

MORAVSKÁ VYSOKÁ ŠKOLA OLOMOUC
ÚSTAV INFORMATIKY

Petr Nesvadba

**Využití nástrojů MS Office k vytvoření databáze hardwaru
ve firmě ExlService Czech Republic, s.r.o.**

Utilizing MS Office Tools for Creating a Hardware Database
in the Company ExlService Czech Republic, Ltd.

Bakalářská práce

Vedoucí práce: Mgr. Květoslav Bártek, Ph.D.

Olomouc 2010

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a použil jen uvedené informační zdroje.

Olomouc 16. dubna 2010

Petr Nesvadba

Děkuji Mgr. Květoslavu Bártkovi, Ph.D. za odborné vedení bakalářské práce.

Obsah

Úvod.....	6
1 Historie ukládání a organizování dat.....	7
2 Funkce a služby, výhody a nevýhody současných systémů DBMS.....	10
3 Relační model.....	12
3.1 Strukturální část relačního modelu	12
3.2 Manipulační část relačního modelu	13
3.3 Pravidla pro zajištění integrity dat.....	13
3.4 Relační databázové jazyky.....	14
3.4.1 Relační databázový jazyk SQL	14
3.4.2 Relační databázový jazyk QBE.....	15
4 Životní cyklus vývoje databázového systému.....	16
5 MS Access a MS SQL Server jako příklady implementace relačního modelu.....	21
5.1 MS Office Access	21
5.1.1 Výhody a nevýhody MS Accessu	21
5.1.2 Topologie konfigurací MS Accessu.....	22
5.2 MS SQL Server.....	25
5.2.1 Výhody a nevýhody MS SQL Serveru.....	25
5.3 Srovnání vlastností MS Accessu a MS SQL Serveru	26
5.4 Topologie MS Access a MS SQL Server	27
6 Představení společnosti EXL.....	29
7 Informační systém pobočky ExlService Czech Republic	30
8 Vytvoření databáze hardwaru.....	31
8.1 Plánování databáze	31
8.1.1 Poslání databáze	31
8.1.2 Dílčí cíle	31
8.2 Definice systému.....	32
8.3 Sběr a analýza požadavků.....	33
8.4 Konceptuální návrh databáze.....	36
8.5 Logický návrh databáze	37

8.6	Výběr DBMS a aplikace pro vytvoření uživ. rozhraní a aplikační logiky	38
8.7	Fyzický návrh databáze a jeho implementace	39
8.8	Testování databáze.....	44
8.9	Přechod do ostrého provozu	44
8.10	Provozní údržba	44
9	Omezení a rizika zvoleného řešení a doporučení pro další rozvoj.....	46
	Závěr	47
	Anotace	48
	Literatura a prameny	49
	Seznam zkratk	50
	Seznam obrázků.....	51
	Seznam tabulek	52

Úvod

Dané téma bakalářské práce jsem si zvolil s cílem využít znalosti získané studiem na Moravské vysoké škole ve svém současném zaměstnání, kde pracuji jako systémový analytik v oddělení informačních technologií, a ve kterém vyvstala potřeba řešit problematiku evidence, inventarizace a získávání informací o veškerém hardwaru firmy. K řešení daného problému bych chtěl využít nástroje MS Office.

Cílem bakalářské práce je tedy popsat a využít možnosti systému MS Office pro vytvoření databáze hardwaru ve firmě ExlService Czech Republic, s.r.o. v kontextu současných databázových technologií. Základním předpokladem je, že tyto prostředky jsou vhodné k dosažení stanoveného cíle. Východiskem k vytvoření tohoto nástroje budou kromě znalostí získaných studiem na Moravské vysoké škole také teoretické poznatky získané z informačních zdrojů.

V teoretické části bych chtěl popsat vývoj databázových systémů, příklady těchto systémů včetně MS Access a životní cyklus vývoje databázových aplikací.

V praktické části bych chtěl aplikaci vytvořit a na základě jejího používání v praxi zhodnotit, zda byly zvolené nástroje pro její vytvoření vhodné, případně doporučit směr dalšího rozvoje aplikace.

1 Historie ukládání a organizování dat

Potřeba ukládání a organizování dat sahá do dávné minulosti, do doby, kdy si lidé začali uvědomovat, že data, informace a znalosti je třeba uchovávat, a mít možnost je v případě potřeby jednoduchým způsobem vyhledat. Slovo databáze se sice začalo používat až v éře počítačů, v obecném slova smyslu bychom jej však mohli definovat jako jakýkoli soubor uspořádaných dat.¹ V dřívější době sloužily k ukládání a organizaci dat především papírové databáze ve formě zatříděných papírových dokumentů nebo papírových kartoték. Některé z nich se v papírové formě používají dodnes. Příkladem mohou být dokumenty zatříděné do šanonů nebo donedávna používané tematicky nebo abecedně řazené listkové kartotéky v knihovnách.

S rozvojem počítačů a počítačových databází se ukládání a získávání informací ve velké míře zefektivnilo, což s sebou však přineslo ukládání stále většího množství dat a nutnost databázové technologie neustále zdokonalovat.

Předchůdcem dnešních databázových systémů byly souborově orientované systémy, které se začaly využívat v 60. letech 20. století, a které používaly index-sekvenční metodu přístupu (Indexed Sequential Access Method, ISAM). Obvykle se v nich pro každou tabulku používal jeden samostatný soubor. Příkladem tohoto typu databáze jsou např. dBase, FoxPro nebo Paradox. Souborově orientované databáze však s sebou přinášely řadu nevýhod mezi něž patřily především:

- Duplikace dat - typicky každé oddělení mělo svá vlastní data a aplikační programy, takže docházelo ke značné duplikaci dat a s tím spojeným vyšším nákladům na úložný prostor pro data, k vyšší časové náročnosti a chybovosti při přepisování dat, k neslučitelnosti formátu dat mezi odděleními, atd.
- Programově-datová závislost – zpracovatelské rutiny byly součástí těchto programů, takže při každé změně struktury dat musely být všechny tyto programy změněny.
- Omezené sdílení dat – sdílení dat mezi odděleními bylo díky decentralizovanému zpracování problematické, nebylo možné získávat informace na základě okamžité potřeby (ad hoc).

¹ Srov. VIEIRA, R., *SQL Server 2000 - Programujeme profesionálně*, s. 2.

Dalším vývojovým stupněm byl hierarchický a síťový model představený v polovině 60. let minulého století. Tyto modely představovaly 1. generaci DBMS (systému řízení databáze). Jejich hlavní nevýhodou bylo, že stále přetrvávala programově-datová závislost a bylo potřeba vytvářet složité programy pro zodpovězení i jednoduchých dotazů.

V roce 1970 byl představen relační model, který všechna data reprezentoval ve formě tabulek. Na základě tohoto modelu vznikl strukturovaný dotazovací jazyk SQL a relační databázové systémy (RDBMS) jako např. Oracle, Microsoft SQL Server a IBM DB2. Tyto relační databázové systémy se označují jako DBMS druhé generace.

Časový rámec	Vývoj	Komentář
Od 60. let	Souborově orientované systémy	Předchůdce databázových systémů, decentralizovaný přístup: každé oddělení uchovává a kontroluje vlastní data
Polovina 60. let	Hierarchický a síťový model dat	Představují 1. generaci DBMS. Hlavním hierarchickým systémem je IMS od firmy IBM, hlavním síťovým systémem IDMS/R od firmy Computer Associated. Nedostatek nezávislosti dat a nutnost vyvíjet složité programy pro zpracování dat.
1970	Návrh relačního modelu	Zveřejněno pojednání E.F.Codda „A relational model of data for large shared data banks“ (Relační model pro velké sdílené databanky), které reagovalo na slabiny systémů 1. generace
70. léta	Vyvíjí se prototypy RDBMS	Během tohoto období se objevily dva hlavní prototypy: projekt INGRES na Kalifornské univerzitě v Berkeley (počátkem roku 1970) a project System R v IBM Research Laboratory v San José v Kalifornii (počátkem roku 1974), který vedl k vývoji SQL

Tab. 1 – Historický vývoj databázových systémů²

V roce 1976 byl představen entitně-relační (ER) model, který vylepšuje modelovací schopnost relačního modelu, a který je v současné době široce přijímanou technikou pro návrh databází.

² CONOLLY, T., BEGG, C., a HOLOWCZAK, R., *Mistrovství - databáze - Profesionální průvodce tvorbou efektivních databází.*, s. 49.

Časový rámec	Vývoj	Komentář
1976	Návrh ER modelu	Zveřejnění pojednání „The Entity Relationship Model Toward a Unified View of Data“ (Entitně relační model k sjednocenému pohledu na data) od Chena. ER modelování se stalo podstatnou součástí metodologie návrhu databází.
1979	Objevily se komerční RDBMS	Objevily se komerční RDBMS jako Oracle, INGRES a DB2. Ty představují 2. generaci DBMS
1987	Standard ISO SQL	ISO (americká organizace pro standardizaci) standardizovala SQL, existují také následné verze standardu 1989, 1992 (SQL2), 1999 (SQL: 1999) a 2003 (SQL: 2003)
90. léta	Objevují se OODBMS a ORDBMS	V tomto období vznikl nejprve OODBMS a později ORDBMS (Oracle 8 s objektovými vlastnostmi byl uvolněn v roce 1997).
90. léta	Datové sklady	V tomto období začali také dodavatelé DBMS uvolňovat systémy skladování dat a poté produkty zaměřené na data mining.
Polovina 90. let	Integrace web-databáze	Objevily se první aplikace databází na internetu. Dodavatelé DBMS a třetí strany rozpoznali důležitost Internetu a podpory integrace databází a webu.
1998	XML	W3C schválilo XML 1.0. Došlo k integraci XML s databázovými produkty a vyvíjí se nativní XML databáze.

Tab. 1 - pokračování

Jako odpověď na vzrůstající složitost databázových aplikací se v 90. letech 20. století objevily 2 nové databázové systémy – objektově orientovaný DBMS (OODBMS) a objektově-relační DBMS (ORDBMS), které jsou označovány jako DBMS třetí generace. V této době byly představeny také systémy na skladování dat (data warehousing) a dolování dat (data mining), které poskytují rozsáhlé možnosti analýzy dat z různých zdrojů, a které je možné využívat např. při strategickém rozhodování díky zobrazování trendů, atd.

V polovině 90. let se objevily také první aplikace databází na internetu. Na konci 90. let dochází k integraci XML s databázovými produkty a k vývoji nativních XML databází.³

³ Srov. CONOLLY, T., BEGG, C., a HOLOWCZAK, R., *Mistrovství - databáze - Profesionální průvodce tvorbou efektivních databází.*, s. 44-50.

2 Funkce a služby, výhody a nevýhody současných systémů DBMS

System řízení databáze (DBMS) je softwarový systém, který uživateli umožňuje definovat, vytvářet a udržovat databázi (soubor uspořádaných dat) a poskytuje řízený přístup k této databázi.

Hlavní funkcí všech databázových systémů je ukládání, vyvolávání a aktualizace dat.

Další vlastností DBMS je poskytování systémového katalogu (metadat, popisu struktury dat uložených v databázi) uživatelům i DBMS.

Významnou vlastností je také podpora transakcí. Transakcí se v DBMS myslí sled operací, které se provedou buď všechny, nebo se neprovede ani jedna z nich. V praxi to znamená, že pokud transakce během provádění jednotlivých kroků selže před jejím dokončením, je možné všechny částečně provedené změny vrátit zpět do původního stavu (může být proveden roll-back těchto operací). Služba, která umožňuje vrácení transakce do konzistentního stavu se nazývá služba zotavení.

Jedním z hlavních cílů DBMS je umožnit současný přístup ke sdíleným datům mnoha uživatelům (řízení souběžného přístupu) pro čtení i zápis, přičemž musí být ošetřeny kolize při zápisu do stejného záznamu nebo položky záznamu.

K ochraně databáze před neautorizovaným přístupem slouží autorizační služby, které umožňují přístup k jednotlivým částem databáze na základě definovaných přístupových práv pro uživatele a skupiny uživatelů.

Dalším prvkem ochrany databáze je služba integrity, která zajišťuje konzistenci dat. Může být například definováno na úrovni databáze, jaký typ dat a s jakými omezeními může být vkládán do určitého sloupce tabulky.

Každý DBMS by také měl podporovat datovou komunikaci pro přístup klienta k databázi pomocí terminálů nebo přes počítačovou síť pomocí komunikačních protokolů.

Standardní DBMS také poskytují službu podpory nezávislosti dat. Hlavní implementací této služby je podpora uživatelských pohledů na podmnožinu dat, která jsou pro daného uživatele relevantní.

Pro správce databáze slouží podpůrné programy (utility), které se využívají pro monitorování databázového systému, zadávání SQL příkazů, zálohování a obnovu databází a další činnosti.

Mezi výhody databázového přístupu patří např. udržování konzistence duplicitních dat, sdílení dat, zlepšení integrity dat možností definovat omezení, zlepšení výkonu díky nezávislosti dat, zabezpečení dat, souběžný přístup, vylepšené služby zálohování a zotavení, atd.

Nevýhodami databázových systémů mohou být jejich složitost (uživatelé musí porozumět funkcionalitě systému), náklady na pořízení, údržbu a školení, větší vliv selhání díky centralizaci zdrojů nebo nižší výkon díky obecnějšímu využití.⁴

⁴ Srov. CONOLLY, T., BEGG, C., a HOLOWCZAK, R., *Mistrovství - databáze - Profesionální průvodce tvorbou efektivních databází.*, s. 53-59.

3 Relační model

Relační systém řízení databáze (RDBMS) představuje 2. generaci DBMS a je v současné době nejpoužívanějším typem DBMS. Je založen na relačním modelu dat, který navrhl E. F. Codd v roce 1970.

Tento model s sebou přinesl vysoký stupeň nezávislosti dat - aplikační programy nejsou ovlivněny změnou fyzického uložení dat, změnou řazení záznamů, apod.

Na základě relačního modelu byl vyvinut strukturovaný dotazovací jazyk SQL (Structured Query Language), který je v současné době používán jako standardní jazyk relačních DBMS, a komerční relační systémy řízení databází jako IBM DB2, Oracle a Microsoft SQL Server. Kromě výše zmíněných serverových implementací relačního modelu existují také implementace založené na PC, jako např. Microsoft Office Access, Microsoft Visual FoxPro, Borland JDataStore, a další.

Relační model stejně jako jiné modely dat slouží ke srozumitelné reprezentaci dat, a popisuje:

- Jakým způsobem mají být data v databázi uložena (strukturální část)
- Jaké operace je možno s daty provádět (manipulační část)
- Pravidla pro zajištění integrity dat

3.1 Strukturální část relačního modelu

Strukturální část relačního modelu popisuje složky databáze, jejich vlastnosti a jakým způsobem spolu interagují. Mezi složky databáze patří:

- Relace – tabulka se sloupci a řádky
- Atribut – pojmenovaný sloupec relace
- Datová n-tice – záznam relace
- Doména – množina přípustných hodnot pro jeden nebo více atributů
- Relační databáze – množina normalizovaných tabulek

Relační tabulky mají následující vlastnosti:

- Tabulka má jedinečné jméno v rámci všech tabulek databáze
- Každý sloupec má jedinečné jméno v rámci tabulky
- Všechny hodnoty v jednom sloupci jsou ze stejné domény
- Pořadí sloupců ani řádků nemá význam
- Každý záznam je jedinečný – tato vlastnost je implementována pomocí primárního klíče tabulky

Jak bylo zmíněno výše, každý záznam v tabulce by měl být jedinečný. K zajištění tohoto předpokladu slouží primární klíč tabulky (seznam názvů sloupců, které jednoznačně identifikují každý záznam v tabulce). Kromě primárního klíče je definován také pojem cizí klíč, který slouží k propojení dceřiné tabulky s rodičovskou tabulkou (cizí klíč dceřiné tabulky odkazuje na primární klíč rodičovské tabulky).

3.2 Manipulační část relačního modelu

Manipulační část relačního modelu definuje typy operací nad daty. Mezi tyto operace patří zadávání, mazání a aktualizace dat, změna struktury databáze, atd. Provedení těchto operací je možné např. pomocí relačních jazyků SQL (Structured Query Language) a QBE (Query by Example).

3.3 Pravidla pro zajištění integrity dat

Integritní pravidla zajišťují přesnost dat. Mezi tato pravidla patří například doménová omezení, která určují množinu přípustných hodnot pro daný sloupec tabulky. Dalšími pravidly jsou entitní a referenční integrita. Entitní integrita zajišťuje, že sloupec primárního klíče nesmí obsahovat hodnoty NULL (prázdné hodnoty), jelikož by mohlo být porušeno pravidlo jednoznačnosti každého záznamu. Referenční integrita naproti tomu zajišťuje, že pokud existuje v tabulce cizí klíč, musí hodnota cizího klíče

odpovídat hodnotě některého záznamu v rodičovské tabulce, nebo musí mít cizí klíč prázdnou hodnotu.⁵

3.4 Relační databázové jazyky

Pro definici a manipulaci relačních databází slouží relační databázové jazyky, z nichž nejvýznamnějšími jsou jazyky SQL a QBE

3.4.1 Relační databázový jazyk SQL

Strukturovaný dotazovací jazyk SQL je v současné době nejrozšířenějším komerčním databázovým jazykem pro definici a manipulaci relačních databází. Umožňuje vytvářet struktury databází a tabulek, provádět údržbu dat včetně vkládání, modifikace a mazání dat z tabulek, a provádět jednoduché i složité dotazy.

Mezi hlavní vlastnosti a výhody SQL patří:

- Je možné se ho relativně snadno naučit
- Je neprocedurálním jazykem – zadávám, jaké informace požaduji, ne jak je získat
- Volný formát zadávání příkazů
- Struktura příkazů je založena na standardních anglických slovech
- Může být používán všemi druhy uživatelů od administrátorů databází přes aplikační programátory až po řídicí pracovníky a koncové uživatele
- Je standardizován a podporován všemi významnými dodavateli moderních databázových produktů

⁵ Srov. CONOLLY, T., BEGG, C., a HOLOWCZAK, R., *Mistrovství - databáze - Profesionální průvodce tvorbou efektivních databází.*, s. 62-74.

Standard SQL ISO má dvě hlavní součásti:

- Jazyk pro definici dat (DDL, Data Definition Language)
- Jazyk pro manipulaci s daty (DML, Data Manipulation Language)

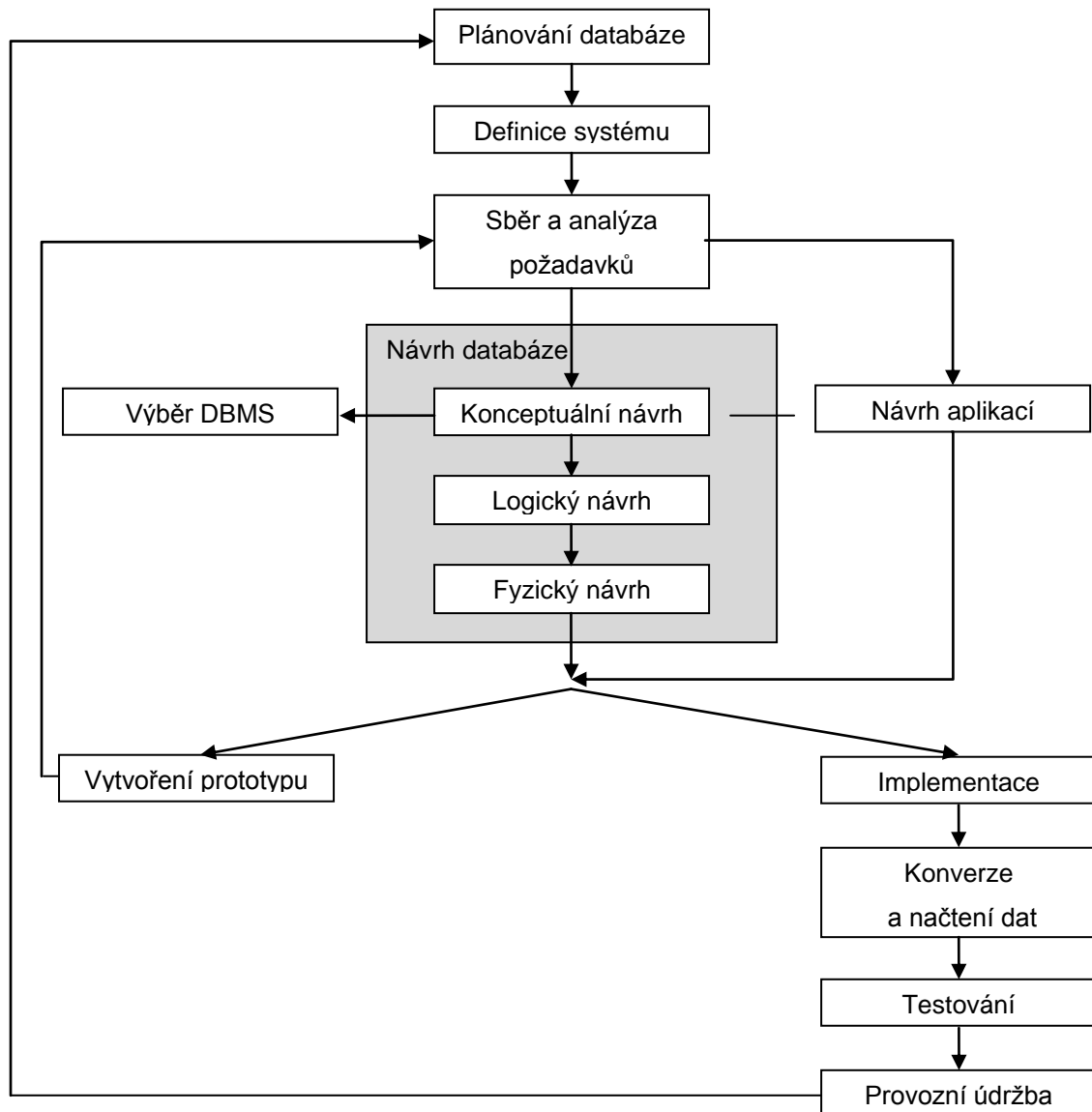
3.4.2 Relační databázový jazyk QBE

QBE (Query-By-Example) je grafickou nadstavbou jazyka SQL určenou pro netechnické uživatele a k zadávání jednodušších dotazů. K formulování dotazů na databázi se využívá grafické prostředí, ve kterém uživatel vybírá ze seznamu tabulky a pole, nad kterými chce provést dotaz a poté zadává, která pole mají být zobrazena ve výsledku a jaká kritéria mají být splněna. Po zadání vstupních dat a kritérií QBE vygeneruje SQL příkaz, který spustí a zobrazí výsledek.⁶

⁶ Srov. CONOLLY, T., BEGG, C., a HOLOWCZAK, R., *Mistrovství - databáze - Profesionální průvodce tvorbou efektivních databází.*, s. 75-106.

4 Životní cyklus vývoje databázového systému

Životní cyklus vývoje databázového systému se skládá z několika fází. Jednotlivé fáze mají mezi sebou vazby a na základě těchto vazeb mohou následovat po sobě nebo se některé mohou opakovat.



Obr. 1 – Životní cyklus plánování databázového systému ⁷

⁷ CONOLLY, T., BEGG, C., a HOLOWCZAK, R., *Mistrovství - databáze - Profesionální průvodce tvorbou efektivních databází.*, s. 110.

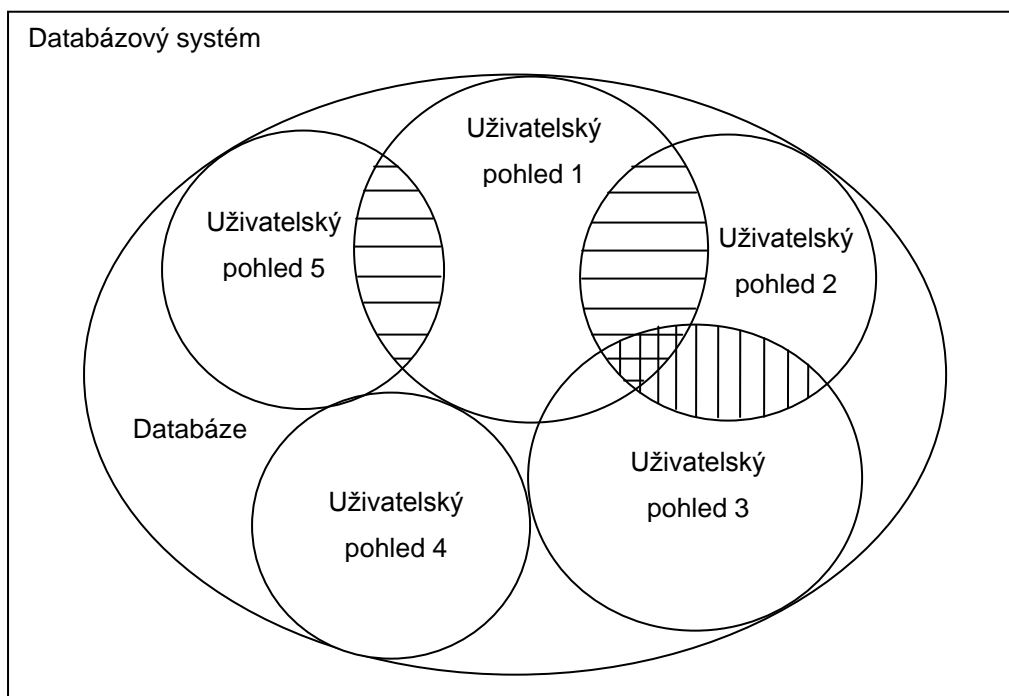
Plánování databáze

Počátečním bodem databázového projektu je stanovení poslání (hlavního cíle) databázového systému a dílčích cílů (úkolů), které by měla databáze řešit. Pokud bude databáze součástí rozsáhlejšího informačního systému, měl by být databázový projekt integrován do celkové strategie IS organizace.

Definice systému

Před návrhem databázového systému by měl být definován rozsah a hranice databázového systému včetně hlavních uživatelských pohledů.

Uživatelské pohledy definují požadavky na databázový systém ze strany jednotlivých pracovních pozic nebo oddělení a pomáhají zajistit, že žádný významný uživatel nebude opomenut při vývoji požadavků na databázový systém. Jednotliví uživatelé mohou mít vlastní pohledy, nebo se mohou překrývat s pohledy jiných uživatelů. Na obrázku 2 je znázorněn databázový systém s pěti uživatelskými pohledy. Uživatelský pohled 4 je samostatný, ostatní pohledy se v některých prvcích překrývají.



Obr. 2 – Uživatelské pohledy na databázový systém⁸

⁸ CONOLLY, T., BEGG, C., a HOLOWCZAK, R., *Mistrovství - databáze - Profesionální průvodce tvorbou efektivních databází.*, s. 112.

Sběr a analýza požadavků

Další fází je sběr a analýza požadavků na databázový systém. Významným zdrojem požadavků je sbírání a analýza informací o organizaci nebo části organizace, které bude databáze sloužit. Sbíráni informací se provádí také pro každý uživatelský pohled.

Po ukončení sběru a analýzy požadavků následuje návrh databáze, který se skládá ze třech fází – konceptuálního návrhu databáze, logického návrhu databáze a fyzického návrhu databáze.

Konceptuální návrh databáze

Konceptuální návrh databáze znamená vytvoření modelu dat používaných v organizaci bez jakýchkoli úvah o fyzické implementaci, identifikuje důležité entity a relace, které je třeba reprezentovat v databázi.

Logický návrh databáze

Logický návrh databáze obsahuje v případě relačního modelu návrh relačních tabulek a relací nezávisle na konkrétní DBMS.

Fyzický návrh databáze

Fyzický návrh databáze znamená implementaci logického návrhu na zvolený DBMS, jako např. Microsoft Access, Oracle, Microsoft SQL Server, atd.

Výběr DBMS

Pro výběr DBMS je vhodné zvolit dobu mezi logickým a fyzickým návrhem. Pokud máme dostatek informací na požadovaný DBMS, jako je požadovaný výkon, možnosti síťování, bezpečnostní funkce, možnosti integritních omezení, atd., je možné zvolit DBMS kdykoli před fází logického návrhu.

Návrh aplikací

Souběžně s návrhem DBMS se většinou navrhuje jedna nebo více aplikací, které poskytují aplikační logiku, přístup k databázi a uživatelské rozhraní. Mezi transakce, které aplikace provádí nad databází patří např. zobrazování údajů a aktualizace údajů.

Vytvoření prototypu databázového systému

V průběhu procesu návrhu databáze je vhodné hlavně u složitějších projektů vytvořit prototyp (fungující model) části databázového systému, který je možné použít pro ověření funkčnosti návrhu databáze, případně ke změně nebo doplnění požadovaných vlastností. Pokud je po otestování prototypu identifikována potřeba změnit nebo doplnit některé vlastnosti, je třeba návrh databáze aktualizovat.

Po ukončení návrhu databáze a úspěšném otestování prototypu (pokud byl použit) je možné databázi a databázové aplikace implementovat (fyzicky realizovat návrh databáze a návrh aplikací).

Implementace návrhu databáze a aplikací

Implementace návrhu databáze se provádí pomocí jazyka pro definici dat (DDL, Data Definition Language) nebo grafického uživatelského rozhraní (GUI, Graphical User Interface), přičemž GUI poskytuje stejné funkce jako DDL, ale skrývá syntaxi DDL příkazů před uživatelem. DDL příkazy slouží k vytvoření požadovaných databázových struktur a prázdných tabulek. V této fázi dochází k implementaci uživatelských pohledů.

Implementace návrhu aplikací se provádí většinou pomocí programovacích jazyků, jako např. Visual Basic, Delphi, Java, Cobol, Fortran, .NET a dalších. Tyto jazyky zabezpečují programovou logiku, interakci s uživateli pomocí obrazovek menu, vstupních formulářů a výstupních sestav a interakci s databází, typicky pomocí jazyka pro manipulaci s daty (DML, Data Manipulation Language). Součástí implementace návrhu aplikací je také zabezpečení a kontrola integrity aplikace.

Konverze a načtení dat

Pokud nahrazujeme starý databázový systém novým, je většinou nutné provést konverzi existujících dat. K tomuto účelu poskytují DBMS konverzní utility. Konverze musí být naplánována tak, aby došlo k hladkému přechodu z jednoho systému do druhého a nenarušila se provozuschopnost firmy.

Testování

Před uvedením systému do ostrého provozu se provádí testování, které se využívá k nalezení a odstranění chyb ve fungování databáze a databázových aplikací a ke zjištění, jestli byly splněny všechny požadavky na systém, které byly definovány v průběhu fáze sběru a analýzy požadavků. Testování by se měli účastnit uživatelé nového systému a měli by testovat reálná data nebo jejich část. Pokud databáze a databázové aplikace neodpovídají požadavkům, jsou navržena, schválena a provedena nápravná opatření a opětovně provedeny potřebné fáze životního cyklu vývoje databázového systému. Pokud požadavkům odpovídají, mohou databáze i databázové aplikace přejít do ostrého provozu.

Provozní údržba

V ostrém provozu je třeba pravidelně provádět provozní údržbu, která zahrnuje monitorování výkonu databázového systému a udržování a aktualizaci databázového systému.⁹

⁹ Srov. CONOLLY, T., BEGG, C., a HOLOWCZAK, R., *Mistrovství - databáze - Profesionální průvodce tvorbou efektivních databází.*, s. 107-121.

5 MS Access a MS SQL Server jako příklady implementace relačního modelu

Jak již bylo zmíněno výše, existuje mnoho implementací relačního modelu. V této kapitole budou porovnány výhody a nevýhody příkladů serverové implementace a implementace založené na PC, konkrétně produktů Microsoft SQL Server a Microsoft Office Access a možnosti spolupráce obou produktů.

5.1 MS Office Access

MS Office Access je databáze určená pro individuální pracovníky a malé týmy, kteří používají systém Microsoft Office pro řízení času, projektů, stanovování priorit, pro komunikaci a zpracování neustále se zvyšujícího množství informací. MS Office Access těmto pracovníkům umožňuje ve známém prostředí efektivně vytvářet aplikace bez hlubokých znalostí profesionálních vývojářů, přistupovat k heterogenním datům pro podporu rozhodování a sdílet informace s ostatními členy týmu.

5.1.1 Výhody a nevýhody MS Accessu

Mezi hlavní výhody MS Office Access patří:

- Jednoduchost použití
- Těsná integrace s ostatními aplikacemi MS Office a Windows SharePoint Services
- Rychlá tvorba tabulek, formulářů, dotazů, reportů a programové logiky
- Rychlá odezva na potřeby firmy, podpora rozhodování ad hoc
- Nízké náklady na pořízení, školení a používání
- Většina Access databází může být vytvořena bez součinnosti IT oddělení

Mezi nevýhody MS Office Access patří:

- Databáze vytvořená v MS Access je vhodná pro menší týmy (do 20 lidí), v opačném případě dochází k větší zátěži sítě a snížení výkonu databáze
- Pokud nebyla databáze vytvořena v součinnosti s IT oddělením, vyvstává riziko špatně navrženého datového schématu, a při odchodu tvůrce databáze riziko ztráty know-how o databázi
- Databázový stroj Microsoft Jet nepoužívá protokol transakcí, takže nelze zcela zabránit poškození databáze¹⁰

5.1.2 Topologie konfigurací MS Accessu

Standardní topologie konfigurace Microsoft Access pro jednoho uživatele se skládá z aplikace Microsoft Access, relačního databázového stroje Microsoft Jet a MDB databáze. Aplikace Microsoft Access zajišťuje uživatelské rozhraní a aplikační logiku, databázový stroj Microsoft Jet slouží jako rozhraní mezi aplikací Access a fyzickými daty, fyzická data jsou uložena v MDB databázi. Access i Jet běží na počítači uživatele a databáze je uložena na lokálním pevném disku v souboru typu .mdb.

Microsoft Jet je relační databázový stroj, který řídí databázové procesy pro Access. Jelikož Jet nemá uživatelské rozhraní, je pro jeho použití potřeba program jako např. Access. Jet umí spolupracovat také s jinými databázemi, jako Paradox nebo dBase, nebo může být zpřístupněn klientskými aplikacemi pomocí ODBC (Open Database Connectivity) driveru.

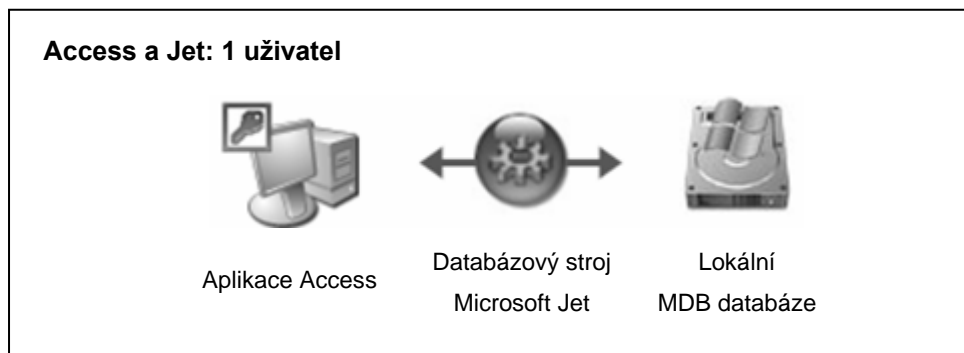
¹⁰ Srov. CHUNG, L. *Microsoft Access or Microsoft SQL Server: What's Right in Your Organization?*.

Access standardně ukládá data do souboru ve formátu databázového stroje Jet (do souboru typu .mdb). Do stejného souboru ukládá Jet také:

- Indexy
- Popis relací mezi tabulkami
- Validační pravidla
- Definice dotazů
- Bezpečnostní práva

Při použití Accessu ukládá Jet do .mdb souboru také následující databázové objekty:

- Formuláře
- Sestavy
- Makra
- Moduly¹¹



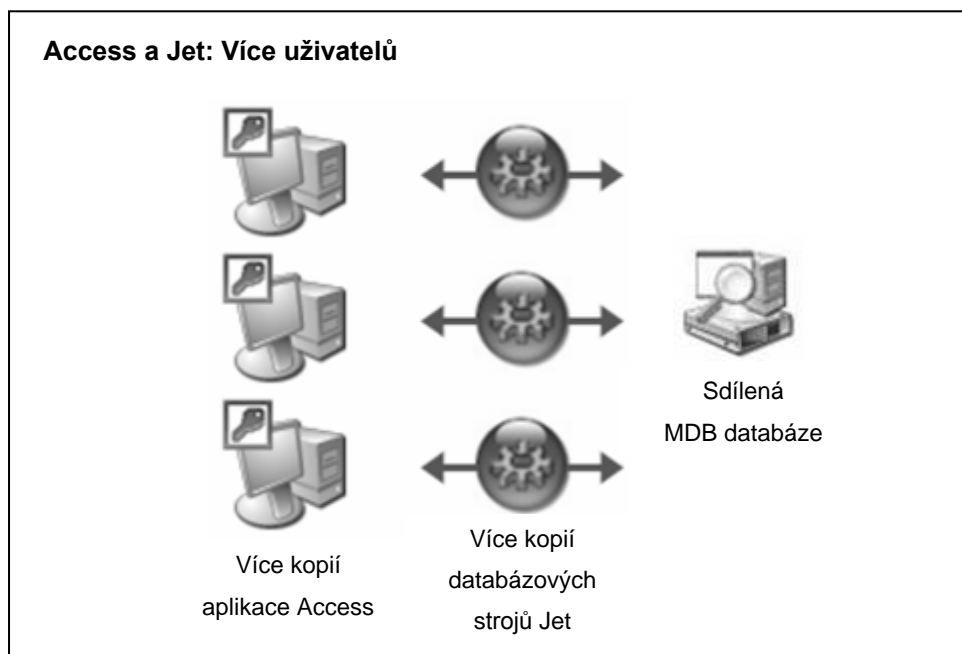
Obr. 3 – Topologie Access a Jet: 1 uživatel ¹²

¹¹ Srov. Microsoft Support, *Description of frequently used terms in Microsoft Access*.

¹² CHUNG, L, HAUGHT, D. *When to Migrate from Microsoft Access to Microsoft SQL Server*.

Access a databázový stroj Jet umožňují také víceuživatelský přístup. V tomto případě běží na počítačích jednotlivých uživatelů lokální kopie aplikace Access i databázového stroje Jet a data jsou uložena ve sdílené MDB databázi na sdíleném disku.

13



Obr. 4 – Topologie Access a Jet: Více uživatelů¹⁴

Kromě zmíněných dvou základních topologií podporuje MS Access také rozšířené možnosti připojení k jiným zdrojům dat, které budou popsány níže.

¹³ Srov. CHUNG, L, HAUGHT, D. *When to Migrate from Microsoft Access to Microsoft SQL Server*.

¹⁴ Tamtéž.

5.2 MS SQL Server

MS SQL Server je příklad pokročilé serverové implementace relačního modelu. Kromě základních funkcí RDBMS nabízí také řadu pokročilých funkcí jako např. fulltextové vyhledávání, English Query, analytické služby, replikace, služby transformace dat, podporu symetrického multiprocessingu (SMP), podporu clusterů, a další.¹⁵

5.2.1 Výhody a nevýhody MS SQL Serveru

Mezi hlavní výhody MS SQL Serveru patří:

- Výkon
- Rozšiřitelnost
- Bezpečnost
- Spolehlivost
- Klient/server databázový stroj

Mezi hlavní nevýhody MS SQL Serveru patří:

- Nutnost plánovat finanční a lidské zdroje s dostatečným časovým předstihem
- Delší doba realizace
- Vyšší náklady na hardwarovou a operační platformu, na školení, personál, údržbu

¹⁵ Srov. VIEIRA, R., *SQL Server 2000 - Programujeme profesionálně*, s. 5-7.

5.3 Srovnání vlastností MS Accessu a MS SQL Serveru

Následující tabulka zobrazuje srovnání vlastností MS Accessu a MS SQL Serveru, které by mohly být brány v úvahu při výběru DBMS jako součásti životního cyklu vývoje databázového systému.

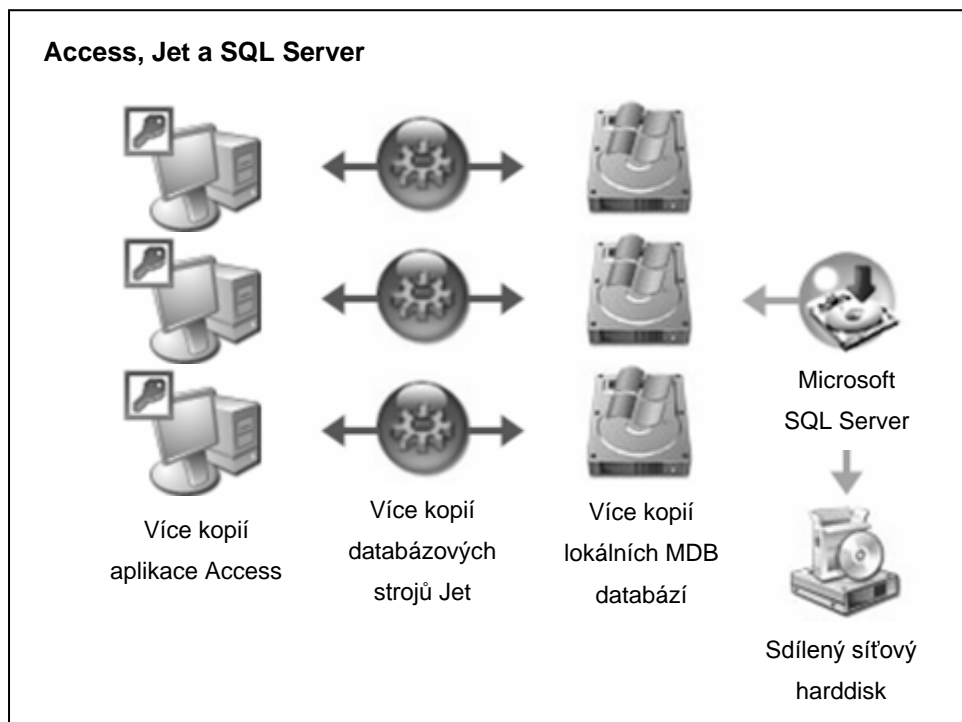
	Access	SQL Server
Popis	Vývojové databázové prostředí, které podporuje tabulky, dotazy, formuláře, sestavy a programovou logiku.	Rozšiřitelný, spolehlivý a více bezpečný databázový engine typu client/server.
Maximální velikost databáze	2 GB	1 TB nebo více
Maximální počet současně pracujících uživatelů	Až 20 současně editujících uživatelů, Až 100 uživatelů spouštějících sestavy.	Neomezen
Bezpečnost	Zabezpečení na úrovni souborového přístupu.	Bezpečnost na úrovni vyspělých podnikových informačních systémů
Výkon	Limitovaný na úrovni sdílení souboru.	Limitovaný pouze hardwarem a návrhem aplikace.
Spolehlivost	Adekvátní pro použití individuálními uživateli a malými týmy. Není implementována podpora zotavení po síťové chybě.	Vysoká spolehlivost. SQL Server je mission-critical databáze. Nástroje pro zálohování a administraci.

Tab. 2 – Srovnání vlastností MS Accessu a MS SQL Serveru ¹⁶

¹⁶ CHUNG, L, HAUGHT, D. *When to Migrate from Microsoft Access to Microsoft SQL Server.*

5.4 Topologie MS Access a MS SQL Server

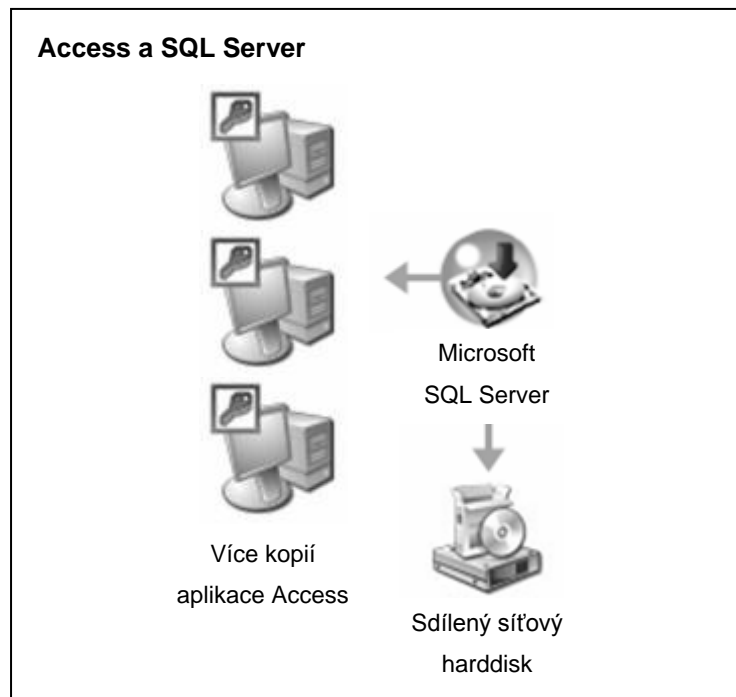
MS Access také umožňuje využívat jako úložiště dat MS SQL Server nebo jiné databáze. Jednou z možností je využití databázového stroje Jet ke spuštění dotazů, k ukládání definic objektů, ke správě dočasných tabulek a k držení bezpečnostních nastavení a vlastní data ukládat do databáze MS SQL Server, jak je znázorněno na obrázku 5.



Obr. 5 – Topologie Access, Jet a SQL Server ¹⁷

¹⁷ CHUNG, L, HAUGHT, D. *When to Migrate from Microsoft Access to Microsoft SQL Server.*

Od verze MS Access 2000 je také možné využívat Access datové projekty, které používají MS Access jako front-end (uživatelské rozhraní a aplikační logika) a MS SQL Server jako back-end (datové struktury a data). MS Access v tomto případě přistupuje k datům uloženým v MS SQL Server databázi přímo bez použití databázového stroje Jet. Tento způsob přístupu je zobrazen na obrázku 6.



Obr. 6 – Topologie Access a SQL Server¹⁸

¹⁸ CHUNG, L, HAUGHT, D. *When to Migrate from Microsoft Access to Microsoft SQL Server.*

6 Představení společnosti EXL

EXL je americká společnost, která byla založena v roce 1999, a je vedoucím světovým poskytovatelem transformačních a outsourcingových služeb v oblastech pojišťovnictví, bankovníctví, finančních služeb, užití, přepravy a logistiky.

Centrála firmy se nachází v New Yorku, kde je také obchodní kancelář. Další obchodní kanceláře má v New Jersey a Londýně, dále pak disponuje 12-ti delivery centry v Indii, Filipínách, a po jednom centru v České republice a Rumunsku. Celá firma má po celém světě více než 10 700 zaměstnanců.

Česká pobočka firmy EXL v Olomouci vznikla v roce 2009 odkupem části firmy Schneider National, která se zabývá přepravou, a která chtěla vyčlenit část svých činností mimo firmu a věnovat se své hlavní podnikatelské činnosti. Firma EXL tak převzala všech 200 zaměstnanců a všechny činnosti, které tito zaměstnanci prováděli a poskytuje všechny tyto činnosti firmě Schneider National formou outsourcingu.

7 Informační systém pobočky ExlService Czech Republic

Informační systém pobočky ExlService Czech Republic pokrývá 3 základní potřeby:

- Provádění požadovaných činností pro zákazníka
- Komunikace a výměna informací s ostatními částmi firmy EXL
- Zabezpečení podpůrných činností pobočky

Celá část informačního systému, která pokrývá provádění činností pro zákazníka včetně osobních počítačů, serverů a aplikací byla migrována k zákazníkovi. Zaměstnanci EXLu přistupují k těmto systémům pomocí vzdáleného přístupu přes webové rozhraní ze svých osobních počítačů. Tato část informačního systému je podporována převážně ze strany zákazníka.

Druhá část informačního systému pokrývající komunikaci a výměnu informací s ostatními částmi firmy EXL byla implementována na základě zvyklostí a politik firmy EXL a je podporována korporátním IT oddělením a IT oddělením pobočky. Pokrývá oblasti typu ukládání dat, tisku, telefonní a datové komunikace, atd.

Třetí část informačního systému slouží podpůrným oddělením pobočky, kam patří především finanční, IT, facility a personální oddělení. Tato část informačního systému je podporovaná převážně zaměstnanci IT oddělení pobočky. Mezi hlavní aplikace patří především účetní systém, systémy pro evidenci a inventarizaci majetku, systémy pro komunikaci s dodavateli služeb, apod.

Jedním z požadavků, který vyvstal při převzetí pobočky Schneider National, byla potřeba vytvoření nástroje pro evidenci a inventarizaci veškerého hardwaru pobočky, který by měl být vytvořen, používán a aktualizován IT oddělením v Olomouci, a jehož výstupy by měly sloužit také pro potřeby finančního oddělení. Tento úkol jsem převzal a popis jeho řešení je součástí následujících kapitol.

8 Vytvoření databáze hardwaru

Jelikož nástroj, který by měl být vytvořen, by měl sloužit k ukládání, aktualizaci a získávání organizovaných dat, rozhodl jsem se tuto problematiku řešit pomocí databázového systému, konkrétně pomocí relační databáze. Vzhledem k tomu, že firma má zakoupenou licenci pro používání databázových systémů Microsoft Office Access a MS SQL Server, které pokrývají serverovou implementaci relačního modelu (MS SQL Server) i implementaci založenou na PC (MS Access), zaměřil jsem svoji pozornost při výběru technologie a konfigurace databázového systému na tyto dva produkty. Při návrhu, implementaci i uvedení databáze do provozu jsem se držel metodologie životního cyklu vývoje databázového systému uvedené v kapitole 4.

8.1 Plánování databáze

Hlavní náplní plánování databáze je definice poslání databázového projektu a určení dílčích cílů nebo úkolů, které musí databáze plnit.

8.1.1 Poslání databáze

Databáze hardwaru bude spravovat data o veškerém používaném hardwaru na pobočce firmy ExlService v Olomouci a umožňovat snadné a rychlé získávání informací pro potřeby evidence, inventarizace a sledování záruční doby hardwaru.

8.1.2 Dílčí cíle

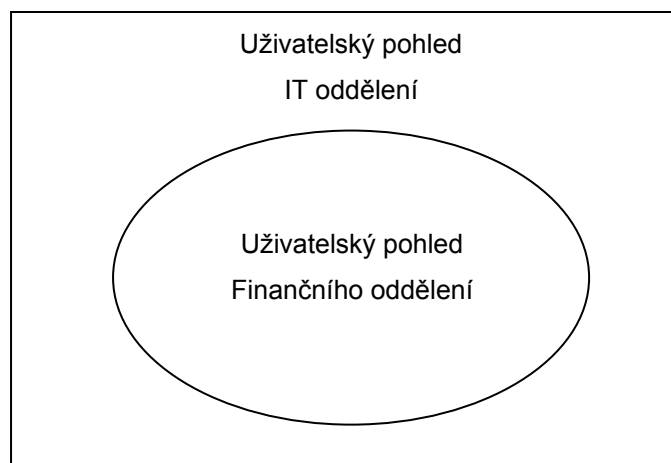
Dílčí cíle jsem zjistil na základě rozhovorů (formou otázek a odpovědí) se zaměstnanci, kteří budou systém využívat.

Na základě těchto rozhovorů vplynuly následující dílčí cíle databázového projektu:

- Udržovat (zadávat, aktualizovat a mazat) data o veškerém hardwaru
- Udržovat data o zaměstnancích, kteří hardware používají
- Udržovat data o odděleních, ve kterých zaměstnanci pracují
- Udržovat data o kancelářích a pracovních místech, kde se hardware nachází
- Udržovat data o výrobcích hardwaru
- Vyhledávat údaje o veškerém hardwaru
- Vyhledávat údaje o zaměstnancích
- Sledovat umístění hardwaru
- Sledovat, kdo hardware používá
- Sledovat záruční dobu jednotlivých zařízení
- Vytvářet sestavy o umístění hardwaru pro provádění inventur
- Vytvářet sestavy o využití hardwaru
- Vytvářet sestavy o záručních dobách hardwaru

8.2 Definice systému

Pro definici systému jsem použil hlavní uživatelské pohledy provozních aplikačních oblastí, konkrétně oddělení IT a finančního oddělení. Tyto pohledy jsem definoval na základě předchozích rozhovorů s pracovníky daných oddělení a na základě vyhodnocení získaných dat.



Obr. 7 – Uživatelské pohledy IT a finančního oddělení

Uživatelský pohled finančního oddělení je podmnožinou uživatelského pohledu IT oddělení.

8.3 Sběr a analýza požadavků

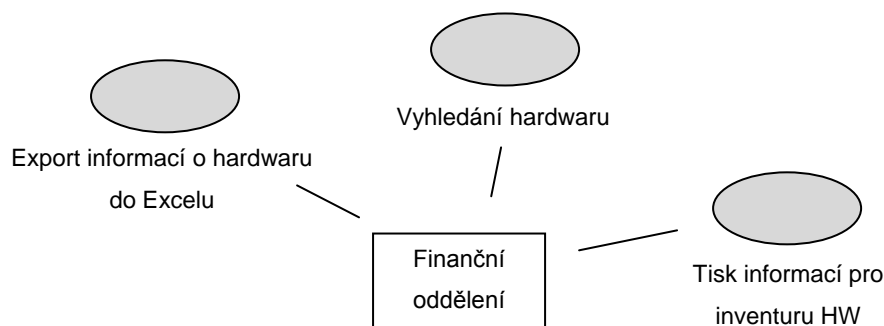
Během fáze sběru a analýzy požadavků jsem získával podrobnější údaje o uživatelských pohledech identifikovaných v předcházející fázi. Pomocí těchto údajů jsem specifikoval uživatelské požadavky, které popisují data, která se budou uchovávat v databázi, a způsob, jakým se budou používat.

Pohled oddělení IT vyžaduje přístup ke všem datům, pohled finančního oddělení pouze k některým z nich.

	Zaměstnanci	Oddělení	Typ HW	Výrobce	Model	Stav HW	Záruka	Umístění HW
IT oddělení	X	X	X	X	X	X	X	X
Fin. oddělení			X	X	X	X	X	X

Tab. 3 – Data využívaná různými pohledy

V následující části je zobrazena funkčnost databázového systému z hlediska obou uživatelských pohledů. Na obrázku 8 je diagram případů užití pro uživatelský pohled finančního oddělení, v tabulce 4 jsou případy užití popsány.



Obr. 8 – Diagram případů užití pro uživatelský pohled finančního oddělení

Užití	Popis
Přidání nového výrobce	Pracovníci IT oddělení mohou vytvářet záznamy o výrobcích hardwaru. Každý výrobce je jednoznačně identifikován číslem výrobce.
Aktualizace výrobce	Pro aktualizaci záznamů o výrobcích.
Přidání nového modelu HW	Pro vytvoření záznamu o novém modelu hardwaru. Každý model je jednoznačně identifikován číslem modelu HW.
Aktualizace modelu HW	Pro aktualizaci záznamů o modelech HW.
Přidání nového stavu HW	Pro vytvoření záznamu o novém stavu hardwaru. Každý stav HW je jednoznačně identifikován číslem stavu HW.
Aktualizace stavu HW	Pro aktualizaci záznamů o stavech HW.
Přidání nového typu HW	Pro vytvoření záznamu o novém typu hardwaru. Každý typ HW je jednoznačně identifikován číslem typu HW.
Aktualizace typu HW	Pro aktualizaci záznamů o typech HW.
Přidání nového oddělení	Pro vytvoření záznamu o novém oddělení. Každé oddělení je jednoznačně identifikováno číslem oddělení.
Aktualizace oddělení	Pro aktualizaci záznamů o odděleních.
Přidání nového zaměstnance	Pro vytvoření záznamu o novém zaměstnanci. Každý zaměstnanec je jednoznačně identifikován číslem zaměstnance.
Aktualizace zaměstnance	Pro aktualizaci záznamů o zaměstnancích.
Přidání nového HW	Pro vytvoření záznamu o novém HW zařízení. Každé zařízení je jednoznačně identifikováno číslem HW.
Aktualizace HW	Pro aktualizaci záznamů o hardwaru.
Přiřazení zaměstnance k oddělení	Když společnost přijme nového zaměstnance, je přiřazen do určitého oddělení.
Přiřazení hardwaru k oddělení	Když společnost koupí nové HW zařízení, přiřadí jej do oddělení, které ho používá.
Přiřazení hardwaru k výrobcí	Když společnost koupí nové HW zařízení, přiřadí jej k výrobcí, který ho vyrobil.
Vyhledání HW	Pracovníci IT oddělení si mohou prohlížet, třídít, filtrovat a vyhledávat informace o hardwaru.
Export informací o HW do Excelu	Informace o umístění, stavu a záruční doby si mohou exportovat do Excelu a poté s nimi dále pracovat (vyhledávat, třídít, zobrazovat podmnožiny souboru, atd.).
Tisk informací pro inventuru HW	Pro potřeby inventur HW. Inventury budou prováděny pracovníky finančního i IT oddělení.

Tab. 5 – Popis případů užití pro uživatelský pohled IT oddělení

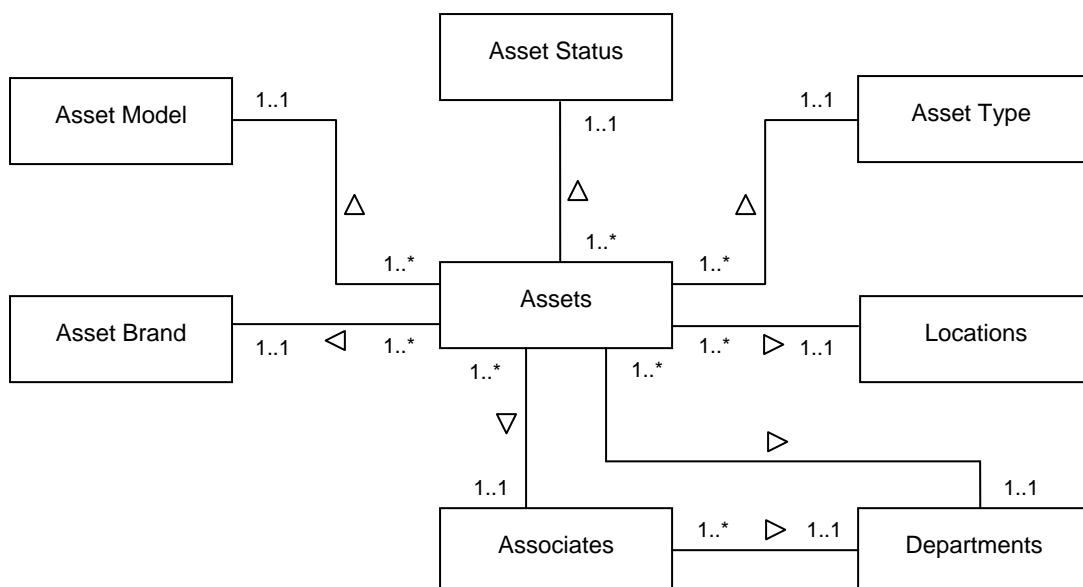
Během této fáze jsem také zjistil požadavky, které se týkají databázového systému obecně. Sem patří např. požadavky na počáteční velikost databáze, předpokládaná velikost v průběhu několika let, požadavky na síťování, sdílený přístup, zabezpečení, zálohování, uživatelské rozhraní, atd.

Zjištěné informace jsou následující:

- V počáteční fázi bude v databázi uloženo cca 1000 záznamů s informacemi o hardwaru, 200 záznamů s informacemi o zaměstnancích, 30 záznamů o odděleních a 15 záznamů o výrobcích hardwaru.
- Během 5-ti let se očekává zvýšení množství zaměstnanců, oddělení, hardwaru i výrobců v databázi na dvojnásobek.
- Přístup k databázi bude z lokální počítačové sítě, s aplikací může pracovat více lidí najednou (v jednu chvíli cca 3 pracovníci).
- Přístup do databáze pro zápis budou mít pouze pracovníci lokálního IT oddělení. Pracovníci finančního oddělení si budou pouze prohlížet data o hardwaru, spouštět exporty do Excelu a tisknout sestavy.
- Každý uživatel se musí před použitím aplikace přihlásit pod svým účtem a heslem. V aplikaci uvidí pouze informace, ke kterým má přístup.
- Databáze se musí zálohovat každý den po pracovní době.
- Z důvodu rychlosti je preferován jako uživatelské rozhraní tlustý klient spouštěný z lokálního počítače nebo ze sítě.

8.4 Konceptuální návrh databáze

Na základě dat, se kterými bude IT a finanční oddělení pracovat jsem vytvořil konceptuální návrh databáze, který identifikuje důležité entity a relace bez úvah o logické nebo fyzické implementaci. K vytvoření konceptuálního návrhu jsem použil entitně-relační modelování.



Obr. 10 – Entitně-relační diagram s entitami, relacemi a multiplicitou

8.5 Logický návrh databáze

V logickém návrhu databáze jsem převedl entitně-relační model do množiny relačních tabulek a doplnil informace o jednotlivých atributech (sloupcích tabulek) a primárních a cizích klíčích.

Assets (ID, Asset Tag, Serial No, Brand ID, Model ID, Status ID, Type ID, Location ID, Department ID, Associate ID, Our Ref No, Date of Tax, Date of Delivery, Guarantee End Day) Primární klíč ID Cizí klíč Brand ID odkazuje na Asset Brand (ID) Cizí klíč Model ID odkazuje na Asset Model (ID) Cizí klíč Status ID odkazuje na Asset Status (ID) Cizí klíč Type ID odkazuje na Asset Type (ID) Cizí klíč Location ID odkazuje na Locations (ID) Cizí klíč Department ID odkazuje na Departments (ID) Cizí klíč Associate ID odkazuje na Associates (ID)	Asset Brand (ID, Asset Brand) Primární klíč ID
	Asset Model (ID, Asset Model) Primární klíč ID
	Asset Status (ID, Asset Status) Primární klíč ID
	Asset Type (ID, Asset Type) Primární klíč ID
	Associates (ID, Personal No, User ID, Last Name, First Name, Department ID) Primární klíč ID Cizí klíč Department ID odkazuje na Departments (ID)
	Departments (ID, Department Name, Department Address, Department Phone) Primární klíč ID

Locations (ID, Location) Primární klíč ID	Departments (ID, Department) Primární klíč ID
---	---

Tab. 6 – Logický návrh relačních tabulek včetně primárních a cizích klíčů

8.6 Výběr DBMS a aplikace pro vytvoření uživ. rozhraní a aplikační logiky

Před fyzickým návrhem databáze je třeba znát databázové a aplikační prostředí, ve kterých bude databáze a aplikace vytvořena. Při výběru DBMS jsem bral v úvahu požadavky na systém, které jsem zjistil během fáze sběru a analýzy požadavků, a ty jsem srovnával s možnostmi, které jsou nabízeny databázovými systémy MS Access a MS SQL Server.

Z tohoto srovnání vyplynulo, že MS Access je dostatečným nástrojem pro vytvoření požadovaného systému, naproti tomu možnosti MS SQL Serveru požadavky na systém mnohonásobně převyšují. Rozhodl jsem se tedy využít pro vytvoření databáze hardwaru relační databázový systém MS Access, konkrétně verzi Microsoft Office Access 2007 a formát databáze MDB (formát kompatibilní s Access 2002 - 2003), který umožňuje zabezpečení na úrovni uživatele.¹⁹

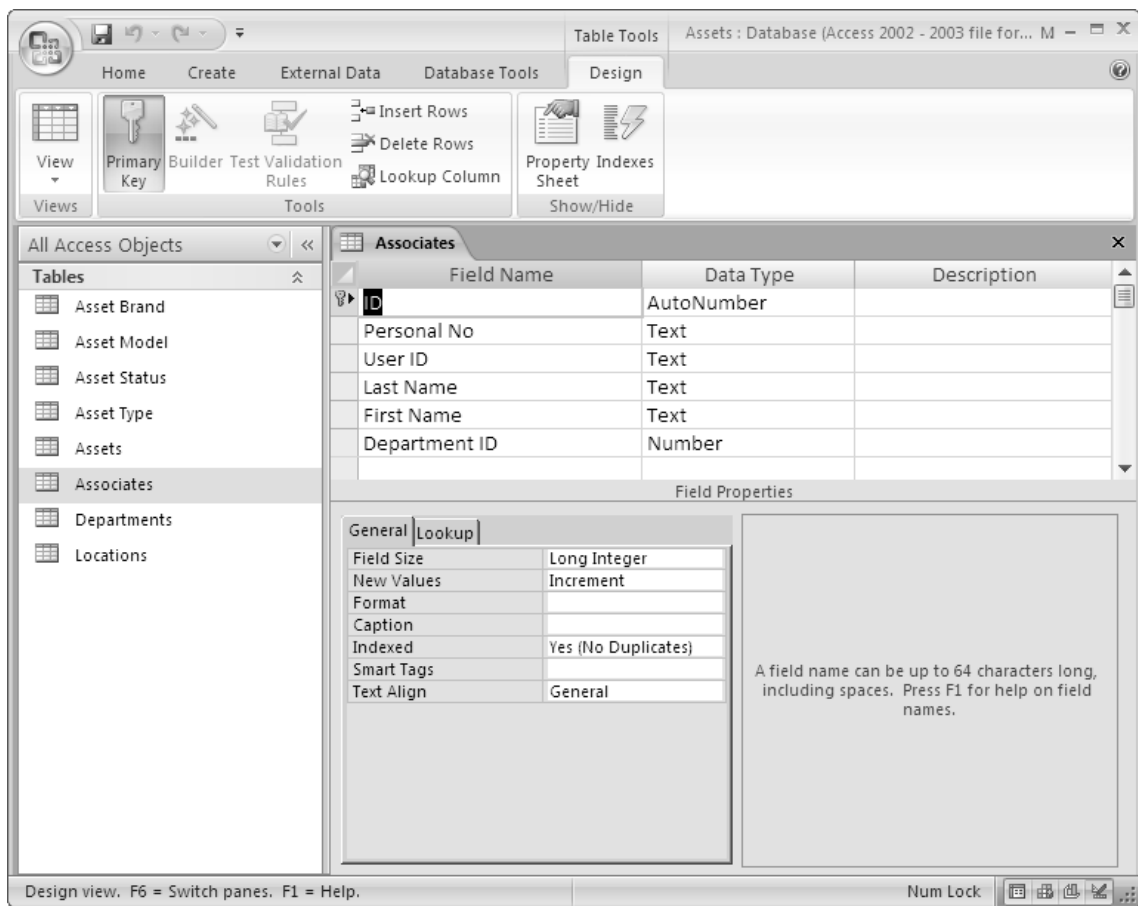
Jelikož MS Access v sobě také obsahuje nástroje pro vytváření uživatelského rozhraní a aplikační logiky, nebylo pro tyto činnosti potřeba vybírat další programové nástroje, jak by tomu bylo v případě použití DBMS MS SQL Server.

Jako topologii konfigurace MS Access jsem zvolil variantu „Access a Jet: Více uživatelů“, která nabízí použití tlustého klienta a přístup uživatelů ke sdílené databázi pomocí databázového stroje Microsoft Jet.

¹⁹ Srov. Microsoft. *Nastavení nebo změna uživatelské úrovně zabezpečení aplikace Access 2003 v aplikaci Access 2007.*

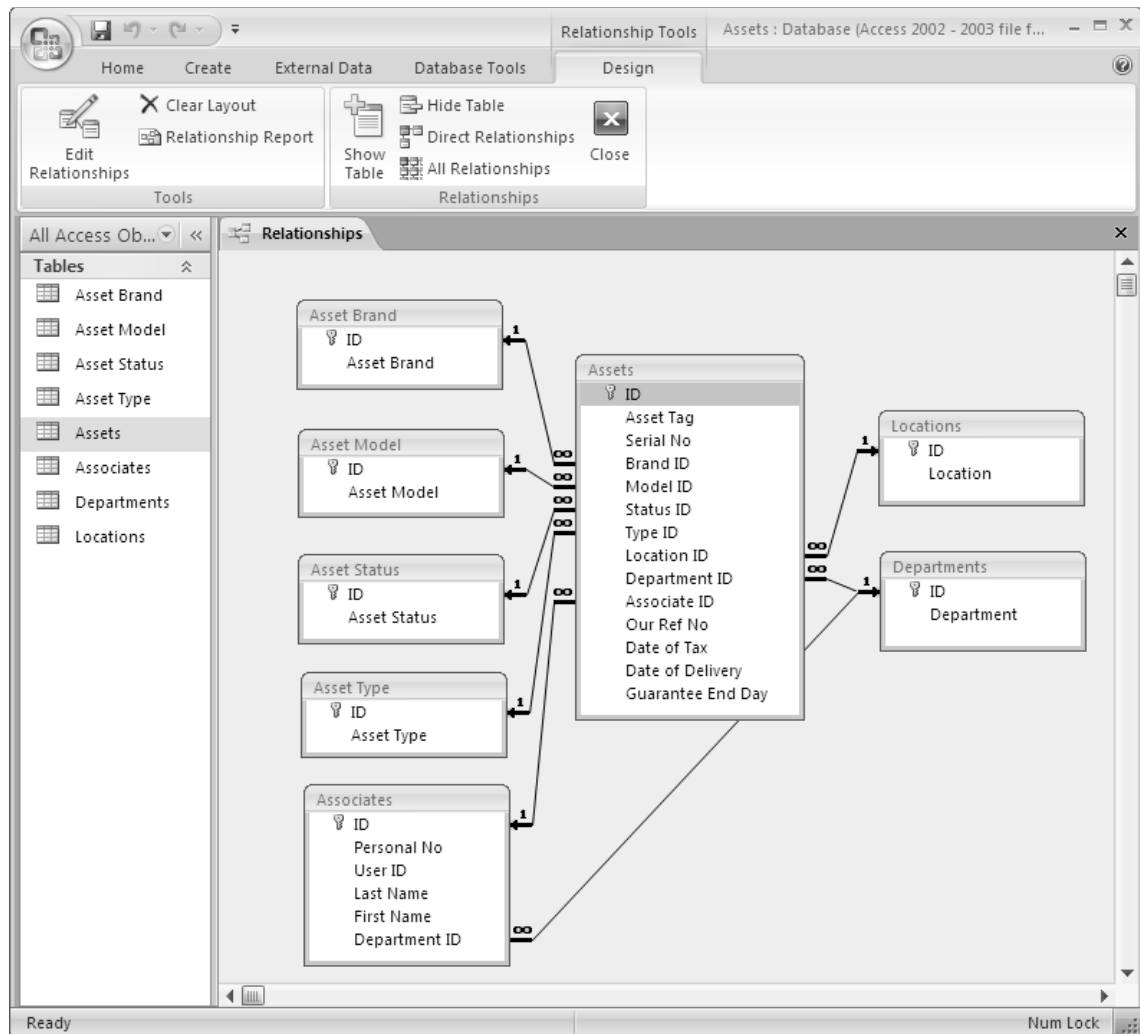
8.7 Fyzický návrh databáze a jeho implementace

Po výběru DBMS jsem vytvořil fyzický návrh databáze, ve kterém jsem převedl logický popis relačních tabulek do prostředí MS Access 2007. Během tohoto procesu jsem vytvořil tabulky, definoval sloupce a nastavil primární klíče. U každého sloupce jsem zadal jeho typ, délku, případná omezení, implicitní hodnotu, může-li obsahovat hodnoty null, a pokud byl sloupec odvozený, pak jakým způsobem má být vypočítán.



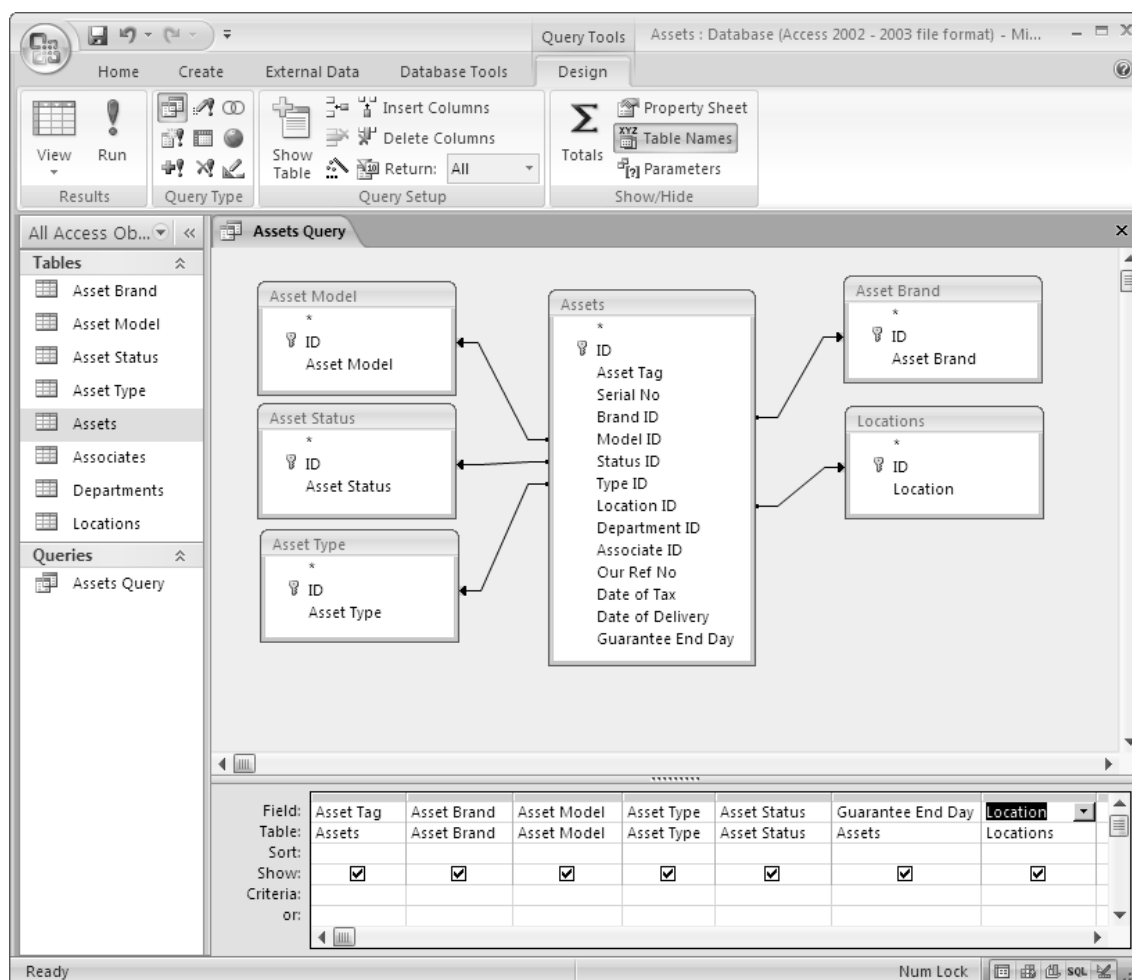
Obr. 11 – Vytváření tabulky Associates v návrhovém zobrazení

Po vytvoření všech tabulek jsem nastavil požadované relace. Pro tuto činnost slouží v Accessu okno Relace a vlastní relace se nastavují přetažením atributu primárního klíče rodičovské tabulky na cizí klíč dceřiné tabulky viz obr. 12.



Obr. 12 – Vytváření relací mezi tabulkami v okně Relace

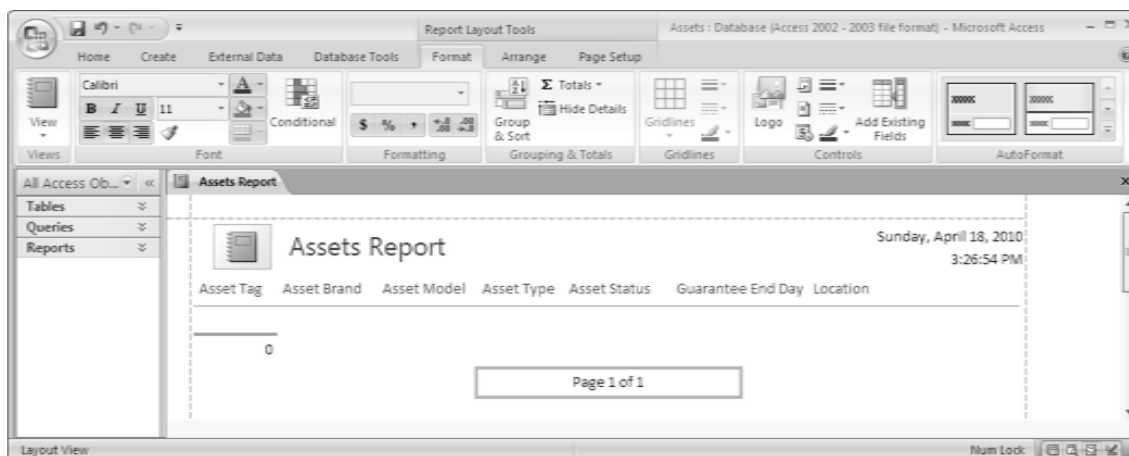
Poté jsem vytvořil uživatelské pohledy. V MS Access k tomuto účelu slouží QBE (Query by Example) viz obr. 13.²⁰



Obr. 13 – Uživatelský pohled vytvořený pomocí QBE

V další fázi jsem vytvořil sestavy pro prohlížení a tisk požadovaných informací a exporty do formátu tabulkového procesoru MS Excel pro další zpracování dat uživateli. Na obrázku 14 je příklad návrhového zobrazení při vytváření reportu pro finanční oddělení.

²⁰ Srov. KRUCZEK, Aleš. *Microsoft Office Access 2007 - Podrobná uživatelská příručka.*



Obr. 14 – Tisková sestava v návrhovém zobrazení

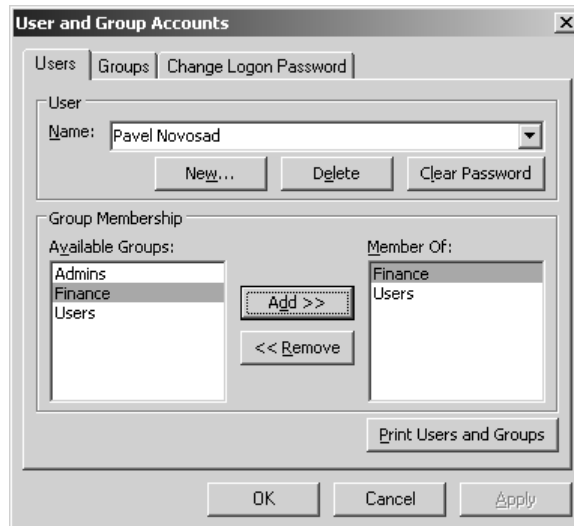
Během fyzického návrhu databáze jsem se také zabýval otázkou, kam umístit databázový soubor MDB, a jakým způsobem tento soubor zabezpečit proti neoprávněnému přístupu a proti jeho poškození nebo smazání.

Vzhledem k tomu, že k databázovému souboru MDB bude přistupovat více uživatelů, navrhl jsem jeho umístění do sdíleného adresáře na souborovém serveru, který bude nastaven tak, aby k němu měli přístup pouze uživatelé z finančního a IT oddělení. Ostatní uživatelé do něj mít přístup nebudou.

Databáze bude pravidelně každý den zálohována spolu s ostatními daty na souborovém serveru, a to v nočních hodinách po půlnoci. Tím bude zabezpečena obnova databáze v případě poškození nebo smazání. Pokud k této situaci dojde, bude možné se vrátit ke stavu databáze z předcházejícího dne.

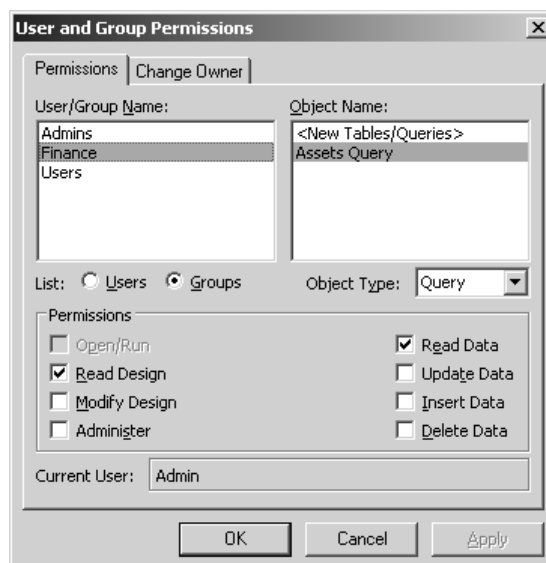
Další otázkou, kterou bylo třeba zodpovědět, byl způsob zabezpečení přístupu uživatelů k jednotlivým objektům databáze. Tento úkol jsem se rozhodl řešit pomocí bezpečnostní funkce MS Access „Zabezpečení na úrovni uživatelů“, která umožňuje vytvářet v rámci databáze uživatele a skupiny uživatelů, přiřazovat uživatele do skupin, a přidělovat uživatelům a skupinám přístupová práva k jednotlivým objektům databáze. Jelikož tato funkčnost je dostupná pouze v Access databázi typu MDB, nemohl jsem použít při tvorbě databáze nejnovější formát ACCDB, ale zmíněný formát MDB, který je v MS Access 2007 také podporován.

Pomocí této funkce jsem zabezpečil, aby měli zaměstnanci finančního oddělení práva pouze pro takové činnosti, které byly definovány během fáze sběru a analýzy požadavků, a aby zaměstnanci IT oddělení měli přístup neomezený. Obrázek 15 zobrazuje členství uživatele ve skupinách.



Obr. 15 – Definování uživatelských a skupinových účtů

Obrázek 16 ukazuje přístupová práva skupiny Finance k objektu Assets Query. Skupina Finance má k tomuto objektu pouze právo ke čtení.



Obr. 16 – Definování uživatelských práv

8.8 Testování databáze

Po fyzickém návrhu databáze a jeho implementaci v prostředí MS Access 2007 následovala fáze testování. Během tohoto kroku jednotliví uživatelé plnili databázi podmnožinou reálných dat a testovali funkčnost systému. Nalezené chyby a neshody s požadavky na systém, které byly nadefinovány během fáze sběru a analýzy požadavků, jsem zapracoval do nové verze databázového systému. Po vytvoření každé nové verze proběhlo testování vybraných částí systému znovu, až do vyřešení všech nedostatků.

Součástí testování bylo také otestování zálohování a obnovy dat na souborovém serveru, přístup uživatelů ke sdílenému adresáři a k MDB souboru, a přístupová práva uživatelů k jednotlivým objektům databáze.

8.9 Přejít do ostrého provozu

Po úspěšném otestování databázového systému následovalo odstranění zkušebních dat, naplnění databáze ostrými daty a rutinní používání systému.

8.10 Provozní údržba

V průběhu používání databázového systému je třeba tento systém pravidelně monitorovat a udržovat. Jako hlavní úkoly provozní údržby jsem definoval:

- Pravidelně sledovat velikosti MDB souboru a provádět údržbu databáze, aby byla zajištěna integrita dat
- Pravidelně zálohovat databázový soubor MDB a kontrolovat, je-li možné jej ze zálohy obnovit
- Aktualizovat přístupová práva ke sdílenému adresáři a k souboru MDB na souborovém serveru při přijetí nového nebo odchodu stávajícího zaměstnance do/z finančního a IT oddělení

- Aktualizovat skupiny Finance a IT definované v databázovém systému při přijetí nového nebo odchodu stávajícího zaměstnance do/z finančního oddělení a IT oddělení
- Monitorovat výkon systému

9 Omezení a rizika zvoleného řešení a doporučení pro další rozvoj

Jak bylo zmíněno v kapitole 5, MS Access má jistá omezení. Jedním z nich je, že je určen pro používání jednotlivci nebo menšími týmy. Pokud databázi začne používat větší množství uživatelů, hrozí riziko snížení jejího výkonu a vysokého zatížení sítě. Toto riziko lze eliminovat nepoužitím databázového stroje Microsoft Jet a MDB databáze, ale využitím podporované topologie MS Access a SQL Server viz obr. 6, s. 28. Na tuto topologii je možné také migrovat z topologií, které využívají databázový stroj Jet rozdělením aplikace na front-end a back-end.

Dalším omezením MS Accessu je, že v nové verzi databáze ACCDB není implementována podpora zabezpečení přístupu k objektům databáze na úrovni uživatelů, jak bylo implementováno ve verzi MDB. Vývojář Access aplikace se tak musí rozhodnout, jestli bude používat starší verzi databáze MDB a přijde o nové možnosti, které nabízí ACCDB, nebo bude používat ACCDB a přijde o možnost zabezpečení na úrovni uživatelů. Pokud by vývojář chtěl využít možnosti ACCDB a zároveň zabezpečení přístupu k objektům na úrovni uživatelů, musel by opět použít topologii MS Access a MS SQL Server nebo MS Access a jiný podporovaný serverový RDBMS.

Mezi další omezení patří menší spolehlivost systému a malá nebo žádná podpora vlastností, které nabízí pokročilé serverové databázové systémy, jako je podpora fulltextového vyhledávání, replikací, transformačních služeb, symetrického multiprocessingu, clusteringu apod. Je však třeba si uvědomit, že podpora těchto vlastností je u těchto systémů vykoupena několikanásobně vyšší cenou, složitostí vývoje, implementace, provozování a údržby a měly by být používány v prostředích pro která jsou dané systémy určeny.

Dle mého názoru topologie „Access a Jet: Více uživatelů“, kterou jsem pro požadovaný systém zvolil, splňuje všechny požadavky na systém kladené. V případě, že vyvstanou požadavky, které převýší jeho možnosti, bude možné je uspokojit změnou této topologie.

Závěr

Předmětem mé bakalářské práce bylo získání teoretických poznatků z oblasti databázových technologií a jejich aplikace při vývoji databáze hardwaru ve společnosti ExlService Czech Republic, s.r.o.

V teoretické části popisuji historii ukládání a organizování dat, systémy pro řízení databáze se zaměřením na relační databáze a metodologii vývoje databázového systému.

V praktické části jsem využil teoretické poznatky pro návrh a implementaci databáze hardwaru pomocí relačního databázového systému Microsoft Office Access 2007.

Hlavním cílem bakalářské práce bylo zjistit, je-li k vytvoření databáze hardwaru možno využít nástroje systému MS Office, konkrétně relačního databázového systému Microsoft Office Access 2007.

Při stanovování vhodnosti systému MS Access jsem porovnával požadavky společnosti na vytvářenou aplikaci s možnostmi MS Accessu. V této fázi jsem zjistil, že možnosti systému MS Access splňují veškeré požadavky na vytvoření požadované databáze. Dalšími kroky bylo vytvoření databáze a zhodnocení jejího praktického používání. I při této fázi se potvrdilo, že MS Access byl dobrou volbou pro vývoj této aplikace. Mezi hlavní silné stránky MS Accessu patřila rychlost a jednoduchost vývoje aplikace, intuitivní ovládání a možnost vytvořit kromě samotné databáze i uživatelské rozhraní a aplikační logiku.

Pokud bych měl zhodnotit MS Access obecně, doporučil bych jej k vývoji jednodušších databází, které budou používány v malých týmech, a u kterých nejsou kladeny vysoké nároky na výkon, rozšiřitelnost, bezpečnost a spolehlivost. Pro tyto aplikace jsou vhodnější serverové implementace RDBMS, jako např. Microsoft SQL Server, IBM DB2 nebo Oracle.

Anotace

Příjmení a jméno autora:	Nesvadba Petr
Instituce:	Moravská vysoká škola Olomouc
Název práce v českém jazyce:	Využití nástrojů MS Office k vytvoření databáze hardwaru ve firmě ExlService Czech Republic, s.r.o.
Název práce v anglickém jazyce:	Utilizing MS Office Tools for Creating a Hardware Database in the Company ExlService Czech Republic, Ltd.
Vedoucí práce:	Mgr. Květoslav Bártek, Ph.D.
Počet stran:	52
Rok obhajoby:	2010
Klíčová slova v českém jazyce:	DBMS, RDBMS, vývoj databázového systému, databáze, tabulka, relace
Klíčová slova v anglickém jazyce:	DBMS, RDBMS, development of a database system, database, table, relation

Anotace v českém jazyce:

V bakalářské práci jsou zpracována teoretická východiska pro tvorbu databázových aplikací pomocí současných databázových systémů se zaměřením na relační databázové systémy.

V praktické části je popsán informační systém firmy ExlService Czech Republic, s.r.o., požadavky na vytvoření databáze hardwaru a způsob přípravy a realizace tohoto projektu pomocí RDBMS MS Access 2007.

Anotace v anglickém jazyce:

In the course of the bachelor diploma work there are elaborated the theoretical ways out to developing of database applications using current database systems with focus to relational database systems.

In the practical part there is a description of information system of the company ExlService Czech Republic, Ltd., requirements for creating of a hardware database, and a method of preparing and realization of the project using RDBMS MS Access 2007.

Literatura a prameny

CHUNG, Luke. *Microsoft Access or Microsoft SQL Server: What's Right in Your Organization?* [online]. July 2006 [cit. 2010-03-15]. Dostupné na WWW:
<<http://download.microsoft.com/download/a/4/7/a47b7b0e-976d-4f49-b15d-f02ade638ebe/SQLAccessWhatsRight.doc>>

CHUNG, Luke, HAUGHT, Dan. *When to Migrate from Microsoft Access to Microsoft SQL Server.* [online]. October 2006 [cit. 2010-04-05]. Dostupné na WWW:
<http://download.microsoft.com/download/5/d/0/5d026b60-e4be-42fc-a250-2d75c49172bc/when_to_migrate_from_access.doc>

CONOLLY, Thomas, BEGG, Carolyn, HOLOWCZAK, Richard. *Mistrovství - databáze – Profesionální průvodce tvorbou efektivních databází.* 1. vyd. Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.

KRUCZEK, Aleš. *Microsoft Office Access 2007 - Podrobná uživatelská příručka.* 1. vyd. Brno: Computer Press, 2007. 368 s. ISBN 978-80-251-1608-1.

Microsoft. *Microsoft Office Access - Nastavení nebo změna uživatelské úrovně zabezpečení aplikace Access 2003 v aplikaci Access 2007.* [online]. [cit. 2010-04-11]. Dostupné na WWW:
<<http://office.microsoft.com/cs-cz/access/HA101662271029.aspx>>

Microsoft Support. *Description of frequently used terms in Microsoft Access.* [online]. July 2004 [cit. 2010-03-24]. Dostupné na WWW:
<<http://support.microsoft.com/kb/831858>>

VIEIRA, Robert. *SQL Server 2000 - Programujeme profesionálně.* 1. vyd. Praha: Computer Press, 2001. 1170 s. ISBN 80-7226-506-7.

Seznam zkratek

DBMS	Database Management System, systém řízení báze dat, systém řízení databáze, databázový systém
DDL	Data Definition Language, jazyk pro definici dat
DML	Data Manipulation Language, jazyk pro manipulaci s daty
ERP	Enterprise Resource Planning, systém plánování podnikových zdrojů, podnikový informační systém
GUI	Graphical User Interface, grafické uživatelské rozhraní
IS	Information System, informační systém
ISAM	Indexed Sequential Access Method, index-sekvenční metoda přístupu
ODBC	Open Database Connectivity
OODBMS	Objektově orientovaný DBMS
ORDBMS	Objektově-relační DBMS
QBE	Query by Example
RDBMS	Relační DBMS
SQL	Structured Query Language, strukturovaný dotazovací jazyk
SMP	Symetrický multiprocessing
XML	Extensible Markup Language – otevřený formát pro výměnu dat definovaný konsorciem W3C

Seznam obrázků

Obr. 1 – Životní cyklus plánování databázového systému	16
Obr. 2 – Uživatelské pohledy na databázový systém	17
Obr. 3 – Topologie Access a Jet: 1 uživatel	23
Obr. 4 – Topologie Access a Jet: Více uživatelů	24
Obr. 5 – Topologie Access, Jet a SQL Server	27
Obr. 6 – Topologie Access a SQL Server	28
Obr. 7 – Uživatelské pohledy IT a finančního oddělení	32
Obr. 8 – Diagram případů užití pro uživatelský pohled finančního oddělení.....	33
Obr. 9 – Diagram případů užití pro uživatelský pohled IT oddělení.....	34
Obr. 10 – Entitně-relační diagram s entitami, relacemi a multiplicitou	37
Obr. 11 – Vytváření tabulky Associates v návrhovém zobrazení.....	39
Obr. 12 – Vytváření relací mezi tabulkami v okně Relace.....	40
Obr. 13 – Uživatelský pohled vytvořený pomocí QBE	41
Obr. 14 – Tisková sestava v návrhovém zobrazení	42
Obr. 15 – Definování uživatelských a skupinových účtů	43
Obr. 16 – Definování uživatelských práv	43

Seznam tabulek

Tab. 1 – Historický vývoj databázových systémů	8
Tab. 2 – Srovnání vlastností MS Accessu a MS SQL Serveru	26
Tab. 3 – Data využívaná různými pohledy	33
Tab. 4 – Popis případů užití pro uživatelský pohled finančního oddělení.....	34
Tab. 5 – Popis případů užití pro uživatelský pohled IT oddělení	35
Tab. 6 – Logický návrh relačních tabulek včetně primárních a cizích klíčů	38