pracuje na stejné úrovni jako databáze „intranetová“. Ovšem jisté položky se neobešly bez uvedení těchto údajů. Nicméně neměl by být problém v budoucnu tyto dva moduly vzájemně propojit a vytvořit tak ještě mocnější nástroj.

4.1 Konceptuální návrh

Na základě poznatků a konzultací se veškeré údaje točí kolem pěti hlavních tabulek jimiž jsou zaměstnanci, brigádníci, dodavatelé, zákazníci a zakázky. Tabulky uchovávají důležité informace příslušící dané entitě. Uvedu například z tabulky zaměstnanců informace potřebné pro komunikaci ať už korespondenční či telefonickou, adresa bydliště s uvedením poštovního směrovacího čísla apod.

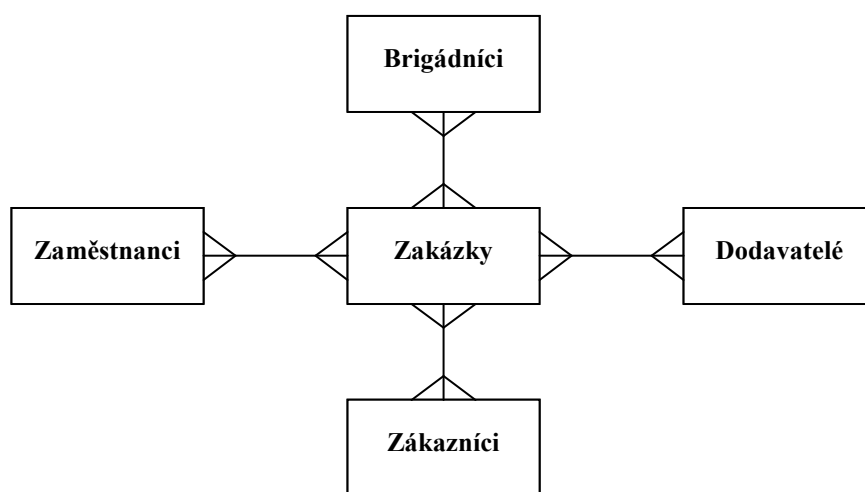
Dalšími entitami vytvořených v databázi jsou rozšiřovacího charakteru jako například tabulky smluv, faktur apod. Jejich údaje v nich jsou bezesporu také velmi důležité. Slouží spíše k propojení dvou hlavních tabulek za pomoci rozšiřovacích k dosažení efektivních výsledků, jež je dosaženo pomocí principů relačních databází.

Posledním typem entit vyskytujících se v návrhu jsou číselníky, tedy jakýsi seznam hodnot, které se smí objevit v na něj propojené tabulce a poli. Jejich princip spočívá v tom, že v hlavní tabulce je potřeba udat údaj, ale nebylo by efektivní jej pokaždé vypisovat ručně. Navíc pokud zvážíme možnosti různého druhu psaní. Píšeme-li například pohlaví muž, někdo zapíše Muž jiný zase muž. Kdežto pokud je propojeno s číselníkem máme zaručenou zobrazovanou pouze hodnotu v něm zapsaném. Proto se v tabulce odkazující na číselník zapíše pouze číslo s datovým typem integer zabírající pouze čtyři bajty celkové velikosti databáze. Toto číslo tvoří primární klíč v tabulce

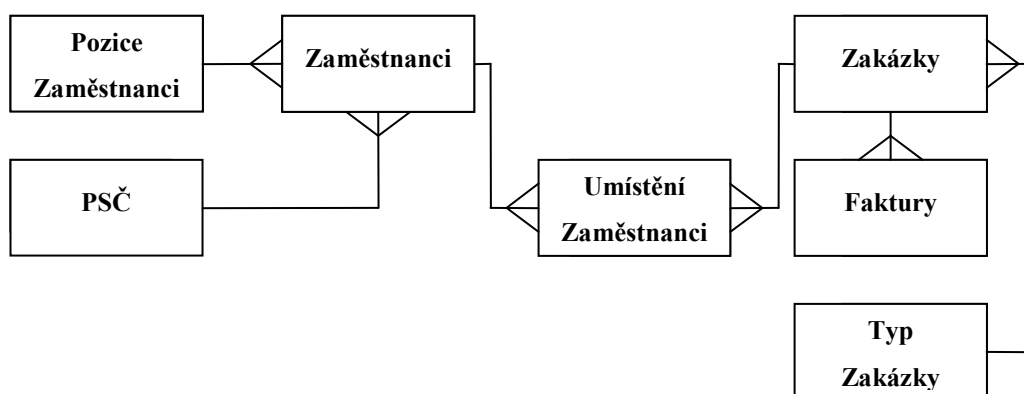
číselníku. Pokud budeme do číselníku vkládat novou hodnotu, je třeba zajistit její automatické označení. Toho se dosáhne za pomoci tzv. automatického primárního klíče, kdy při každém novém vložení se předešlá hodnota zvýší o jedničku. Typickým případem číselníku je tabulka poštovních směrovacích čísel.

Spousta položek má dáno omezení jakožto nenulové pole. Ty jsou společností označeny za povinné a nezbytné pro administraci.

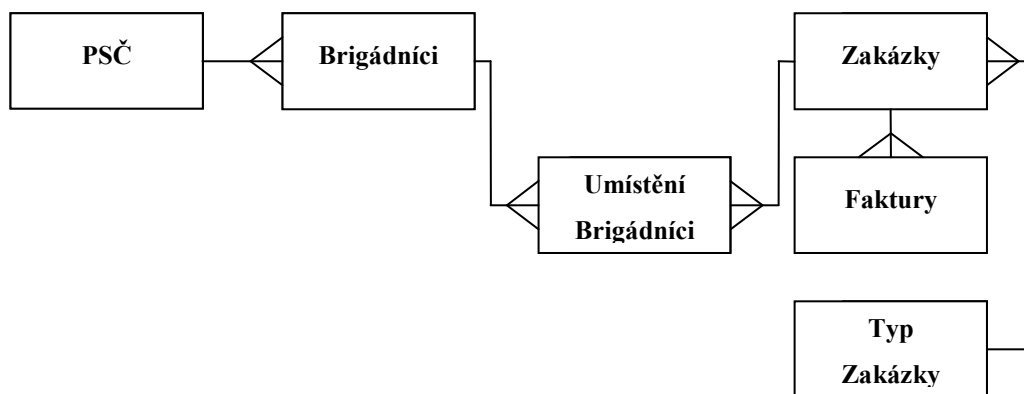
Obrázek č. 1: Konceptuální návrh databáze



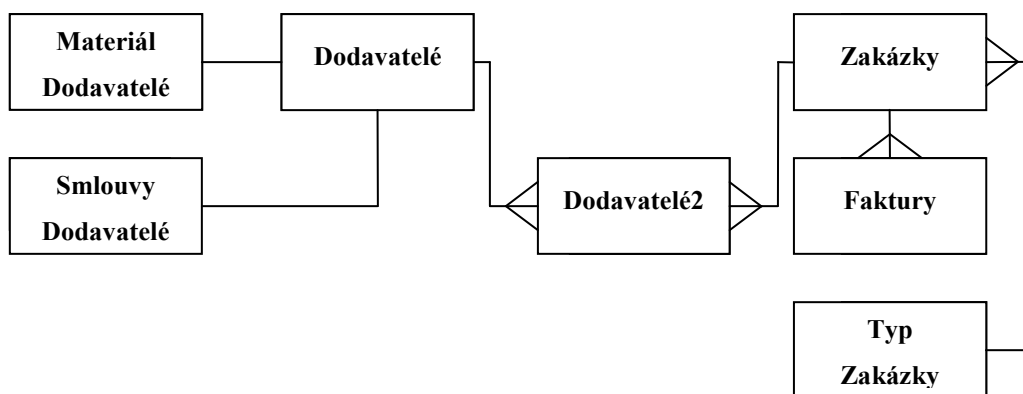
Obrázek č. 2: Dekompozice vazby Zaměstnanci / Zakázky



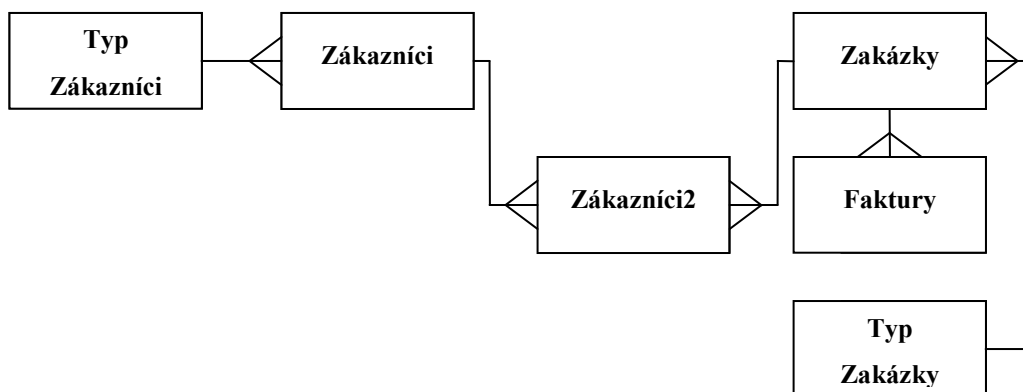
Obrázek č. 3: Dekompozice vazby Brigádníci / Zakázky



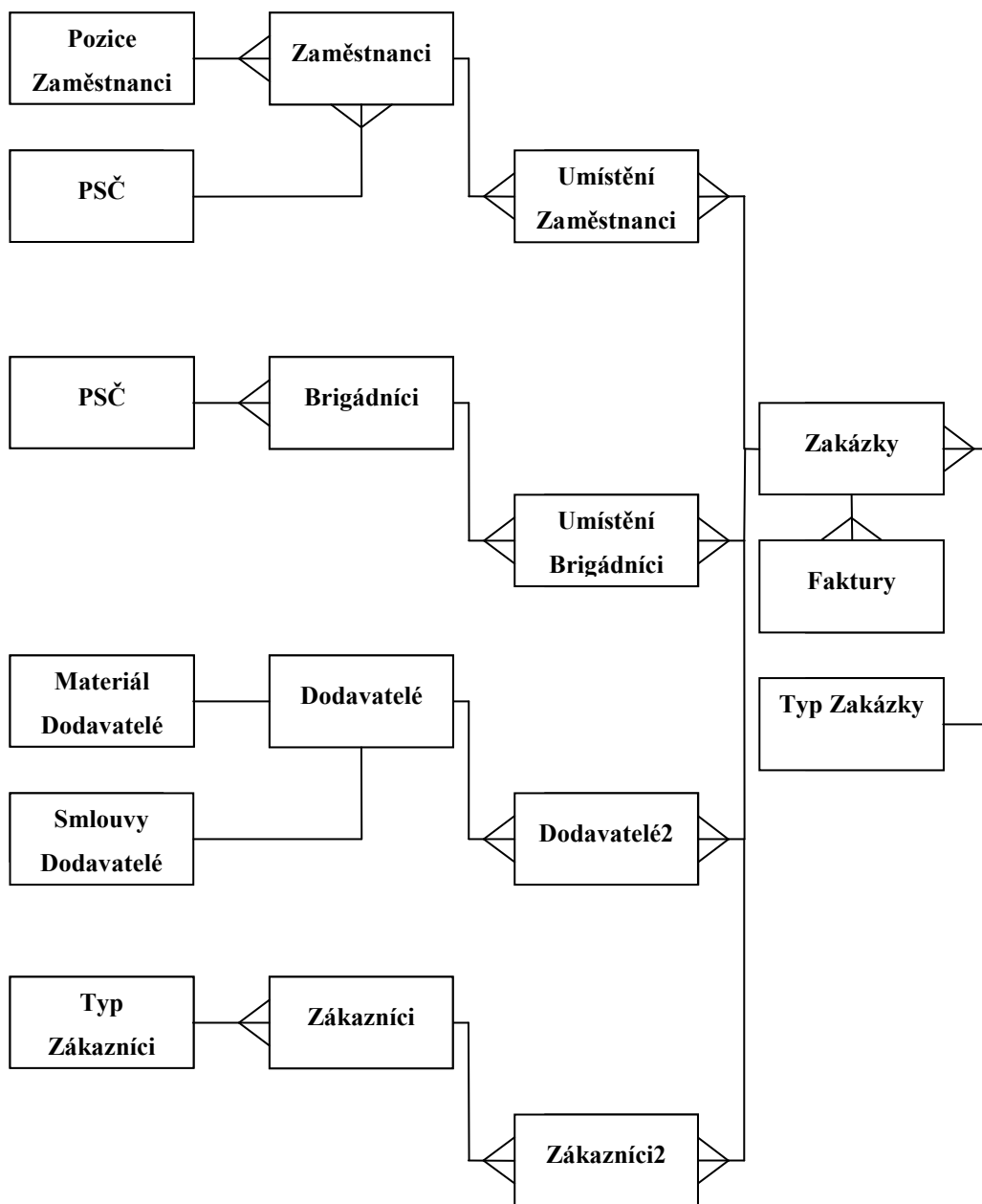
Obrázek č. 4: Dekompozice vazby Dodavatelé / Zakázky



Obrázek č. 5: Dekompozice vazby Zákazníci / Zakázky



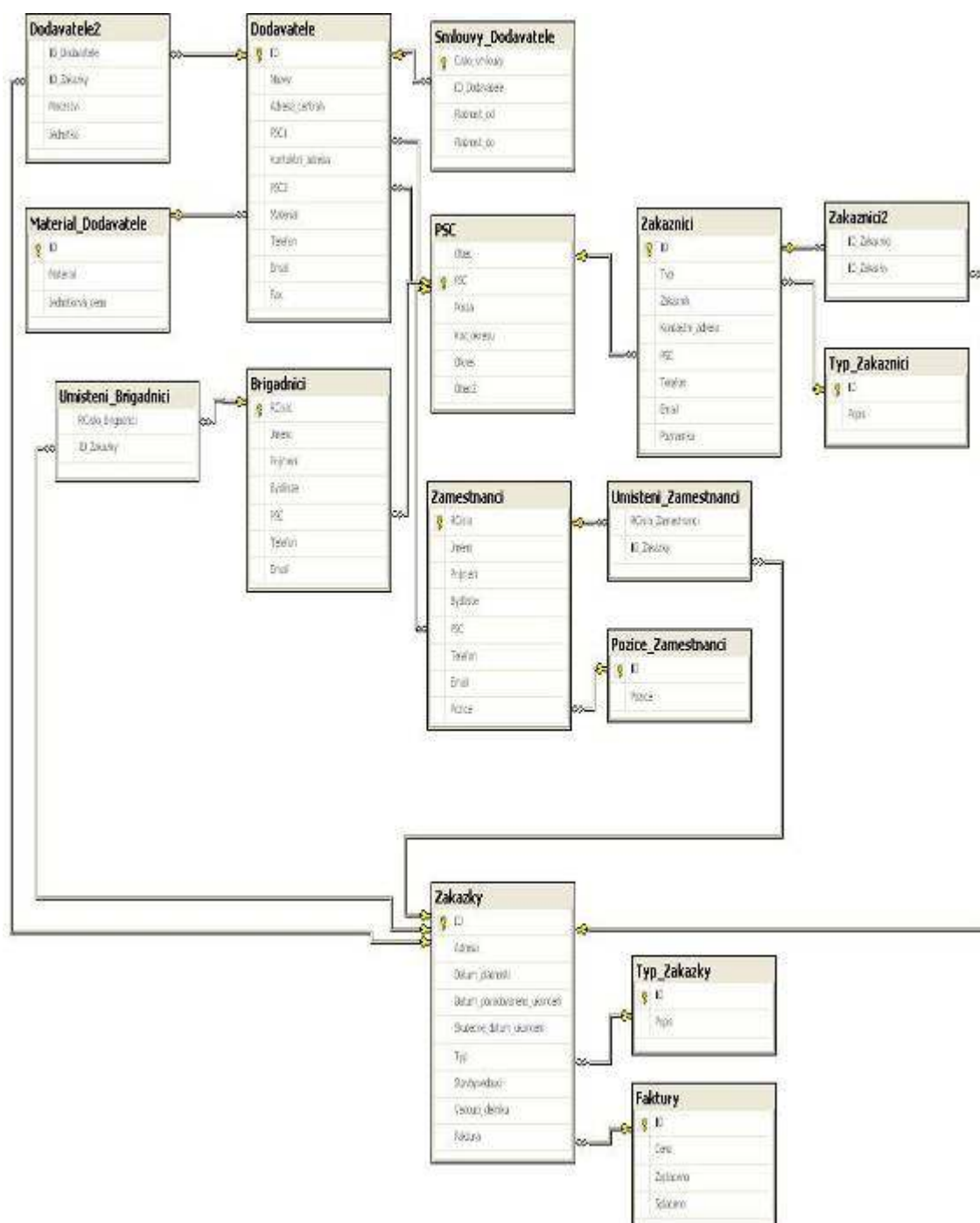
Obrázek č. 6: Datový model



4.2 Schéma databáze na logické úrovni

Při návrhu databázových struktur by po konceptuálním návrhu mělo následovat zobrazení schématu na logické úrovni. Je zde zobrazena vzájemná propojenost tabulek a relačních vazeb. Vidíme zde nejen tabulky hlavní a rozšiřující, ale i číselníky.

Obrázek č. 7: Schéma relačních vazeb databáze



4.3 Datový slovník

4.3.1 Zaměstnanci

Jak již název napovídá tato entita obsahuje veškeré potřebné informace o všech zaměstnancích ve firmě. RCISLO představuje zaměstnancovo rodné číslo sestávající z 11-ti znaků, tedy deseti číslic a lomítka, jež je na sedmé pozici. Pro ověření platnosti se využije platných pravidel pro rodná čísla (součet musí být dělitelný jedenácti). Toto ověření zajistí funkce v jazyce SQL. Bude zde použito více operací. Nejprve odstraníme z řetězce lomítko, následně převedeme na číslo a to podělíme jedenácti. Pokud modulo (zbytek po dělení) bude rovno nule, pak je zadané rodné číslo validní. Počáteční písmeno u položek JMENO, PRIJMENI A BYDLISTE by mělo být velké, což zajistí opět funkce jazyka SQL. PSC je zapsáno bez mezer. TELEFON je vyžadován z důvodu případného rychlého kontaktu pracovníka. Jedinou nepovinnou položkou je EMAIL, jelikož kontakt tímto způsobem je určen jako sekundární. Posledním atributem je POZICE, která je cizím klíčem tabulky POZICE_ZAMESTNANCI, kde číslo představuje označení dané pozice.

Tabulka č. 1: Přehled datových typů a vlastností ZAMĚSTNANCI

Atribut	Datový typ	Délka	Omezení	Poznámka
RCislo	char	11	not null	primární klíč
Jmeno	varchar	20	not null	
Prijmeni	varchar	30	not null	
Bydliste	varchar	60	not null	
PSC	char	5	not null	cizí klíč
Telefon	char	16	not null	
E-mail	varchar	60	null	
Pozice	integer	2	not null	cizí klíč

4.3.2 Pozice Zaměstnanci

Číselník s výčtem typu pracovních pozic. Každému jedinečnému automaticky generovanému číslu je přiřazena jedna pracovní pozice. Propojení s tabulkou zaměstnanců probíhá na základě shodného čísla v poli ID této tabulky a čísla položky POZICE tabulky zaměstnanců.

Tabulka č. 2: Přehled datových typů a vlastností POZICE ZAMĚSTNANCI

Atribut	Datový typ	Délka	Omezení	Poznámka
ID	integer	2	not null	primární klíč
Pozice	varchar	20	not null	

4.3.3 Umístění Zaměstnanců

Znázorňuje jednoduché přiřazení daného zaměstnance identifikovaného rodným číslem ke konkrétní zakázce označenou jedinečnou hodnotou. Na základě rodného čísla mohou získat veškeré potřebné informace o zaměstnanci a dle identifikátoru zakázky všechny její náležitosti. Jednoduše řečeno tabulka říká jaký zaměstnanec pracoval na jaké zakázce. Obě položky jsou sice cizími klíči, ale současně i tvoří složený primární klíč.

Tabulka č. 3: Přehled datových typů a vlastností UMÍSTĚNÍ ZAMĚSTNANCŮ

Atribut	Datový typ	Délka	Omezení	Poznámka
RCislo_Zamestnanci	char	11	not null	cizí klíč
ID_Zakazky	integer	10	not null	cizí klíč

4.3.4 PSČ

Prostý číselník poštovních směrovacích čísel. OBEC značí v jaké obci se místo bydliště nachází. Primárním klíčem je údaj PSC zadávaný bez mezer. Pro účely

poštovních zásilek určení, která POSTA je v jeho blízkosti. OKRES nám dává informaci v jakém okrese se místo nachází.

Tabulka č. 4: Přehled datových typů a vlastností PSČ

Atribut	Datový typ	Délka	Omezení	Poznámka
Obec	varchar	45	null	
PSC	char	5	not null	primární klíč
Posta	varchar	36	null	
Okres	varchar	19	null	

4.3.5 Brigádníci

Jedná se o velmi podobnou tabulku tabulce zaměstnanců s tím rozdílem, že se zde neudává pozice. Jinak zde platí stejná pravidla.

Tabulka č. 5: Přehled datových typů a vlastností BRIGÁDNÍCI

Atribut	Datový typ	Délka	Omezení	Poznámka
RCislo	char	11	not null	primární klíč
Jmeno	varchar	20	not null	
Prijmeni	varchar	30	not null	
Bydliste	varchar	60	not null	
PSC	char	5	not null	cizí klíč
Telefon	char	16	not null	
E-mail	varchar	60	null	

4.3.6 Umístění Brigádníků

Propojovací tabulka brigádníků a zakázek na základě jejich identifikátoru. Podle rodného čísla (RCISLO_BRIGADNICI) lze zjistit veškeré uložené informace o brigádníkovi a dle ID_ZAKAZKY informace o ní. Jednoduše slouží ke zjištění, na které

zakázce jaký brigádník pracoval. Obě položky jsou sice cizími klíči, ale současně i tvoří složený primární klíč. Stejný princip jako u tabulky zaměstnanců.

Tabulka č. 6: Přehled datových typů a vlastností UMÍSTĚNÍ BRIGÁDNÍKŮ

Atribut	Datový typ	Délka	Omezení	Poznámka
RCislo_Brigadnici	char	11	not null	cizí klíč
ID_Zakazky	integer	10	not null	cizí klíč

4.3.7 Dodavatelé

Jelikož není firma výrobním podnikem, veškerý potřebný materiál k uskutečnění svých zakázek získává prostřednictvím smluv s jinými firmami, které tyto materiály přímo vyrábějí a dodávají. ID označuje pouze orientační číslování bez jakéhokoliv klíče pomocí automaticky generovaných hodnot. Do pole NAZEV se zapisuje celý název firmy, tedy i s přívlastkem právní společnosti. Pro případ různých nejasností či vyřízení problémů, které není schopna řešit místní pobočka, slouží ADRESA_CENTRALY, PSC1 poštovní směrovací číslo této adresy. Ovšem ve většině případů mají dodavatelé své centrály zde v Brně a pokud výjimečně sídlí v jiném městě komunikace s touto jednotkou probíhá opravdu v krajních případech. KONTAKTNI_ADRESA značí kontaktní místo v Brně, kde sídlí pobočka či přímo centrála dodavatele, PSC2 příslušné poštovní směrovací číslo. MATERIAL představuje dodávaný materiál, jež je cizím klíčem tabulky MATERIAL_DODAVATELE.

Tabulka č. 7: Přehled datových typů a vlastností DODAVATELÉ

Atribut	Datový typ	Délka	Omezení	Poznámka
ID	integer	2	not null	primární klíč
Nazev	varchar	100	not null	
Adresa_centraly	varchar	60	not null	
PSC1	char	5	not null	cizí klíč
Kontaktني_adresa	varchar	60	not null	

PSC2	char	5	not null	cizí klíč
Material	integer	3	not null	cizí klíč
Telefon	char	16	not null	
E – mail	varchar	60	null	
Fax	varchar	16	null	

4.3.8 Materiál Dodavatelé

Číselník související s předchozí tabulkou dodavatelů propojený atributy materiál z tabulky dodavatelů a ID entity materiál dodavatelé na základě jejich číselné shodnosti.. Hodnoty jsou automaticky generovány (ID). V položce MATERIAL nalezneme o jaký materiál se jedná, jež je předmětem písemných smluv s daným dodavatelem. Poslední atribut JEDNOTKOVA_CENA představuje částku, za kterou je možno odebírat materiál, např. za litr, kilogram, kus apod.

Tabulka č. 8: Přehled datových typů a vlastností MATERIÁL DODAVATELÉ

Atribut	Datový typ	Délka	Omezení	Poznámka
ID	integer	3	not null	primární klíč
Material	varchar	50	not null	
Jednotkova_cena	integer	5	null	

4.3.9 Smlouvy Dodavatelé

Určující konkrétní informace o smlouvě s dodavatelem. Jednoznačnou identifikaci zajišťuje atribut CISLO_SMLOUVY, jež je automaticky generován s přírůstkem o jedničku. Propojení na informace o dodavatelích probíhá pomocí ID_DODAVATELE. Položka PLATNOST_OD představuje datum kdy vstoupila daná smlouva v platnost a s tím související další atribut PLATNOST_DO určující datum vypršení aktuální smlouvy..

Tabulka č. 9: Přehled datových typů a vlastností SMLOUVY DODAVATELÉ

Atribut	Datový typ	Délka	Omezení	Poznámka
Císlo_smlouvy	integer	10	not null	primární klíč
ID_Dodavatele	integer	2	not null	cizí klíč
Platnost_od	date	- -	not null	
Platnost_do	date	- -	not null	

4.3.10 Dodavatelé2

Další z propojovacích tabulek, která nám říká kteří dodavatelé dodali materiál na jakou zakázku. ID_DODAVATELE představuje identifikaci dodavatele a ID_ZAKAZKY označuje konkrétní zakázku. Na základě těchto dvou položek lze zjistit potřebné informace jak o dodavateli, tak o zakázce. Když už dodavatel dodá materiál, je potřeba evidovat jaké množství, což je zachyceno položkou MNOZSTVI. JEDNOTKA pouze označuje v jakých jednotkách je materiál dodáván, např. litr, kus, metr apod.

Tabulka č. 10: Přehled datových typů a vlastností DODAVATELÉ2

Atribut	Datový typ	Délka	Omezení	Poznámka
ID_Dodavatele	integer	2	not null	cizí klíč
ID_Zakazky	integer	10	not null	cizí klíč
Mnozstvi	integer	5	null	
Jednotka	varchar	5	null	

4.3.11 Zákazníci

Zde jsou zaznamenány informace o zákaznících ať dřívějších zakázek tak právě probíhajících. ID označuje zákazníka. V případě, že se jedná o osobu je uvedeno rodné číslo osoby, se kterou se vedou jednání. Pokud se jedná o firmu, запиše se její IČO (identifikační číslo organizace). Cizím klíčem k tabulce TYP_ZAKAZNICI je atribut

TYP, jež značí, zda-li se jedná o zákazníka fyzické povahy (např. soukromou osobu) či firmu. ZÁKAZNÍK již označuje konkrétní identifikaci zákazníka. V případě fyzické osoby její jméno a příjmení, v případě firmy celý její název včetně typu společnosti. KONTAKTNI_ADRESA představuje trvalé bydliště zákazníka, jelikož může samozřejmě nastat situace, že místo stavby se nachází na jiném místě než zákazník bydlí. Posledním atributem POZNAMKA jsou rozuměny všelijaké připomínky, kupříkladu “neplatí včas“, “ velmi seriózní jednání“ apod.

Tabulka č. 11: Přehled datových typů a vlastností ZÁKAZNÍCI

Atribut	Datový typ	Délka	Omezení	Poznámka
ID	varchar	11	not null	primární klíč
Typ	integer	1	not null	cizí klíč
Zakaznik	varchar	100	not null	
Kontaktni_adresa	varchar	60	not null	
PSC	char	5	null null	cizí klíč
Telefon	char	16	not null	
E – mail	varchar	60	null	
Poznamka	varchar	100	null	

4.3.12 Typ Zákazníci

Další číselník, tentokrát k tabulce zákazníků. Popis jejich atributů je podrobně popsán výše, ale pro přehlednost jejich význam uvedu znova. OZNACENI automaticky generované číslo s přírůstkem o jedničku. V atributu POPIS je vepsáno konkrétní pojmenování typu zákazníka..

Tabulka č. 12: Přehled datových typů a vlastností TYP ZÁKAZNÍCI

Atribut	Datový typ	Délka	Omezení	Poznámka
Oznaceni	integer	2	not null	primární klíč
Popis	char	5	not null	

4.3.13 Zákazníci2

Jednoznačná identifikace k sobě patřící zakázky a jí odpovídající zákazník. Názvy polí dostatečně vystihují jejich význam. Oba dva atributy představují i složený primární klíč.

Tabulka č. 13: Přehled datových typů a vlastností ZÁKAZNÍCI2

Atribut	Datový typ	Délka	Omezení	Poznámka
ID_Zakaznici	varchar	11	not null	cizí klíč
ID_Zakazky	integer	10	not null	cizí klíč

4.3.14 Faktury

Již ze začátku kapitoly bylo řečeno, že jsem nechtěl do tohoto návrhu moc připlétat problematiku finančních záležitostí, přesto si myslím, že tabulka faktur je nezbytnou součástí k celému návrhu, i když je jen jednoduchou formou jejich řízení. Automaticky generované identifikační číslo zahrnuje ID. Velmi důležitá je CENA udávající údaj celkové ceny kompletní zakázky. Záměrně je povolena její nulová hodnota z důvodu možné změny ceny. ZAPLACENO představuje částku, která již byla zaplacená z celkové. Pomocí agregačních funkcí lze zjistit jaká finanční částka zbývá k zaplacení. SPLACENO smí nabývat pouze dvou logických hodnot a to logické jedničky a nuly. Nula značí, že ještě nebylo zaplacené kompletně a jednička kompletní splacení veškerých financí za zakázku zákazníkem. Za pomoci jazyka SQL lze docílit efektu, kdy při stejné hodnotě polí cena a zaplacené nabude splaceno hodnoty jedna a naopak.

Tabulka č. 14: Přehled datových typů a vlastností FAKTURY

Atribut	Datový typ	Délka	Omezení	Poznámka
ID	integer	10	not null	primární klíč
Cena	integer	12	null	
Zaplaceno	integer	12	null	
Splaceno	bit	- -	not null	

4.3.15 Zakázky

Asi stěžejní tabulka celé aplikace. Prolínají se zde jména zaměstnanců, vedení a zákazníků. Jsou tu uloženy i velmi důležité údaje ohledně finančních záležitostí. ID představuje automaticky generované číslo pro snadnou identifikaci. ADRESA nám uchovává informaci o ulici, kde zakázka probíhá. DATUM_PLATNOSTI představuje datum, kdy došlo k podpisu smlouvy, respektive, kdy smlouva vstupuje v platnost. DATUM_POZADOVANEHO_UKONCENI značí datum do kdy má být dle smlouvy zakázka dokončena. SKUTECNE_DATUM_UKONCENI slouží pouze pro statistické účely firmy, kdy budou sledovány odchylky od původně smluveného termínu. Samozřejmě jsou zapisována i data s dřívějším dokončením. TYP je cizím klíčem, tentokrát tabulky TYP_ZAKAZKY. Položka představuje jednoduché označení, jednalo-li se o rodinný dům, panelový dům či jiný druh budovy. Další atribut jež je STAVBYVEDOUCI obsahuje informaci o tom kdo je za zakázku zodpovědný. Možná jsou pouze tři jména (dva jednatelé a jeden společník). Pokud není stavbyvedoucí přítomen měl by být ustanoven někdo kdo v takovéto situaci zastane vedoucí pozici. Od toho jsou VEDOUCI_DELNIKU, jež jsou též zapisováni formou rodného čísla. Poslední jest FAKTURA, udávající číslo faktury spojené s danou zakázkou.

Tabulka č. 15: Přehled datových typů a vlastností ZAKÁZKY

Atribut	Datový typ	Délka	Omezení	Poznámka
ID	integer	10	not null	primární klíč
Adresa	varchar	100	not null	
Datum_platnosti	date	- -	not null	
Datum_pozadovaneho_ukoncení	date	- -	null	
Skutecne_datum_ukoncení	date	- -	null	
Typ	integer	1	not null	cizí klíč
Stavbyvedouci	char	11	not null	
Vedouci_delniku	char	11	not null	
Faktura	integer	10	not null	cizí klíč

4.3.16 Typ Zakázky

Poslední číselník označující typ objektu, na kterém je zakázka prováděna. Prozatím nabývá pouze dvou položek a to rodinný dům a panelový dům.

Tabulka č. 16: Přehled datových typů a vlastností TYP ZAKÁZKY

Atribut	Datový typ	Délka	Omezení	Poznámka
Oznaceni	integer	1	not null	primární klíč
Popis	varchar	15	not null	

4.4 Výběr platformy

Jelikož databázových serverů je v současnosti vcelku velké množství, byla stanovena kritéria, podle kterých se nakonec vybral „vítěz“.

Kritéria pro výběr:

- Co nejnižší pořizovací náklady
- Co nejnižší průběžné výdaje na produkt
- Co nejméně místa na disku
- Co nejmenší HW nároky
- Jednoduchost, přehlednost, mocné nástroje (pro administrátora)

Na základě těchto požadavků byl vybrán produkt společnosti Microsoft **SQL SERVER 2005 EXPRESS EDITION**. Pro porovnání uvádím jakým způsobem odpovídá požadavkům.

Požadavky SQL SERVERu 2005:

- Zdarma
- 0 Kč
- Přibližně 200 MB

- Procesor - Pentium III 600 MHz (1 GHz doporučeno)
Paměť - 256 MB RAM (512 MB doporučeno)
Operační systém – Windows XP, Vista, příp. Server 2000 a novější
- Grafické prostředí, jednoduché ovládání, spousta užitečných funkcí
(trigger, indexace, přístupová práva atd.)

Po srovnání požadavků s tím co nabízí SQL SERVER je, podle mého názoru, velmi vhodnou volbou jak pro řešení této práce, tak reálné nasazení do každodenního běhu firmy.

5 PŘÍNOS (EFEKTIVNOST) NÁVRHŮ ŘEŠENÍ

Implementace navržené databázové struktury by měla především přinést zefektivnění, zrychlení a zpřehlednění každodenních běžných úkonů, se kterými se firma setkává. Navrhovaná část neobsahuje prvky zabývající se problematikou účetnictví a finančních záležitostí obecně, jelikož již je používán kvalitní software pro tuto práci určený. Nicméně v případě úspěšného nasazení mého návrhu se tyto dva moduly dají spojit v jeden ještě mocnější nástroj s vzájemnou propojeností a komunikací. Zmiňovanou efektivitu by mělo tvořit snadné a především rychlé vyhledávání potřebných údajů. Identifikace pracovníků či brigádníků by měla jistě také přispět k efektivitě.

Na základě relačních vazeb a chytrého propojení lze ihned získat potřebné informace například o zakázce, s tím související dodavatelé, jež dodali materiál k této zakázce, dále například informaci o smlouvě s tímto dodavatelem, nesmím opomenout ani důležitý poznatek k jaké zakázce byl zaměstnanec či brigádník přiřazen a v neposlední řadě i informace o zákazníkovi, který si zakázku objednal.

Pro kvalitní splnění požadavků firmy, především na co nejnižší náklady, a snadná orientace pro administrátora, tedy mne, byl vybrán produkt od společnosti Microsoft a to SQL Server 2005 Express Edition. Obecně se sice jedná o komerční produkt, ale tahle edice je distribuována jako volně ke stažení, disponuje spousty užitečnými nástroji a může se pochlubit, dle mého názoru, pěkným grafickým uživatelským prostředím.

ZÁVĚR

V úvodních kapitolách je nastíněn hlavní problém a s tím související cíl této práce. Uveden je i úkol, který je nad rámec zadání bakalářské práce, přesto byl vypracován z důvodu úzké souvislosti s prací. Je zde představena firma ADAPTA s.r.o., kde nechybí stručná SWOT analýza a především rozbor technického vybavení společnosti.

Po úvodní části následuje část teoretická. Ta se snaží o zavedení do dané problematiky postupnými kroky od seznámení se s pojmem databáze, stručnou historií a databázovým systémem přes podrobný popis systému řízení báze dat (SŘBD), jeho rozdělení architektury až po výběr databázového serveru s možností komerčního využití či open source programy. Neméně důležité jsou i podkapitoly týkající se datového a funkčního modelování zabývající se konkrétními pravidly pro správné vytvoření databázové aplikace se zaměřením na normalizace a odstranění redundance dat.

Nejpodstatnější kapitolou jsou návrhy na řešení situace popsané v úvodní části. Zde se věnuji vytvoření a především popisu databázových tabulek, jejich datových struktur, které jsou tvořeny dle požadavků vedení společnosti a vzájemné integrity dat.

Ona předpokládaná zvýšená efektivnost práce by měla být zajištěna především pomocí primárních klíčů tvořících rodná čísla u osob a jedinečný unikátní identifikátor označující firmy (dodavatele, zakázky apod.) zvyšující se při každém novém vložení záznamu o jedničku. To vše je obsaženo v poslední kapitole zabývající se zhodnocením a shrnutím vlastních návrhů práce.

Z důvodu požadavku minimalizace nákladů jsem se orientoval na výběr open source produktu. Nakonec byl vybrán od společnosti Microsoft a to SQL Server 2005 Express Edition, jež se sice řadí ke komerčnímu využití, ovšem tato edice je distribuována zcela zdarma bez velkých omezení, tedy alespoň pro využití firmou postačujících

Seznam obrázků a tabulek

OBRÁZEK Č. 1: KONCEPTUÁLNÍ NÁVRH DATABÁZE	29
OBRÁZEK Č. 2: DEKOMPOZICE VAZBY ZAMĚSTNANCI / ZAKÁZKY	29
OBRÁZEK Č. 3: DEKOMPOZICE VAZBY BRIGÁDNÍCI / ZAKÁZKY	30
OBRÁZEK Č. 4: DEKOMPOZICE VAZBY DODAVATELÉ / ZAKÁZKY	30
OBRÁZEK Č. 5: DEKOMPOZICE VAZBY ZÁKAZNÍCI / ZAKÁZKY	30
OBRÁZEK Č. 6: DATOVÝ MODEL	31
OBRÁZEK Č. 7: SCHÉMA RELAČNÍCH VAZEB DATABÁZE	32

TABULKA Č. 1: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ ZAMĚSTNANCI	33
TABULKA Č. 2: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ POZICE ZAMĚSTNANCI	34
TABULKA Č. 3: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ UMÍSTĚNÍ ZAMĚSTNANCŮ	34
TABULKA Č. 4: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ PSČ	35
TABULKA Č. 5: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ BRIGÁDNÍCI	35
TABULKA Č. 6: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ UMÍSTĚNÍ BRIGÁDNÍKŮ	36
TABULKA Č. 7: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ DODAVATELÉ	36
TABULKA Č. 8: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ MATERIÁL DODAVATELÉ	37
TABULKA Č. 9: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ SMLOUVY DODAVATELÉ	38
TABULKA Č. 10: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ DODAVATELÉ2	38
TABULKA Č. 11: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ ZÁKAZNÍCI	39
TABULKA Č. 12: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ TYP ZÁKAZNÍCI	39
TABULKA Č. 13: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ ZÁKAZNÍCI2	40
TABULKA Č. 14: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ FAKTURY	40
TABULKA Č. 15: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ ZAKÁZKY	41
TABULKA Č. 16: PŘEHLED DATOVÝCH TYPŮ A VLASTNOSTÍ TYP ZAKÁZKY	42

SEZNAM POUŽITÉ LITERATURY

Písemné zdroje publikované

- [1] BRÁZA, J. *PHP 4 Praktické příklady*. Praha: Grada Publishing, 2003. 224 s. ISBN 80-247-0441-2.
- [2] HERNANDEZ, M.J. *Návrh databází*. Praha: Grada Publishing, 2005. 408 s. ISBN 80-247-0900-7.
- [3] KOCH, M. *Datové a funkční modelování*. Brno: Akademické nakladatelství CERM, s.r.o., 2004. 108 s. ISBN 80-214-2724-8.
- [4] KREJČÍ, L. *PHP Kapesní přehled*. Brno: Computer Press, 2006. 108 s. ISBN 80-251-0808-2.
- [5] KŘÍŽ, J. a DOSTÁL, P. *Databázové systémy*. Brno: Akademické nakladatelství CERM, s.r.o., 2005. 111 s. ISBN 80-214-3064-8.
- [6] LACKO, L. *PHP a MySQL Hotová řešení*. Brno: Computer Press, 2005. 304 s. ISBN 80-251-0397-8.
- [7] LACKO, L. *SQL Hotová řešení*. Brno: Computer Press, 2003. 300 s. ISBN 80-7226-975-5.
- [8] LACKO, L. *SQL Kapesní přehled*. Brno: Computer Press, 2005. 96 s. ISBN 80-251-0788-4.
- [9] LACKO, L. *Web a databáze*. Praha: Computer Press, 2001. 252 s. ISBN 80-7226-555-5.
- [10] OPPEL, A. *Databáze bez předchozích znalostí*. Brno: Computer Press, 2006. 320 s. ISBN 80-251-1199-7.
- [11] RIORDAN, R.M. *Vytváříme relační databázové aplikace*. Brno: Computer Press, 2000. 294 s. ISBN 80-7226-360-9.
- [12] STEPHENS, K.R. a PLEW, R.R. *Naučte se SQL za 21 dní*. Brno: Computer Press, 2004. 584 s. ISBN 80-7226-870-8.
- [13] ULLMAN, L. *PHP a MySQL*. Brno: Computer Press, 2004. 536 s. ISBN 80-251-0063-4.

Internetové adresy

- [14] [online] Dostupné z <http://www.root.cz/clanky/historie-relacnich-databazi/>
- [15] [online] Dostupné z <http://www.fi.muni.cz/usr/jkucera/pv109/2004/xbukovsk.htm>
- [16] [online] Dostupné z <http://interval.cz/clanky/database-a-jazyk-sql/>

Seznam použitých zkratk

SWOT – analýza hodnotící silné (Strengths) a slabé (Weaknesses) stránky, příležitosti (Opportunities) a hrozby (Threats), jež jsou spojené s určitým projektem, typem podnikání

SQL – Structured Query Language – strukturovaný dotazovací jazyk, který se používá pro práci s daty v databázích

.

LAN – Local Area Network – malá lokální síť např. v rámci jednoho podniku.

SŘBD – Systém řízení báze dat – softwarové vybavení zajišťující práci s databází, tvoří rozhraní mezi aplikačními programy a uloženými daty

PHP – Hypertext Preprocessor – skriptovací programovací jazyk, především pro tvorbu dynamických internetových stránek

INTRANET – prostředí přístupné pouze firmě nikoliv někomu jinému „z venčí“

PŘÍLOHY

Seznam příloh

PŘÍLOHA Č. 1: ZDROJOVÉ KÓDY VYTVOŘENÍ TABULEK	51
PŘÍLOHA Č. 2: UŽIVATELSKÉ PROSTŘEDÍ INTRANETU	56
PŘÍLOHA Č. 3: VYBRANÉ POHLEDY, SPOUŠTĚ A PROCEDURY	58

Příloha č. 1: Zdrojové kódy vytvoření tabulek

```
CREATE DATABASE ADAPTA;
```

```
go
```

```
use ADAPTA;
```

```
CREATE TABLE Pozice_Zamestnanci (
    ID integer identity(1,1) primary key,
    Pozice varchar(20) not null
);
```

```
CREATE TABLE PSC (
    Obec varchar(45),
    PSC char(5) primary key not null,
    Posta varchar(36),
    Okres varchar(19),
);
```

```
CREATE TABLE Zamestnanci (
    RCislo char(11) primary key,
    Jmeno varchar(20) not null,
    Prijmeni varchar(30) not null,
    Bydliste varchar(60) not null,
    PSC char(5) not null,
    Telefon char(16) not null,
    Email varchar(60),
    Pozice integer not null
    FOREIGN KEY (PSC)
    REFERENCES PSC(PSC),
    FOREIGN KEY (Pozice)
    REFERENCES Pozice_Zamestnanci(ID)
);
```

```

CREATE TABLE Brigadnici (
    RCislo char(11) primary key,
    Jmeno varchar(20) not null,
    Prijmeni varchar(30) not null,
    Bydliste varchar(60) not null,
    PSC char(5) not null,
    Telefon char(16) not null,
    Email varchar(60)
    FOREIGN KEY (PSC)
    REFERENCES PSC(PSC)
);

CREATE TABLE Typ_Zakaznici(
    ID integer identity(1,1) primary key,
    Popis char(5) not null
);

CREATE TABLE Zakaznici (
    ID varchar(11) primary key,
    Typ integer not null,
    Zakaznik varchar(100) not null,
    Kontaktni_adresa varchar(60) not null,
    PSC char(5) not null,
    Telefon char(16) not null,
    Email varchar(60),
    Poznamka varchar(100)
    FOREIGN KEY (Typ)
    REFERENCES Typ_Zakaznici(ID),
    FOREIGN KEY (PSC)
    REFERENCES PSC(PSC)
);

```

```
CREATE TABLE Material_Dodavatele(
    ID integer identity(1,1) primary key,
    Material varchar(50),
    Jednotkova_cena integer
);
```

```
CREATE TABLE Dodavatele (
    ID integer identity (1,1) primary key,
    Nazev varchar(100) not null,
    Adresa_centraly varchar(60) not null,
    PSC1 char(5) not null,
    Kontaktni_adresa varchar(60) not null,
    PSC2 char(5) not null,
    Material integer not null,
    Telefon char(16) not null,
    Email varchar(60),
    Fax varchar(16)
    FOREIGN KEY (PSC1)
    REFERENCES PSC(PSC),
    FOREIGN KEY (PSC2)
    REFERENCES PSC(PSC),
    FOREIGN KEY(Material)
    REFERENCES Material_Dodavatele(ID),
);
```

```
CREATE TABLE Typ_Zakazky (
    ID integer identity(1,1) primary key,
    Popis varchar(15) not null
);
```

```
CREATE TABLE Faktury (
    ID integer identity(1,1) primary key,
    Cena integer,
    Zaplaceno integer,
    Splaceno bit
);
```

```

CREATE TABLE Zakazky      (
    ID integer identity(1,1) primary key,
    Adresa varchar(100) not null,
    Datum_platnosti datetime not null,
    Datum_pozadovaneho_ukonceni datetime,
    Skutecne_datum_ukonceni datetime,
    Typ integer not null,
    Stavbyvedouci char(11) not null,
    Vedouci_delniku char(11) not null,
    Faktura integer not null
    FOREIGN KEY (Typ)
    REFERENCES Typ_Zakazky(ID),
    FOREIGN KEY (Faktura)
    REFERENCES Faktury(ID)
);

```

```

CREATE TABLE Smlouvy_Dodavatele(
    Cislo_smlouvy integer identity(1,1)
    primary key,
    ID_Dodavatele integer not null,
    Platnost_od datetime not null,
    Platnost_do datetime not null
    FOREIGN KEY (ID_Dodavatele)
    REFERENCES Dodavatele(ID)
);

```

```

CREATE TABLE Umisteni_Zamestnanci (
    RCislo_Zamestnanci char(11) not
    null,
    ID_Zakazky integer not null
    FOREIGN KEY (RCislo_Zamestnanci)
    REFERENCES Zamestnanci(RCislo),
    FOREIGN KEY (ID_Zakazky)
    REFERENCES Zakazky(ID)
);

```

```

CREATE TABLE Umisteni_Brigadnici(
                                RCislo_Brigadnici char(11) not null,
                                ID_Zakazky integer not null
                                FOREIGN KEY (RCislo_Brigadnici)
                                REFERENCES Brigadnici(RCislo),
                                FOREIGN KEY (ID_Zakazky)
                                REFERENCES Zakazky(ID)
                                );

```

```

CREATE TABLE Dodavatele2(
                                ID_Dodavatele integer,
                                ID_Zakazky integer,
                                Mnozstvi integer,
                                Jednotka varchar(5)
                                FOREIGN KEY (ID_Dodavatele)
                                REFERENCES Dodavatele(ID),
                                FOREIGN KEY (ID_Zakazky)
                                REFERENCES Zakazky(ID)
                                );

```

```

CREATE TABLE Zakaznici2(
                                ID_Zakaznici varchar(11) not null,
                                ID_Zakazky integer not null
                                FOREIGN KEY (ID_Zakaznici)
                                REFERENCES Zakaznici(ID),
                                FOREIGN KEY (ID_Zakazky)
                                REFERENCES Zakazky(ID)
                                );

```

Příloha č. 2: Uživatelské prostředí intranetu

Hlavní stránka aplikace



Rozbalení nabídky menu



Výběr položky menu -> zobrazení formuláře

The screenshot displays a web browser window with the URL `http://localhost/adapta/zamestnanci/novy.html`. The page features a header with the ADAPTA logo and the word 'intranet'. On the left, a sidebar menu lists various categories: Zaměstnanci, Brigádníci, Dodavatelé, Zakázky, Zakazníci, and Oddělení. The 'Zaměstnanci' category is expanded, showing sub-options: Nový, Upravit, Novozat, Seznam, and Hledat. The main content area is titled 'ZAMĚSTNANCI >> Nový' and contains a form for adding a new employee. The form fields include: Rodné číslo (with a dropdown for gender), Jméno, Příjmení, Bydliště, PSČ, Telefon (with a country code dropdown), E-mail, and Povol (with a dropdown for position). A 'Vložit' button is located at the bottom of the form.

Zaměstnanci

- Nový
- Upravit
- Novozat
- Seznam
- Hledat

Brigádníci

Dodavatelé

Zakázky

Zakazníci

Oddělení

ZAMĚSTNANCI >> Nový

Rodné číslo:

Jméno:

Příjmení:

Bydliště:

PSČ:

Telefon: +420

E-mail:

Povol:

Příloha č. 3: Vybrané pohledy, spouště a procedury

```
create view PStavbyvedouci as
select RCislo,Jmeno,Prijmeni,Pozice
from Zamestnanci
where Zamestnanci.Pozice = 1 or Zamestnanci.Pozice = 2;
```

```
create view Zbyva_zaplatit as select *,(Cena - Zaplaceno)
as Zbyva
from Faktury;
```

```
create view Cena_za_material as
select m.ID,m.Material,m.Jednotkova_cena, d.Mnozstvi,
(Jednotkova_cena * Mnozstvi) as celkem
from Material_Dodavatele m
inner join Dodavatele2 d on d.ID_Dodavatele = m.ID;
```

```
create trigger Kontrola_Rcisla
on Zamestnanci
for insert as
begin
select *,cast(replace(RCislo,'/','') as integer) % 11
as konverze from zamestnanci
go
if exists (select RCislo from Zamestnanci)
print 'Zaměstnanec s daným rodným číslem již existuje'
end
go
```

```
create procedure Hledani
@RCislo char(11),@prijmeni char(30) output
as
    select Prijmeni from Zamestnanci where RCislo = @RCislo
go
    declare @vracene_prijmeni char(30)
execute Hledani '590915/6673',@vracene_prijmeni output
print @vracene_prijmeni
go
```