



Uplatnění produktu MS Access v informačním zabezpečení podnikatelských subjektů

Autor: **Ondřej Filípek**

Vedoucí BP: **Dr. Ing. Václav Vostrovský**

© 2006

Prohlášení:

Prohlašuji, že jsem bakalářskou práci na téma „Uplatnění produktu MS Access v informačním zabezpečení podnikatelských subjektů“, vypracoval samostatně a použil jsem jen pramenů, které cituji a uvádím v příloženém seznamu literatury.

V Praze, dne 22. června 2006

Ondřej Filípek

Poděkování:

Rád bych touto cestou nejprve poděkoval vedoucímu bakalářské práce, panu Dr. Ing. Václavu Vostrovskému za odborné vedení a konzultace.

Dále bych rád poděkoval firmě Callida, s.r.o., za možnost využít její firemní informační prostředky a za četné odborné konzultace s jejími odborníky v oboru tvorby informačních systémů.

V neposlední řadě bych také rád poděkoval všem, kteří mi poskytli všestrannou podporu, zejména své přítelkyni Martině Boušové.

Uplatnění produktu MS Access v informačním zabezpečení podnikatelských subjektů

Souhrn:

Tato práce se zabývá databázovými technologiemi se zaměřením na relační databázové systémy. Zvláštní pozornost je poté věnována relačnímu databázovému systému společnosti Microsoft, běžně dostupnému jako součást sady kancelářských programů MS Office, a jeho možnosti uplatnění v informačním zabezpečení firemních subjektů.

V rámci diskuze ohledně využitelnosti produktu MS Access v informačním systému firem, jsou v této práci uvedeny jak kladné tak i záporné vlastnosti produktu. A vzhledem k velkým nárokům informačních systémů středních a velkých firem na některé prvky databázového systému, tato práce nedoporučuje využití produktu MS Access jako jediné součásti informačního zabezpečení firemních subjektů. A to i přes některé jeho nesporně kladné stránky (Například snadné ovládání programu, velmi dobrá spolupráce s jinými databázovými systémy či ostatními programy sady MS Office nebo velmi propracované vývojové součásti), které se zcela jistě dobře využijí při použití MS Accessu v informačním systému malých firem nebo personálních databázových systémech.

Klíčová slova:

Databázový systém; Databáze; SŘBD (Systém řízení báze dat); Informační systém; Datový model; Relace; Tabulka; Entita; Atribut; Doména; Vztah; Primární klíč; Datová normalizace; Datové modelování; Dotaz; Jazyk SQL; Kód VBA (Visual Basic for Application); Microsoft Access; Objekt MS Access.

Using product MS Access in domain of information security of corporation

Summary:

This work deals with database technology intent on relation database system. Special part of this work is focused on database system developed by Microsoft company. This program is available like part of MS Office. Corporation use MS Access as the way of information security.

In this work we can find qualities and negations of this product to discuss them. Like quality we can specify for example user friendly interface or quite good cooperation with other database systems and with MS Office. This information system is fully sufficient for small companies. We will not recommend Ms Access like exclusive information security for big companies because of their huge requirements on information systems.

Key Words:

Database system; Database, DBMS (Databáze Management Systém); Information system; Data model; Relation; Sheet; Entity; Attribute; Domain; Relationship; Primary key; Data normalisation; Data modeling; Query; SQL; Visual Basic for Applications (VBA), Microsoft Access; Object MS Access.

Obsah:

1.	Úvod.....	3
2.	Cíl práce a metodika.....	4
2.1.	Cíl práce	4
2.2.	Metodika	4
3.	Databázový systém a jeho základní pojmy	5
3.1.	Data nejsou informace	5
3.2.	Historie vývoje zpracování dat.....	5
3.3.	Charakteristika databázového systému	6
3.4.	Datové a konceptuální modely.....	7
3.4.1.	Síťový datový model.....	8
3.4.2.	Hierarchický datový model.....	8
3.4.3.	Relační datový model.....	9
3.4.4.	Objektově orientovaný datový model	10
4.	Relační datový model podrobněji	12
4.1.	Základní vlastnosti relačního datového modelu.....	12
4.2.	Terminologie relačního datového modelu	13
4.3.	Datové modelování	14
4.3.1.	Entity	14
4.3.2.	Atributy	14
4.3.3.	Domény	14
4.3.4.	Vztahy	15
4.3.5.	Diagramy entit a vztahů	15
4.4.	Datová normalizace.....	16
4.4.1.	Normální formy.....	16
4.4.2.	Jednoznačná identifikace záznamů	17
4.4.3.	Funkční závislost.....	18
5.	Produkt Microsoft Access	19
5.1.	Základní vývojové prvky programu Microsoft Access.....	19
5.1.1.	Tabulka.....	20

5.1.2.	Dotazy	21
5.1.3.	Formuláře	23
5.1.4.	Sestavy	24
5.1.5.	Datové stránky	25
5.1.6.	Makra	26
5.1.7.	Moduly	26
5.2.	Zabezpečení dat.....	27
5.2.1.	Kódování a dekodování souboru databáze.....	28
5.2.2.	Zobrazení nebo skrytí objektů v okně databáze.....	29
5.2.3.	Použití možností pro spuštění	29
5.2.4.	Použití hesla databáze	29
5.2.5.	Použití individuálního řízení přístupu.....	30
5.2.6.	Zabránění uživatelům v provádění replikace a nastavení databáze	31
5.2.7.	Zabezpečení kódu jazyka VBA.....	31
5.2.8.	Nedostatky v zabezpečení programu MS Access	32
5.3.	Víceuživatelské rozhraní Access.....	33
5.4.	Access ve spolupráci s jinými programy	34
5.5.	Náhled na Microsoft Office Access 2007	35
6.	Závěr	37
7.	Seznam použité literatury.....	39

1. Úvod

V dnešním světě informačních technologií již takřka nikdo nepochybuje o významu získávaných informací, které jsou člověku k dispozici z mnoha informačních kanálů, jako například televize, rádia, novin, internetu, ale mnohdy také i z volné diskuse mezi přáteli. V tomto velkém množství informací, kterými jsme den co den možná až zahlceni, jsou však některé informace pro konkrétního jedince téměř nepotřebné a některé naopak životně důležité. Vezměme si například manželský pár chystající se jednoho jarního sobotního odpoledne na večerní divadelní představení. Tento pár při snaze vhodně zvolit množství svého oblečení a způsob dopravy do divadla, jistě přivítá informace o předpokládaném vývoji počasí na daný večer nebo informace o dopravní situaci ve městě, kde se nachází divadlo, do kterého mají namířeno. Spolu s těmito praktickými informacemi jistě ocení i jiné, méně potřebné skupiny informací, které by mohly k dané cestě za kulturou využít. Tento manželský pár však daného odpoledne nikterak kladně neohodnotí například informaci o průměrné hodnotě naměřeného tlaku vody ve vodovodní síti zásobující pražské domácnosti, nebo snad informaci o snižující či zvyšující se efektivitě práce pracovníků firmy Callida, s.r.o. (Samozřejmě pokud ani jednoho z daného manželského páru se toto téma nikterak netýká profesně.). Je zřejmé již jasné, že v této souhrnné záplavě informací, které máme k dispozici, musíme všichni dobře vybírat ty, které jsou pro naše využití co možná nejpotřebnější nebo alespoň přímo související.

Pokud tento fakt nyní převedeme do českého firemního prostředí plného konkurenčního boje o zákazníka a dalších neméně důležitých prvků tvořících volný konkurenční trh, můžeme konstatovat, že české firmy (ale v podstatě nejen české) jsou, v současné době postaveny před důležitý úkol. A to vybudovat si kvalitní a přehledný informační systém, který jim jednak usnadní tok informací ve firmě, ale také napomůže získávat další životně důležité informace z interních, ale i externích zdrojů. K tomuto úkolu jim může velmi pomoci vhodně zvolený databázový systém, který se v dnešním trendu informačních technologií, stává vlastním jádrem informačního systému, nebo alespoň často jeho nepostradatelnou částí.

2. Cíl práce a metodika

2.1. Cíl práce

Hlavním cílem této práce je popsání a zhodnocení základních vývojových principů jednoho finančně velmi snadno dostupného databázového prostředku s názvem Microsoft Access a také zhodnocení možnosti jeho uplatnění v informačním zabezpečení firemních subjektů.

Vzhledem k tomu, že MS Access je systém pro řízení relační databáze, bude pro tyto výše uvedené účely dále v této práci také nezbytné uvést některé teoretické základy databázových systémů, problematiku relačního datového modelu, včetně ujasnění terminologie k tomuto modelu vázané, a v neposlední řadě také problematiku datové normalizace a datového modelování.

2.2. Metodika

Při tvorbě této bakalářské práce bylo postupováno v několika etapách. V první etapě bylo shromážděno co největší množství informací souvisejících s daným tématem práce. V tomto případě byla preferována knižní literatura, případně odborné internetové servery. Druhou etapou bylo následné nastudování problematiky tématu a ověření věrohodnosti a platnosti shromážděných zdrojů informací. To bylo většinou provedeno konfrontací těch informačních zdrojů, které poskytují shodné informace, mezi sebou. Nebo ověřením některých postupů v praktickém využití. Ve třetí etapě byly informace vztahující se k tématu práce opět seskupeny a byl vytvořen koncept bakalářské práce. Poslední etapou bylo sepsání samotné bakalářské práce.

3. Databázový systém a jeho základní pojmy

3.1. *Data nejsou informace*

Účelem databázových systémů, jakým je Microsoft Access, je přeměna dat na informace. Mnoho lidí používá tyto dva výrazy, aniž by je rozlišovali, ale přitom je mezi nimi propastný rozdíl, pokud uvažujeme o informacích jako o vědomostech. Data jsou souborem údajů a informace jsou data organizována a prezentována ve smyslu užitečnosti pro rozhodování [1].

Neboli data jsou údaje získané pozorováním nebo měřením. Oproti tomu informace jsou smysluplné interpretace dat a vztahů mezi nimi. Informace nám umožňují a usnadňují rozhodování a lze je definovat vždy jen ve vztahu k systému, pro který jsou určena. Velké množství dat tudíž nutně nemusí znamenat, že získáme požadovanou informaci [2].

Informace je možné posuzovat z hlediska *kvantitativního* (číselné vyjádření množství informace obsažené v dané zprávě), *kvalitativního* (podílu na snížení neinformovanosti příjemce) a *pragmatického* (praktické hodnoty a užitečnosti) [3].

Přičemž však platí, co je uvedeno v [4] a [5], že pro transformaci dat do informací slouží databázové systémy.

3.2. *Historie vývoje zpracování dat*

Jak je uvedeno v [5], bylo v průběhu 50. let zpracování dat organizováno způsobem, kdy součástí každého aplikačního programu byla i vlastní data a kdy každý program pracoval zvlášť s použitím omezeného počtu dat.

Od první poloviny 60. let si poté vývoj uživatelských programů, které již začaly pracovat s větším objemem dat, vyžádal oddělení těchto dat od vlastních programů. Následným mohutným nárůstem množství dat (který přetrvává do dnešní doby), bylo zapříčiněno vytvoření *systémů pro zpracování souborů*. Data v těchto systémech již mohla být zpracovávána více programy současně, což v dané době představovalo výrazný vývojový krok vpřed.

V druhé polovině 60. let byly však tyto systémy pro zpracování dat na jejich pomyslném výsluní vystřídány prvními *systemy řízení bází dat (SŘBD, nebo také DBMS – database management system)*. V této době měly SŘBD především za úkol odstranit některé z nedostatků svých předchůdců. Ukázalo se však, že nový způsob uložení dat v **databázi** (většinou centralizovaně na jednom místě) a dosažení určité nezávislosti dat na uživatelském programu (respektive nezávislost uživatelských programů na datech) prostřednictvím převratného komunikačního rozhraní, se časem stali základními prvky databázové technologie, které se v SŘBD používají dodnes. Pro další podrobnější popis historie vývoje zpracování dat mi vzhledem k rozsahu této práce prosím dovolu odkázat například na [6], kde se autor tímto tématem v jedné části své knihy zabývá podrobněji.

3.3. Charakteristika databázového systému

Zatím jsme se dozvěděli o existenci SŘBD a databázi (DB), které dohromady vytvářejí databázový systém (DBS). Schematicky lze tento vztah vyjádřit jako $SŘBD + DB = DBS$ [5].

Stručně řečeno, databázové systémy slouží k modelování určité části reálného světa, na který se díváme z konkrétního úhlu pohledu a který také nazýváme *prostor problému (nebo také problémová doména)*.

Každý prostor problému je ve své podstatě velmi komplikovaný a z našeho pohledu také chaotický. Proto vytváříme takzvaný *datový model (nebo také databázový model)*, který nám má pohled na prostor problému usnadnit. Z toho vyplývá, že datový model ještě není výsledkem modelování, jak by se mohlo zdát, nýbrž jde o prostředek modelování. Výsledkem aplikace tohoto prostředku je následně **databázové schéma (nebo také schéma databáze)**.

V publikaci [5] se uvádí, že databázové schéma je tedy fyzické rozvržení databáze (čímž se rozumí seznam tabulek, pohledů, nebo také databázové procedury aj.) a vzniká z konceptuálního modelu – tento termín je blíže vysvětlen v kapitole 3.4. Toto schéma je tedy nadále pouze symbolické (myšlenkové) a proto jej také můžeme nazývat *logické schéma databáze*.

Pod pojmem *fyzické schéma databáze* autor publikace [5] definuje popis uložení dat na vnější paměťové médium. Můžeme tedy říci, že přeměnu logického databázového schématu na fyzické databázové schéma pro naše účely zprostředkovává *databázový stroj*¹ (pomocí programového kódu nebo v nějakém interaktivním grafickém prostředí jaké má například Microsoft Access), který vytvoří určité fyzické objekty, do nichž posléze můžeme ukládat data.

Databázi se poté nazývá sjednocení vlastních dat a takto vytvořené struktury. A jestliže z definice SŘBD víme, že data jsou oddělena od uživatelského programu, je již jasné, že *databázovým systémem* je také spojení komponent *databáze*, *databázového stroje*, a *uživatelského programu*, jak je tomu například u produktu Microsoft Access.

3.4. Datové a konceptuální modely

Pod pojmem datový model se většinou rozumí konceptuální (myšlenkový) popis prostoru problému. Datový model zahrnuje definici *entit*, jejich *atributů*, *omezení entit*, *domén* (oborů hodnot), *vztahů* mezi entitami a veškerá *omezení platná pro tyto vztahy* [7].

Podle některých publikací (například [8]) se datový model rozděluje na takzvaný *konceptuální datový model* – řešení struktury dat je obecné a nebere v potaz typ databázového systému. Dále na *logický datový model* – v tomto typu modelu se již konkretizuje konceptuální datový model na určitý databázový systém. A nakonec na *datový model v definičním jazyce* – jedná se o přetvoření logického datového modelu na definiční jazyk konkrétního databázového systému.

Nyní si zde stručně popíšeme čtyři, dnes již nejrozšířenější, datové modely a to síťový datový model, hierarchický datový model, relační datový model a v neposlední řadě i objektově orientovaný datový model.

¹ V případě produktu Microsoft Access se jedná o databázový stroj Microsoft JET. Více se tomuto prvku MS Access budeme věnovat v kapitole 5.3.

3.4.1. Síťový datový model

Patří do skupiny souborově orientovaných datových modelů a je historicky nejstarší. Jeho architektura byla zveřejněna v roce 1971. Základem síťového datového modelu jsou propojené množiny záznamů, přičemž záznamy jednoho typu jsou obvykle uloženy v jednom souboru. Celá databáze je složena ze dvou hlavních množin. První z nich je **množina záznamů** (obvykle rozložená do více souborů), druhou je **množina spojek**, přičemž spojka je zvláštní typ záznamu o dvou položkách obsahujících fyzické adresy záznamů, které spojují [8].

Atributy záznamů v síťových modelech mohou být čtyř druhů – **jednoduché, opakující se, složené nebo složené opakující se**. V nejjednodušším případě jde o dále nedělitelné (atomické) hodnoty podílející se na existenci záznamu (např. TITUL nějaké knihy). Opakující se atribut je např. AUTOR v případech titulů s více autory. Jako složený můžeme vidět atribut ADRESA, rozlišujeme-li v ní jednotlivé části (ULICE, PSČ apod.). ADRESA také může být složený opakující se atribut, jde – li např. o čtenáře – instituci existující v několika pobočkách. Poznamenejme ještě, že opakujícím se atributům se říká v terminologii souborů tradičně *opakující se skupiny (repeating groups)* [5].

3.4.2. Hierarchický datový model

Hierarchické databázové modely jsou opět souborově orientované modely. Z matematického hlediska jsou hierarchie speciálním případem síťového modelu. Diagram datové struktury je v hierarchickém přístupu stromem, případně lesem (několik stromových struktur). Typy záznamu vyskytující se v hierarchických modelech se podobají typům záznamů známých ze síťových modelů, pouze s tím rozdílem, že jsou jednodušší – obsahují pouze jednoduché atributy, případně opakující se jednoduché atributy [5].

Typ vztahu je definován přesně nad dvěma typy záznamu. Přebírá se terminologie známá ze stromových struktur, tj. místo vlastník – člen se používá označení rodič – dítě, otec – syn apod. [5].

Protože ke každému typu záznamu existuje nejvýše jeden typ rodiče, není nutné označovat typy vztahů v hierarchiích jménem. Při použití hierarchického databázového modelu okamžitě vystane otázka, jak se vyrovnat s problémem vztahů M:N. Jsou možná dvě řešení. Buď duplikované záznamy (tj. k jednomu záznamu typu dítě se vytvoří duplicitní záznam tak, aby se daly realizovat všechny vztahy s odpovídajícími rodičovskými záznamy), nebo více definičních stromů (tj. jeden typ vztahu M:N se rozloží na dva 1:N) [5].

3.4.3. *Relační datový model*

Relační datový model byl odborné veřejnosti poprvé představen **Dr. E. F. Coddem** v roce 1970 a to v publikaci nazvané „*A Relation Model for Large Shared Data Banks*“ vydané nakladatelstvím CACM. Tento matematik a významný pracovník firmy IBM, založil relační datový model na souboru základních matematických principů odvozených z teorie množin a matematické logiky (konkrétně predikátového kalkulu 1. řádu), kde relace² tvoří základní strukturu při interpretaci predikovaných symbolů. Relační datový model můžeme (ze své podstaty) zařadit do skupiny souborově orientovaných datových modelů a následně formálně rozdělit na modely *normalizované* a *nenormalizované (NFNF – non fires normal form)*, přičemž normalizované dále dělíme na „*klasické*“ *relační* a *objektově relační* (teoreticky se takto dají rozdělit i nenormalizované datové modely).

Tento model v podstatě definuje způsob, jakým je možné data reprezentovat (tedy strukturu dat), způsoby jejich ochrany (neboli integrity dat) a operace, které můžeme nad těmito daty provádět (manipulace s daty) [7].

S relačním datovým modelem také souvisí prostředky, jako například *neprocedurální relační dotazovací jazyky*, které vznikly právě aplikací matematické logiky relačních datových modelů. Dr. E. F. Codd spolu s relačním datovým modelem totiž vytvořil i jazyk původně nazvaný **Structured English QUery Language** neboli **SEQUEL**

² Jedná se o pojem matematické relace, což je modifikovaný pojem Relace R^* nad množinou atributů Ω . Bližšímu popisu tohoto termínu, stejně jako matematickým principům relačních datových modelů, se věnuje autor v publikaci [5].

(vyslovováno „síkwel“ a přeloženo doslova jako „strukturovaný anglický dotazovací jazyk“). Postupem času, se toto označení zkrátilo na *SQL* neboli *Structured Query Language*³. V 80. letech poté americká standardizační organizace *ANSI (American National Standards Institute)*, vstoupila do vývoje SQL a vytvořila specifikaci jazyka SQL pod názvem *ANSI-SQL*⁴.

V dnešní době již existuje spousta jazykových mutací od tohoto standardu ANSI-SQL a mnohdy i české firmy využívají „Open-source“ SQL jazyků, které si přetvářejí dle vlastní potřeby využití a tím v některých bodech porušují tento standard. Dá se tedy říci, že na tento standard se již v dnešní době nesmíme dívat jako na fanatické dogma, ale jako na doporučený standard, který je třeba dodržovat, má li to nějaký smysl.

Další informace o tomto datovém modelu naleznete v následující kapitole 4., kde se relačnímu datovému modelu věnuji podrobněji.

3.4.4. Objektově orientovaný datový model

Objektový datový model vznikl jako reakce na určité nedostatky relačního datového modelu. V rámci objektové orientace v databázích lze rozlišit dva základní směry vývoje. První z nich jsou „nerelační“(čisté) *objektové databáze*, které jsou založené na tzv. objektovém datovém modelu. Druhým z nich jsou tzv. „relačně“ *objektové databáze*, které využívají většinu vlastností relačního datového modelu, který potom působí jako „nosič“ pro ukládání objektů [9].

Díky objektově orientovaným mechanismům, mají objektové databáze oproti ostatním zde popisovaným typům ještě například schopnost skládání a zapouzdření.

³ V českých firmách se mnohdy setkáte s odborníky, kteří chybně používají termín „síkwel“ pro označení specializovaného SQL serveru (výkonného počítače, který je specializován pouze pro databázové systémy) a termínem SQL označují obecně skupinu neprocedurálních dotazovacích jazyků (včetně jazyku Quel nebo QBE).

⁴ Toto vytvoření standardu SQL se později ukázalo být v relačních databázových systémech velmi významným krokem, protože standard znamenal zvýšení kompatibility mezi jednotlivými výrobci databází (mezi první výrobce patří v 70. letech firmy Oracle a Sybase, později také IBM se svými systémy SQL/DS a DB2).

Objektově orientované datové modely poté můžeme ještě dělit na modely *hypertextové*, *fulltextové*, *sémantické sítě* a *klasické (třídně instanční)*.

4. Relační datový model podrobněji

O vzniku, zařazení a prvotním popisu relačního datového modelu a jazyka SQL informuje odstavec 3.4.3 této práce.

4.1. Základní vlastnosti relačního datového modelu

Různé odborné publikace uvádí rozličné vlastnosti relačního datového modelu, avšak na těch zásadních se ve výsledku shodují. Uvedme nyní shrnutí těchto zásadních vlastností tak, jak jsou výstižně jmenovány v [8]:

- Hodnoty v tabulkách musí být **atomické**, tj. nedělitelné (nesmějí se skládat z dalších hodnot).
- Hodnoty musejí být **skalární**, tj. jednorozměrné⁵.
- Hodnoty v tabulkách existují jako prvky jednotlivých *domén* (sloupců). Všechny prvky dané domény musejí být mezi sebou porovnatelné a musejí náležet jednomu datovému typu.
- Pro práci s tabulkami se používá operací **výrokové logiky**.
- V každé tabulce hodnoty v jednom nebo více sloupcích slouží k jednoznačné identifikaci řádků mezi sebou. Tyto hodnoty jsou označeny jako **primární klíče** tabulky.
- V některých tabulkách hodnoty v jednom nebo více sloupcích mají vztah k hodnotám v jiných tabulkách (nebo ve zvláštním případě i k hodnotám vlastní tabulky). Tyto hodnoty jsou označeny jako **cizí klíče**.
- V tabulkách lze definovat podmnožiny řádků nebo podmnožiny sloupců. Operace vybírající podmnožinu řádků je označována jako **selekce** tabulky. Operace vybírající podmnožinu sloupců je označována jako **projekce** tabulky.
- Více tabulek lze kombinovat mezi sebou jako běžné množiny pomocí operací sjednocení, rozdíl, průnik a kartézský součin množin. Kombinace kartézského součinu a selekce se nazývá **spojení** tabulek.

⁵ To znamená, že se v každém řádku a sloupci dané relace nesmí nacházet více než jedna hodnota.

4.2. Terminologie relačního datového modelu

Základním pojmem relačních databází je termín *relace*, který představuje uložená data, reprezentovaná v pravidelně uspořádaných strukturách. V některých publikacích (například [12]) je tento termín vysvětlován jako definice vztahu mezi jednotlivými tabulkami, ve kterých jsou uložena data. To je však nepřesné tvrzení, neboť pro definici vztahu mezi relacemi⁶ (z neformálního pohledu může relace být reprezentována tabulkou) máme jiné vyjádření, a to kardinalita vztahu (tento pojem je popsán v kapitole 4.3).

Relace však ve skutečnosti nepotřebují žádnou fyzickou reprezentaci. Určitá množina záznamů se tak může opravdu mapovat na reálnou, fyzickou tabulku, stejně tak ale může vyjadřovat pouze jisté sloupce, vybrané z několika tabulek a doplněné o několik vypočtených polí, které se samozřejmě nikam fyzicky neukládají⁷. Relací může být v podstatě cokoliv, co se ukládá do struktury řádků a sloupců a co obsahuje skalární hodnoty. Existence určité relace je tedy zcela nezávislá na její fyzické reprezentaci [7].

S pojmem relace souvisí i termín *vektor hodnot (neboli vektor souřadnic – anglicky tuple)*, který představuje jednotlivé řádky dat v relaci. Poté platí, že každý řádek je n-rozměrný vektor, kdy počet vektorů určuje *kardinalitu relace*. Sloupce v relaci se následně nazývají *atributy* a jejich počet v dané relaci dat určuje její *stupeň*.

Dále je pravidlem, že množiny vektorů hodnot jsou v relaci neuspořádané (neboli na pořadí jednotlivých řádků nesmí záležet) a že podmínkám relace vyhovuje množina o žádném a více prvcích (neboli relace může být také prázdnou množinou řádků). Naprosto nezbytnou podmínkou je poté **jednoznačná identifikace** vektorů hodnot, neboť v množině vektorů hodnot nesmějí být dva nebo více naprosto shodných vektorů.

⁶ Dr. E. F.: Codd při tvorbě relačního datového modelu zvolil termín „relace“, protože tento termín nebyl tak zatížen vůči jiným pojmům. V českém jazyce se však díky nedokonalému překladu mnohdy zavádí některé nepřesnosti v terminologii.

⁷ Takovým případem relace může být například takzvaný „virtuální pohled“, používaný v produktu Microsoft Access, který na první pohled vystupuje jako fyzická tabulka, avšak není tvořen fyzickou tabulkou, nýbrž dotazem SQL jazyka. Tento princip, kdy prováděnou operací na relaci v databázi získáváme další relaci, nazýváme mechanismus *uzávěru*.

4.3. Datové modelování

V kapitole 3.3 jsme se již setkali s důvodem vytváření datového modelu. V kapitole 3.4 byla následně uvedena definice a členění datového modelu. V kapitole 4 si nyní přiblížíme jednotlivé součásti datového modelu tak, jak byly definovány v kapitole 3.4.

4.3.1. Entity

Entitami v datovém modelování rozumíme konkrétní data, opsaná z reálného světa. Jedná se o jednotlivé podstaty opsané z reálného světa do množiny dat, které chceme uchovávat v databázi. Takovou entitní množinou může být naprosto cokoliv, například evidovaný lék v databázi lékárny nebo zaměstnanec v nějaké firemní evidenci.

4.3.2. Atributy

Každá entita má ze své podstaty nějaké vlastnosti. Těmto vlastnostem se v datovém modelování přezdívá atributy. Například v případě výše zmíněné entitní množiny *zaměstnanec* mohou být jednotlivými atributy *jméno a příjmení zaměstnanec*, *adresa bydliště zaměstnanec* vystupující jako *město, ulice, popisné číslo domu, poštovní směrovací číslo*, a jiné atributy, které by nás jistě napadli. V případě určování jednotlivých atributů entit, je však zapotřebí určovat jejich význam k použití těchto atributů v modelované databázi (jinak by mohlo docházet k uchování naprosto nepotřebných a mnohdy obecně jasných dat – například v evidenci zaměstnanců nás pravděpodobně atribut zaměstnanec typu *tloušťka lebeční kosti* jistě zajímat nebude, avšak v evidenci pacientů neurochirurga může být opodstatněná).

4.3.3. Domény

Doména neboli obor hodnot je množina přípustných platných hodnot, které smí určitý atribut obsahovat. Obory hodnot se často zaměňují s *datovými typy*. Datový typ je totiž fyzický pojem, zatímco obor hodnot je čistě logický. „Číslo“ je tedy datový typ, zatímco „Věk“ představuje obor hodnot [7].

4.3.4. Vztahy

Jak již bylo zmíněno v kapitole 4.2, definici vztahů mezi relacemi se říká **kardinalita vztahu**. Termínem kardinalita vztahu se v podstatě vyjadřuje *násobnost* vztahu mezi dvěma entitami (počet výskytů dvou entit v daném směru jejich vztahu). Typ základní kardinality vztahu máme trojí. Známe kardinalitu **jedna k jedné (1:1)**, **jedna k více (1:N)** a **více k více (N:M)**.

V souvislosti s účastí daných entit ve vztahu můžeme dále rozeznávat ještě kardinality **s účastí úplnou** (dané entity nemůžou existovat bez účasti ve vztahu), nebo **s účastí částečnou** (daná entita se ve vztahu nemusí nacházet – taková entita může existovat i samostatně, neboli izolovaně).

Dále můžeme o kardinalitě vztahu říci, že tak jako počet atributů u relace definuje stupeň relace, počet entit v kardinalitě vztahu určuje **stupeň vztahu** (je důležité tyto pojmy nezaměňovat, neboť jak bylo řečeno již v kapitole 4.2, dochází nepřesným překladem pojmů relace a vztah mezi relací k chybám v terminologii⁸). Typy stupně vztahů známe:

- **Unární** – relace v takovémto vztahu je vázána sama na sebe a tudíž se jedná o vztah s jedním účastníkem. Takový typ vztahu je například vazba voják důstojník, kdy voják má skoro vždy nad sebou nějakého důstojníka, který je však také voják.
- **Binární** – jedná se o nejčastěji vyskytující se vztahy o dvou účastnících.
- **Ternární** – takovéto vztahy mají obvykle tvar kdy „někdo“ dělá „něco“ pro „někoho“. Jedná se tedy o vztahy o třech účastnících.

4.3.5. Diagramy entit a vztahů

Model entit a vztahů, který popisuje data jako entity, atributy a vztahy, zavedl poprvé Peter Pin Shan Chen v roce 1976. Současně navrhl metodu jeho zobrazení do diagramů, které nazval **diagramy entit a vztahů** (Entity Relationship Diagrams – E/R Diagram), a které se záhy rychle rozšířily a staly se obecně uznávanými. V E/R diagramech

⁸Pro porovnání a ujasnění rozdílu mezi *stupněm relace* a *stupněm vztahu* zde tyto termíny uvádíme v kontrastu.

se entity popisující pomocí obdélníků, atributy se vyjadřují jako elipsy a vztahy se znázorňují pomocí kosočtverců [7].

4.4. Datová normalizace

Pojmem datová normalizace rozumíme převod databáze do *normální formy*. V podstatě se jedná o proces zjednodušování struktury postupným nahrazováním relací tak, aby došlo k odstranění problémů s aktualizací anomálií.

Proces tohoto převodu relace, musí vyhovovat *beztrátové dekompozici*, která zaručuje v případě zpětného převodu relace, že nedojde k jakékoli ztrátě dat (proces je tudíž reverzibilní).

4.4.1. Normální formy

Při datové normalizaci upravujeme databázi tak, aby vyhovovala určitému stupni normální formy (NF). Přičemž jednotlivé stupně NF (kromě první NF, která je základní) jsou vždy rozšířením NF nižšího stupně, tudíž například 3.NF vyhovuje zároveň i 2.NF a 1.NF.

První normální forma uvádí, že by každé pole všech tabulek databáze mělo obsahovat pouze dále nedělitelnou hodnotu, neboli zaručuje takzvanou *atomičnost dat* (v tabulce jsou všechny data samostatná a na sobě nezávislá). Na této úrovni NF se prakticky určují takzvané primární klíče (viz kapitola 4.4.2) jednotlivých relací, dále se veškeré opakované skupiny dat přesunují do nově vytvořených relací s určitou vazbou na relaci, ze které vznikly a těmto nově vytvořeným relacím se opět určují primární klíče.

Druhá normální forma říká, že daná relace je v 2.NF, jestliže je v 1.NF⁹ a dále za předpokladu, že všechny její neklíčové atributy jsou funkčně závislé (tento pojem je objasněn v kapitole 4.4.3) na celém klíči¹⁰.

*Třetí normální forma*¹¹ je opět rozšířením 2.NF a to o pravidla, že všechny neklíčové atributy relace (na kterou uplatňujeme 3.NF) musí být *vzájemně nezávislé*¹², a že daná

⁹ Jedná se tedy o rozšíření předchozí normální formy.

¹⁰ Jak je řečeno v kapitole 4.4.2, *primární i kandidátní klíče* mohou být *složené*.

relace nesmí obsahovat jakékoli odvozené údaje. Dá se tedy říci, že jestliže relace splňuje 3.NF, pak všechny její atributy jsou závislé pouze na celém klíči (na ničem jiném).

Někde mezi 3.NF a 4.NF je normální forma, známá jako **Bouyce-Coddova**. Tato NF je vlastně rozšířením 3.NF a má za úkol řešit situace s několika překrývajícími se kandidátními klíči¹³. Kandidátní klíče se přitom mohou překrývat, jestliže všechny kandidátní klíče v tabulce budou složenými klíči a pokud v tabulce budou existovat minimálně dva kandidátní klíče a každý z nich bude mít alespoň jeden atribut společný s jiným kandidátním klíčem.

Čtvrtá normální forma v sobě opět obsahuje předchozí stupeň NF a navíc požaduje, aby složený klíč nebyl tvořen z nezávislých dat. Je to situace, kdy relace sice vyhovuje 3.NF, avšak jeden atribut složeného klíče je sám o sobě závislý na jiných sloupcích klíče.

Pátý stupeň normální formy (nad rámec 4.NF) definuje stav relace, kdy složený klíč této relace nesmí obsahovat *párové cyklické závislosti*. Obsahuje-li je, musíme tyto závislosti odstranit tak, že je extrahujeme do samostatných relací, které nám umožní zachovat informaci o zmíněných závislostech i v případě, že původní relace neobsahuje záznam s odpovídajícími hodnotami jeho složeného klíče.

4.4.2. Jednoznačná identifikace záznamů

V kapitole 4.2 jsme se dozvěděli o potřebě jednoznačně identifikovat vektor hodnot relace. Aby tato jednoznačná identifikace byla možná, musí se v dané množině vektoru hodnot vyskytovat alespoň jeden atribut, který je schopen v jakémkoli okamžiku jednoznačně identifikovat všechny možné množiny vektorů hodnot relace. Takovéto atributy se poté nazývají **kandidátní klíče**.

¹¹ V praxi se v naprosté většině setkáme nejvýše s 3.NF. Uplatnění vyššího stupně NF je výjimečné a mnohdy takřka teoretické. V mnoha odborných publikacích se dokonce uvádí pouze první tři stupně normální formy.

¹² Tj. stav, kdy žádný atribut nezávisí na neklíčovém atributu.

¹³ Zakazuje tedy jakoukoli funkční závislost mezi kandidátními klíči.

Protože však může nastat situace, že kandidátních klíčů je několik, je v tomto případě zapotřebí určit jeden, takzvaný **primární klíč**, který bude jednoznačně určovat danou identifikaci množiny vektorů hodnot a pomocí kterého se bude definovat vazba relace. Kandidátní klíč (respektive primární klíč), však nemusí být představován pouze jedním atributem (Takový klíč, který je tvořen jediným atributem, je označován přívlastkem **jednoduchý**), neboť jej lze také definovat jako složení několika atributů dané relace – takový klíč označujeme jako **složený**.

V praxi poté může nastat situace, kdy ani jeden z atributů relace nevyhovuje podmínkám jednoznačné identifikace množiny vektoru hodnot. Pak je zapotřebí takovýto atribut do relace přidat¹⁴ (manuálně jej vytvořit) a zajistit, aby následně fungoval jako primární klíč relace.

4.4.3. Funkční závislost

Jestliže je dán vektor hodnot T se dvěma množinami atributů $\{X_1, \dots, X_n\}$ a $\{Y_1, \dots, Y_n\}$ (tyto množiny nemusí být disjunktí) a pro každou platnou hodnotu X existuje pouze jedna platná hodnota vektoru Y , pak říkáme, že množina Y je **funkčně závislá**, na množině X . Pokud je množina X kandidátním klíčem, musí být na ní funkčně závislé veškeré ostatní atributy Y (to přímo vyplývá z definice kandidátního klíče). Jestliže množina X není kandidátním klíčem a funkční závislost není triviální (tedy $Y \subsetneq X$), pak relace obsahuje určitou míru redundance a musíme tedy provést další normalizaci [7].

¹⁴ Databázové stroje různých databázových systémů mají možnost takovéto „manuálně vytvořené primární klíče“ do relací tvořených databází vkládat a automaticky generovat jejich hodnoty tak, aby odpovídaly pravidlům primárních klíčů. Například Microsoft Access používá takovýto speciální atribut, který využívá datový typ nazvaný „AutoIncrement“ (v české verzi MS Access se používá termín „Automatické číslo“).

5. Produkt Microsoft Access

Microsoft Access řadíme do skupiny souborově orientovaných databází a to i přesto, že nepoužívá pro každou tabulku jeden samostatný soubor jako např. dBase a jemu podobné. Access totiž pro způsob uložení a přístupu k datům využívá *indexsekvenční metodu přístupu (Indexed Sequential Access Method, ISAM)*, jenž je právě pro souborově orientované databáze typická. Nejedná se tedy o pravý relační databázový systém, ale pouze o databázi typu ISAM.

V souvislosti se soubory, do kterých MS Access ukládá své databáze, uveďme, že v MS Accessu existují buďto klasické *databáze Accessu (soubory *.MDB)*, nebo takzvané *projekty Access (soubory *.ADP)*, které však pracují s databázemi Microsoft SQL serveru prostřednictvím architektury komponent OLE DB. Popis těchto projektů Access by se však zabýval především prvky MS SQL serveru, který není tématem této práce, a proto se jimi zde zabývat nebudeme.

5.1. Základní vývojové prvky programu Microsoft Access

V kapitole 5.1 si přiblížíme jednotlivé základní vývojové prvky produktu Microsoft Access. Těmto prvkům však v terminologii spjaté s MS Access přezdíváme *objekty*¹⁵. Mezi základní objekty patří *tabulky, dotazy, formuláře, sestavy, datové stránky, makra a moduly*.

Vzhledem k rozsahu této práce, se však popisy podrobných postupů práce v těchto základních objektech zabývat nebudeme. Pro tento podrobnější popis můžeme odkázat na rozsáhlejší publikace věnující se mimo jiné také této problematice. Jsou to například publikace [1], nebo [11].

Dále v této kapitole věnujme především popisu verze 9 (MS Access 2000), verze 10 (MS Access 2002) a případně i verzi vyšší.

¹⁵ Toto označení svádí ke spojení MS Accessu s objektově orientovanými databázemi, avšak nenechme se zmást, toto propojení je pouze slovní.

5.1.1. Tabulka

Tabulka je základním objektem databázového prostředí. Plní především funkci správce dat, které se do ní vkládají. Informace, které je potřeba evidovat, se do tabulek vkládají ve formě dat. Data se mohou dále zpracovávat a je tak možno získávat další informace – data v jiné podobě. S daty se dále může aktivně pracovat jak na základní úrovni – přímo v tabulkách, tak na vyšších úrovních – prostřednictvím ostatních hlavních objektů (*dotazy, formuláře, sestavy*). V tabulce nelze vzhledem k základní funkci provádět jakékoli výpočty tak, jak je zvykem v prostředí tabulkového editoru, které tak často tabulka asociuje. K výpočtům jsou určeny jiné hlavní objekty, jako jsou *dotazy*, nebo *formuláře*. [10].

V publikaci [7] definuje autor tabulku také jako *fyzickou instanci určité relace v databázovém schématu*.

Tabulky lze v MS Accessu vytvářet čtyřmi způsoby:

- **Prostřednictvím návrhového zobrazení** – jedná se o jednoduché a intuitivní prostředí MS Accessu, ve kterém se tabulky definují manuálně. Pro označení jednotlivých sloupců je zde použito výrazu „pole“.
- **Pomocí průvodce** – výhodou je především velmi intuitivní a jednoduché zadání struktury tabulky, které jistě přivítají začátečníci v oboru vývoje databází.
- **Prostřednictvím VBA kódu** – tento způsob je, oproti vytváření tabulek prostřednictvím průvodce, určen pro zkušené vývojáře, kteří ovládají znalost tvorby *VBA (Visual Basic for Application)* kódu, o kterém bude ještě bližší zmínka v kapitole 5.1.7
- **Za pomoci DDL jazyka SQL** – tato možnost spočívá v použití **DDL (Data Definition Language)** jazyka SQL ve zvláštních typech dotazů, v takzvaných **definičních dotazech**. Tato technika se v praxi často kombinuje s vytvářením tabulek prostřednictvím programového VBA kódu, ze kterého lze příkaz SQL také spustit.

Při vytváření tabulek v MS Accessu, je možné se setkat s některými odlišnostmi spojenými s **datovými typy sloupců**. V MS Accessu se totiž nevyskytuje v praxi běžně používaný datový typ pro řetězec pevné délky (tzv. CHAR), ale pouze pro znakový

řetězec proměnlivé délky (tzv. VARCHAR), zvaný *Text*. Tento fakt je zapotřebí vzít v úvahu při volbě primárního klíče, protože pokud je primární klíč v datovém typu VARCHAR, bude se zvyšováním objemu dat v databázi snižovat výkonnost dotazů na danou tabulku. Příčinou tohoto efektu je použité UNICODE kódování¹⁶ v datovém typu *Text*. Dalším nestandardním datovým typem v MS Accessu je typ *Object OLE*, který se využívá k ukládání různých OLE prvků (například obrázky, zvuky, videa) nebo datový typ nazvaný *Hypertextový odkaz* sloužící k ukládání odkazu souboru uloženého na síti. Posledním nestandardním datovým typem je *Memo*, do kterého může uživatel vložit nestandardně až 32 000 znaků.

Pro efektivnější zpracování dotazů (které se dotazují na tabulku s velkým množstvím dat) je možné definovat takzvané *indexy*. Ve většině případů se vytváří tyto indexy nad *jedním sloupcem* tabulky, někdy i nad *více sloupci*.

Jako každý systém má i MS Access svá omezení. Mezi mnohými zde uvedme ty nejvýznamnější:

- Tabulka může mít **maximální počet 255 sloupců a 32 indexů**.
- Index může být definován **maximálně nad 10-ti sloupci a součet délek těchto sloupců nesmí přesáhnout 255 bajtů**.
- **Řádek v tabulce nesmí být delší než 4KB** (s výjimkou datových typů *Memo* a *Objekt OLE*).
- **Počet záznamů je dán maximální velikostí databáze 2GB**. Situace plynoucí z tohoto omezení však lze vyřešit spojením několika databází do jedné hlavní databáze, což přináší neschopnost zajistit na úrovni hlavní databáze *referenční integritu*.

5.1.2. Dotazy

Název dalšího z hlavních objektů – dotaz – vystihuje zcela přesně jeho funkci, která je mu v databázovém prostředí přidělena. Pomocí dotazu můžeme položit tabulce otázku začínající tázacími výrazy např. „Který...“, „Kolik...“, „Kdo...“, „Kdy...“, „Co...“, „Jak

¹⁶ Toto kódování je 16bitové, tudíž jeden znak má velikost 2 bajty.

často...“. Výsledek dotazu tvoří konkrétní pohled na data v tabulce, přičemž jedné tabulce můžeme položit více dotazů. [10]

Pro vytvoření dotazů v MS Accessu můžeme využít následujících způsobů:

- Prostřednictvím *průvodce dotazem* – jedná se o jeden z povedených automatických průvodců pro práci v MS Accessu;
- Pomocí *návrhového prostředí* – jedná se o jednoduchý a intuitivní grafický prostředek typu *QBE (Query By Example)*, v jehož pozadí se automaticky generuje příkaz jazyka SQL.
- Přímým napsáním dotazu v *jazyce SQL* – jak je uvedeno v publikaci [14], až do verze MS Access 2000 bylo využíváno jazyka SQL odpovídajícího pouze normě *ANSI 89 Level 1* (tzn. Microsoft Jet SQL). S uvedením databázového stroje Microsoft Jet verze 4.X (MS Access 2002) se však tato syntaxe přiblížila více syntaxi jazyka ANSI 92 SQL, což usnadňuje převod databází MS Accessu na MS SQL Server, jehož SQL jazyk je také kompatibilní s ANSI 92¹⁷.

Dotazy v MS Accessu lze rozdělit do několika základních skupin:

- *Výběrové dotazy* – jedná se o nejběžnější typ dotazu, který načítá data z jedné nebo více tabulek a výsledek zobrazuje v datovém listu, kde lze provádět aktualizaci záznamů (s určitými omezeními). Výběrový dotaz se také používá k seskupení záznamů a výpočtu součtu, počtu, průměru a dalších typů souhrnů. [14]
- *Parametrické dotazy* – jsou dotazy, které po spuštění zobrazí dialogové okno. V tomto okně se zadávají informace, například kritéria (podmínky zadané za účelem omezení počtu záznamů, které budou zahrnuty do sady výsledků dotazu nebo filtru) pro výběr záznamů nebo hodnoty, které chcete zadat do pole. V rámci jednoho dotazu si lze vyžádat zadání několika informací zároveň, například dvojice kalendářních dat. Na základě těchto dat pak aplikace

¹⁷ Pro přesnější prostudování jednotlivých odlišností daných verzí jazyka SQL v MS Accessu a normy ANSI SQL, odkažme na zdroje [14] nebo [15], kde jsou tyto rozdíly podrobně uvedeny.

Microsoft Access vyhledá záznamy, které svým datem spadají do zadaného období. [14]

- **Křížové dotazy** – tyto dotazy usnadňují analýzu dat pomocí výpočtů a změny struktury dat. Pomocí křížových dotazů lze vypočítat součty, průměry, počty a různé souhrnné výpočty pro data seskupená na základě dvou typů informací (informací po levé straně datového listu a informací na horním okraji listu). [14]
- **Akční dotazy** – jedná se o typy dotazů, které provádí změny nebo přesuny mnoha záznamů v rámci jediné operace. [14]. Existují čtyři typy akčních dotazů:
 - **Vytvářecí dotaz** – slouží pro rychlé vytvoření tabulky společně s vložením určité množiny záznamů. (SQL příkaz: „*select into*“).
 - **Přidávací dotaz** – do již existující tabulky přidá množinu záznamů z jedné nebo více zdrojových tabulek/dotazů. (SQL příkaz: „*insert into*“).
 - **Aktualizační dotaz** – slouží pro aktualizaci hodnot v určených záznamech. Hodnoty se aktualizují na základě vypočítaných hodnot, konstant nebo získaných údajů z jiných tabulek. (SQL příkaz: „*update*“ nebo „*set*“).
 - **Odstraňovací dotaz** – slouží k odstranění jednoho, nebo více záznamů určených klauzulí SQL „*where*“ (pokud tato klauzule „*where*“ chybí, odstraňují se všechny záznamy ve vybrané tabulce. Příkaz SQL zní: „*delete*“).
- **Definiční dotazy** – nebo také **dotazy DDL (Data Definition Language)** se tvoří pomocí příkazu SQL jazyka a používají se ke správě relačních databází – tvoření, odstraňování a změně tabulek v databázi.

5.1.3. Formuláře

Formuláře slouží k zobrazení, editaci či vkládání dat do tabulek. S jeho pomocí, můžete vytvořit poutavější a někdy i přehlednější pracovní prostředí pro správu dat, než jaké

poskytuje samotná tabulka nebo dotaz. Slouží tedy často jako prvotní rozhraní mezi uživatelem a aplikací vytvořenou v MS Accessu.

Mezi základní funkce formulářů poté patří:

- **Zobrazování a úprava dat** – jedná se o nejčastější využití formulářů;
- **Řízení toku aplikace** – formuláře se mohou také navrhnout tak, aby pracovali s procedurami VBA nebo aby obsahovali příkazová tlačítka. Pomocí těchto prvků je tedy možné otevírat další formuláře, spouštět nové dotazy, provádět příkazy z voleb atd.
- **Příjem vstupů** – takto navržené formuláře slouží výhradně pro vkládání nových dat do databáze.
- **Zobrazování hlášení** – formuláře se také využívají jako nápovědy pro aplikaci v MS Accessu.

Formuláře jde samozřejmě vytvořit jak pomocí **automatického průvodce**, který obsahuje několik tématických šablon formulářů, nebo pomocí **grafického návrhového prostředí**.

Při vytváření formulářů, se také mnohdy využívá **ovladačů ActiveX**, které jsou dostupné i v ostatních aplikacích sad MS Office. Tímto způsobem lze do formulářů zahrnout prvky jako například *TextBox (textové pole)*, *ComboBox (pole se seznamem)*, *ListBox (seznam)*, *Button (tlačítko)*, *TabControl (ovladač karty)*, *RadioButton (přepínač)*, *ImageList (kontejner bitmap a ikon)*, *ProgressBar (ukazatel průběhu úlohy)*, *ToolBar (panel nástrojů)* atd.

5.1.4. Sestavy

Objektem, který je určen pro prezentaci dat v tištěné podobě, ve formě tabulek či grafů, je sestava. Na úrovni sestavy můžeme data nejenom zobrazovat, ale také s nimi provádět výpočty, sumarizovat, řadit či seskupovat. [10]

Velkou výhodou sestav je možnost jejich vytvoření pomocí **automatického průvodce**, který ve většině případů požadavků na tiskové sestavy postačí. V případě potřeby je pro tvorbu tiskových sestav v MS Accessu přítomno i speciální grafické návrhové prostředí,

ve kterém se pracuje podobně jako v návrhovém prostředí pro formuláře a pomocí kterého lze vytvářet dynamické sestavy (i za pomoci kódu VBA).

5.1.5. *Datové stránky*

Datové stránky (Data Access Page, DAP) byly jednou z největších a nejpokrokovějších novinek v MS Accessu 2000 (verze 9.0)¹⁸. Jedná se totiž o možnost zobrazit a zadávat data do databáze MS Accessu prostřednictvím internetového prohlížeče¹⁹.

Datové stránky jsou vytvářeny jako HTML dokumenty, které jsou připojeny na databázový stroj Microsoft Jet (Komunikují s databází MS Accessu prostřednictvím ODBC ovladačů databázového stroje MS Jet) v případě databáze MDB, nebo na MS SQL Server (Respektive na jeho výkonné jádro MSDE – Microsoft Data Engine) v případě projektů ADP.

Výraznými výhodami datových stránek je jejich možnost rychlého vytvoření prostřednictvím *automatického průvodce* a možnost je také oddělit od aplikace MS Access – tedy *vyexportovat* jako typickou *HTML stránku*, která je nezávislá na spuštění MS Accessu a která také samostatně komunikuje s databází prostřednictvím databázového stroje Microsoft Jet

Datové stránky lze samozřejmě vytvořit i prostřednictvím speciálního grafického prostředí nebo změnou již existujících HTML stránek. Pro tvorbu datových stránek platí, že umějí využívat podobných komponent jako formuláře nebo sestavy a dále pak prvků typických pro HTML stránky.

¹⁸ Jedná se o první náznaky snahy Microsoftu připojit MS Access k SharePoint Serveru. V následujících verzích tyto snahy byly dále rozšiřovány a dle zkušebního provozu balíku MS Office 2007 a dle vyjádření Microsoftu, bude nová verze MS Access již obsahovat přímé připojení datových stránek k SharePoint Serveru 2007. Více v kapitole 5.5.

¹⁹ Tuto možnost však Microsoft zaručuje pouze v případě programu Internet Explorer od verze 5. Konkurenční internetové prohlížeče (např. FireFox, Opera atd.) musí mít pro toto využití doinstalovány speciální Plug-iny třetích stran.

5.1.6. Makra

Při práci s daty v databázi se vyskytnou některé často opakované činnosti, které však lze do určité míry zautomatizovat za účelem zefektivnění práce v MS Accessu. Toto zautomatizování se provádí právě prostřednictvím maker.

Samotná makra však nelze využít v mnoha případech, kdy je potřeba dosáhnout složitějších procesů zde musí nastoupit samostatná implementace prostřednictvím kódu VBA²⁰ (takové případy nepostradatelnosti VBA jsou například *řešení chybových situací, deklarace a volání rutin API* nebo *provádění úloh spojených s replikacemi databází*).

Je dobré si však uvědomit, že ve verzích MS Accessu 9.0 a 10.0, již jde nahradit většina maker vlastním kódem VBA²¹. Využití makra tak zůstává k dispozici uživatelům, kteří neovládají kód VBA.

5.1.7. Moduly

Modul je kolekce deklarací, příkazů a procedur uložených společně v pojmenovaném celku. Moduly slouží k organizaci kódu VBA. V aplikaci MS Access existují dva základní typy modulů:

- **Standardní moduly** – obsahují obecné procedury, které jsou nezávislé na objektech existujících v souboru databáze MS Access. Na tyto procedury mohou, díky jejich nezávislosti, odkazovat všechny objekty v rozsahu celé databáze (aplikace MS Access).
- **Moduly tříd** – existují buď samostatně, nebo ve spojení s *formuláři* a *sestavami*. K procedurám umístěným v modulu třídy lze zpravidla přistupovat i z jiných modulů. **Modul formuláře** je modul svázaný s formulářem a obsahující přinejmenším jednu deklaraci nebo proceduru. Modul formuláře se vytvoří automaticky v okamžiku vytvoření první událostní procedury buď formuláře,

²⁰ Samotný kód VBA je v MS Accessu začleněn od verze 8 a toto začlenění bylo významným krokem k rozšíření možností uplatnění MS Accessu.

²¹ Snad kromě případu makra *AutoKeys*, kdy jde o přiřazení akce (sady akcí) klávesové kombinaci.

nebo ovládacího prvku umístěného na formuláři. **Moduly sestavy** poté fungují stejně jako moduly formulářů. Liší se jen jinou sadou událostí.

Jednoduše řečeno, moduly jsou souborem jednotlivých částí kódu VBA, vytvořeného uživateli pro práci s databází MS Access. Moduly jsou velmi silnou stránkou vývojářské části MS Accessu a bez využití kódu VBA (modulů) ztrácí MS Access na své efektivitě. Možnost pracovat s VBA kódem v MS Accessu je od verze 8.0. Ve verzi 9.0 (MS Access 2000) má poté verze VBA 6.0 v MS Accessu shodné součásti jako Visual Basic 6.0. Od verze MS Access 2002 je již použita verze VBA 6.3. Z tohoto faktu také vyplývají další pravidla při tvorbě kódu VBA. Pro podrobnější prostudování tvorby kódu VBA v aplikaci MS Access můžeme odkázat na publikaci [13], která je celá věnována tomuto tématu.

5.2. Zabezpečení dat

Databázové systémy mnohdy spravují velmi důležitá data, která jsou navíc v celém informačním systému firmy zcela jistě nepostradatelnou a velmi cennou součástí. Proto jakákoli ztráta těchto dat může znamenat opravdu závažný problém.

Z tohoto důvodu je také jedním z nejdůležitějších úkolů databázových systémů zajistit automatickou správu a zabezpečení dat tak, aby k žádné ztrátě dat (ať už ve smyslu poškození, nebo krádeže) nikdy nedošlo. Je samozřejmé, že počítače jsou pouze stroje a jako takové jsou jistě také poruchové a jistotu, že k žádné ztrátě dat nedojde, nemůžeme mít nikdy. Mnohým databázovým systémům (například MS SQL Server, DB2, Oracle) se však úspěšně daří riziko vzniku chyby držet na velmi nízkém procentu.

V této kapitole proto nyní představme některé z nejznámějších způsobů zabezpečení databáze v systému MS Access tak, jak je předkládá samotný výrobce, firma Microsoft, ve zdrojích [14], [16], [17] nebo [18].

Podle těchto zdrojů lze pro zabezpečení a řízení úrovně přístupu k databázi aplikace MS Access použít několik různých strategií:

- ***Kódování a dekodování souboru databáze***
- ***Zobrazení nebo skrytí objektů v okně databáze***
- ***Použití možností pro spuštění***

- *Použití hesla databáze*
- *Použití individuálního řízení přístupu*
- *Zabránění uživatelům v provádění replikace a nastavení databáze*
- *Zabezpečení kódu jazyka VBA*

5.2.1. Kódování a dekódování souboru databáze

Standardní databáze produktu MS Access je uložena ve speciálním souboru typu MDB. Tento soubor je však v nekódovaném formátu, a proto jej lze některými textovými editory otevřít a editovat.

Aby však tento soubor databáze MDB nešlo otevřít jiným programem než aplikací MS Access, lze databázi zakódovat. Vlastní proces zakódování databáze se provádí prostřednictvím speciální funkce v MS Accessu, která soubor databáze také zkomprimuje (databáze poté zabírá méně prostoru na disku). Je však důležité si uvědomit, že toto zakódování databáze, která nevyužívá žádná z dalších opatření zabezpečení, nemá žádný účinek, protože kdokoli může databázi otevřít a získat úplný přístup ke všem objektům a datům uloženým v databázi.

Pro provedení kódování nebo dekódování (Dekódování je opačný proces kódování.) databáze MS Access, musí být daný uživatel buď *vlastníkem databáze* (Při použití zabezpečení heslem se jedná o uživatelský účet ovládající databázi nebo databázový objekt. Ve výchozím nastavení je vlastníkem ten uživatelský účet, který databázi nebo databázový objekt vytvořil.) nebo, pokud databáze využívá opatření zabezpečení, členem *skupiny Administrators (Správci)* v informačním souboru pracovní skupiny (Tento soubor obsahuje informace o uživatelích v pracovní skupině a nastavení jejich oprávnění vzhledem k práci s databází). Dále daný uživatel musí mít oprávnění otevřít databázi ve **výhradním režimu** (Zde se jedná o typ přístupu k datům v databázi sdílené v síti. Po otevření databáze ve výhradním režimu je ostatním uživatelům zabráněno tuto databázi otevřít.).

5.2.2. Zobrazení nebo skrytí objektů v okně databáze

Dalším způsobem zabezpečení objektů v databázi před příležitostnými uživateli je možnost skrytí objektů v okně Databáze. Tento způsob ochrany však nabízí *nejnižší míru zabezpečení*, protože je relativně snadné skryté objekty znovu zobrazit.

5.2.3. Použití možností pro spuštění

Možnosti při spuštění určují některá nastavení, například zobrazení úvodního formuláře, který se automaticky otevře při otevření databáze. Tato nastavení nám tudíž umožňují okno databáze skrýt a zobrazit vlastní přepínací panel, který uživateli poskytne jen ty funkce, které byly předem nadefinovány.

Užití pouze tohoto způsobu ochrany databáze, stejně jako již zmíněného předchozího způsobu (popsaného v kapitole 5.2.2), je zcela nedostačující pro její zabezpečení a je možné ho doporučit pouze jako doplňující metodu. Změny možností pro spuštění totiž může provádět kterýkoliv uživatel (Pokud není tato funkce změny zablokována, pomocí jiných způsobů zabezpečení popsaných níže.), tudíž pro středně pokročilého uživatele MS Accessu není toto zabezpečení problémem odstranit.

5.2.4. Použití hesla databáze

Další jednoduchou metodou zabezpečení je nastavení hesla pro otevření databáze MS Access (soubor MDB). Jakmile je nastaveno heslo databáze, zobrazí se při jejím otevření dialogové okno, které požaduje od uživatele zadání hesla. Otevření databáze je poté povoleno pouze uživatelům, kteří zadají správné heslo. V takto otevřené databázi jsou již danému uživateli opět zpřístupněny všechny její objekty (Pokud již nebyly definovány jiné druhy zabezpečení, které jsou popsány níže.).

V aplikaci MS Access je však heslo uloženo v nešifrovaném tvaru proto, pokud to může ohrozit zabezpečení databáze, nedoporučuje se databázi chránit použitím hesla databáze. Použití hesla databáze má také nevýhodu při replikaci databáze, neboť v takovém případě nelze databázi synchronizovat.

5.2.5. Použití individuálního řízení přístupu

Podle zdroje [14] patří použití individuálního řízení přístupu k nejpružnější a nejrozsáhlejší metodě využívání opatření zabezpečení databáze. Pomocí této metody lze přidělit jednotlivým uživatelům různé úrovně oprávnění k přístupu a práci v databázi MS Access. V případě přidělení takovýchto práv uživatelům databáze, jsou tito uživatelé při spuštění aplikace MS Access povinni zadat heslo (V případě verze MS Access 2003 je heslo vyžadováno až při otevření databáze.). Aplikace Access pak načte informační soubor pracovní skupiny (SYSTEM.MDW), kde jsou jednotliví uživatelé identifikováni jedinečnými identifikačními čísly (takzvanými **SID** – **Security ID**, stejně jako v prostředí Windows). Úroveň přístupu a dostupné objekty jsou pak řízeny na základě identifikace daného SID a hesla (heslo je také součástí souboru SYSTEM.MDW).

Pro správné a funkční zabezpečení databáze MS Access tímto způsobem je zcela nepostradatelné pochopit funkci identifikačního čísla SID a informačního souboru SYSTEM.MDW. V aplikaci MS Access jsou totiž prostřednictvím souboru SYSTEM.MDW přístupni vždy dva uživatelé – uživatel „Admin“ a uživatel „User“. Dále soubor SYSTEM.MDW také vždy obsahuje dvě skupiny uživatelů – skupinu „Admins“ a skupinu „Users“ (Uživatel *Admin* poté patří do skupiny *Admins* a uživatel *User* do skupiny *Users*). To, co je na těchto uživatelích ve vztahu k zabezpečení databáze důležité, je fakt, že tito **uživatelé a jejich skupiny mají vždy stejná SID** (Každá instalace MS Accessu má tedy tyto uživatele naprosto identické.). Tato skutečnost, spolu s principem **vlastníka databáze** (popsaným v kapitole 5.2.1) tedy může v praxi znamenat fatální selhání tohoto způsobu zabezpečení. Neboť vytvoříme-li například databázi MS Access pod uživatelem *admin*, zapíše se tento uživatel jako **vlastník** této databáze. Jestliže poté tuto databázi přeneseme na jiný počítač, kde je nainstalována aplikace MS Access (spolu s identickým uživatelem *admin* – má stejné SID), můžeme databázi otevřít (Navzdory jakýmkoli dalším bezpečnostním nastavením

– např. hesla.) a nemáme vůči této databázi žádná omezení²² – neboť uživatel *admin* je jejím vlastníkem. V případě využití tohoto způsobu zabezpečení je tedy zcela nezbytné naprosto vyřadit tyto výchozí uživatele z používání a zakládat tedy databáze pod speciálně vytvořeným uživatelem.

Nastavení individuálního řízení přístupu k databázi se provádí poměrně jednoduše a rychle prostřednictvím „*Průvodce uživatelskou úrovní zabezpečení*“.

5.2.6. Zabránění uživatelům v provádění replikace a nastavení databáze

Při vytváření zabezpečení databáze může být také nutné zabránit uživatelům v replikaci databáze (Jedná se o proces vytvoření kopie databáze, při kterém jsou dvě nebo více kopií schopné synchronizovat.), nebo v nastavení hesla databáze (viz kapitola 5.2.3) či snad ve změně vlastností pro spuštění (viz kapitola 5.2.2).

Nemá-li však sdílená databáze definované individuální řízení přístupu, nemůže být uživatelům v těchto změnách zabráněno, neboť toto nastavení se provádí právě na úrovni definice práv skupině nebo jedinému uživateli. Je-li tedy individuální řízení přístupu definováno, poté uživatel nebo skupina uživatelů musí mít k replikaci databáze, popřípadě nastavení hesla databáze nebo ke změně vlastností databáze po spuštění přiřazené speciální oprávnění, kterému se říká *Správce*. Toto oprávnění Správce však mají pouze členové skupiny *Administrators* aktuální pracovní skupiny a platí, že přidáním nebo odebráním tohoto oprávnění, se přístup k těmto třem výše uvedeným úlohám definuje společně. [14]

5.2.7. Zabezpečení kódu jazyka VBA

Existuje několik různých způsobů, jak pomoci chránit kód jazyka Microsoft VBA ve standardních modulech a modulech tříd. Jedním z těchto způsobů je například chránit kód pomocí hesla, které bude nutné zadat na začátku každé relace. Takovéto heslo

²² Stejného výsledku můžeme také dosáhnout přímo na výchozím počítači, a to pouhým smazáním souboru SYSTEM.MDW, který se při dalším spuštění aplikace MS Access vytvoří nový, který bude obsahovat výchozí uživatele *Admin* a *User* bez jakýchkoliv bezpečnostních omezení.

pomáhá zabránit neoprávněným uživatelům v jakékoliv úpravě nebo kopírování kódu VBA.

5.2.8. Nedostatky v zabezpečení programu MS Access

Výše popsanými způsoby lze poměrně úspěšně zabezpečit data v databázi, vůči přístupu prostřednictvím aplikace MS Access. To, co je však velmi důležité pro celkové zabezpečení dat, je možnost přímého přístupu k databázi. Proto, aby mohlo s databází pracovat více uživatelů, je totiž zapotřebí, aby fyzický soubor databáze (MDB) byl umístěn na místě sdíleném spolu s ostatními uživateli databáze. Tedy aby tito uživatelé měli přímý přístup k souboru databáze (Tak jako je to například s dokumenty programu MS Word.). To ovšem znamená, že i přes veškerá výše zmiňovaná zabezpečení, není aplikace MS Access schopna zajistit, aby nedošlo k případnému přímému poškození či krádeži celého souboru (MDB) databáze. V tomto případě se tedy jakékoli důkladnější zabezpečení databáze přesouvá již na úroveň zabezpečení souborového systému, který je však pro řešení databázového systému MS Access nedostačující²³.

Dalším znepokojujícím faktorem v zabezpečení dat v databázi opět přímo souvisí s přístupem k souboru databáze. Systém MS Access je totiž konstruován takovým způsobem, že při případné práci více uživatelů najednou, má každý z těchto uživatelů přímý přístup k datům v souboru databáze. To ovšem znamená, že ve chvíli vyvolání více požadavků na databázi najednou (nejčastěji čtení, zápis, úprava dat) mají požadavky všech uživatelů stejná práva, kvůli čemu musí nutně docházet ke kolizím v přístupu – například dva uživatelé ukládají do stejného záznamu rozdílná data, což samozřejmě musí znamenat ztrátu dat, alespoň pro jednoho z nich (druhý uživatel přepíše záznam prvnímu). Tento přístupový problém je sice v aplikaci MS Access ošetřen zobrazením případného dialogového okna s upozorněním, že s danými daty nyní pracuje jiný uživatel, ale prakticky toto řešení nezabrání vyvolání tohoto přístupového konfliktu.

²³ Zde zmíníme, že je samozřejmě možné toto zabezpečení také zajistit prostřednictvím speciálních „kryptovacích“ programů a jiných produktů třetích stran. Takováto řešení však již suplují způsob zabezpečení, který je u jiných databázových systémů běžně přístupný.

5.3. Víceuživatelské rozhraní Access

K tomu, aby mohlo více uživatelů přistupovat k datům v databázi MS Access, je zapotřebí, aby fyzický soubor databáze (MDB) byl umístěn ve sdílené složce na síti, kam má každý uživatel této databáze oprávnění přistupovat. Tento předpoklad vychází z přímého přístupu klientské části aplikace MS Access k souboru databáze – aplikace MS Access vlastní soubor databáze vlastně otevírá podobným způsobem, jako jsou otevírány například dokumenty programu MS Word (v rámci zabezpečení dat v takovéto databázi však přímý přístup k souboru databáze představuje bezpečnostní nedostatek, který je blíže diskutován v kapitole 5.2).

Z přímého přístupu aplikace MS Access k souboru databáze dále vyplývá, že pro otevření databáze jedním nebo více uživateli (na více počítačích v pracovní síti) je nutné, aby každý z těchto uživatelů měl nainstalován na svém počítači systém MS Access nebo konkrétně jeho databázový stroj Microsoft Jet.

Dále z přímého přístupu k souboru databáze vyplývá, že při tomto víceuživatelském přístupu k databázi se daní uživatelé fakticky dělí v reálném čase o tento soubor a každý z nich má tudíž naprosto stejná práva pro zápis nebo čtení ze souboru (Tato práva sice lze upravit v rámci zabezpečení MS Accessu, avšak ona úprava neřeší faktický přístup do souboru databáze.). Je tedy zřejmé, že množství uživatelů, kteří mohou současně přistupovat do databáze systému MS Access, je omezen množstvím a složitostí jejich požadavků na obsluhu v této databázi.

V praxi toto znamená, že k databázi MS Accessu je umožněno přistupovat přibližně 3 až 4 uživatelům současně²⁴, neboť v případě přístupu více uživatelů mnohdy dochází ke kolizím při ukládání dat a k případnému selhání systému (některé informační prameny, například [12] se sice zmiňují o vyšším množství uživatelů, avšak žádný z nich neuvádí vyšší počet než 10). Tuto relativní neschopnost obsluhovat více uživatelů současně také potvrzuje samotné tvrzení firmy Microsoft (dle zdroje [16]), která produkt MS Access

²⁴ Počet uživatelů schopných přistupovat k datům databáze, byl zjištěn v rámci testů prováděných na databázovém systému MS Access firmy Callida, s.r.o., za předpokladů běžného zadávání informací do takovéto databáze. Více uživatelů systém MS Access nebyl schopen, bez kolizí při ukládání dat, obsloužit.

radí především do personálních programů a sama deklaruje pro poslední verzi databázového stroje MS Accessu (Microsoft Jet 4.5), maximální počet uživatelů okolo dvaceti.

Ve srovnání s jinými databázovými systémy (například MS SQL Server, nebo Oracle), ve kterých je přístupu více uživatelů současně dosaženo způsobem uchování databáze mimo přístup uživatelů a zároveň obslužením uživatelů této databáze prostřednictvím speciální služby (Tato služba řídí způsob přístupu k databázi bez účasti uživatelů, a to prostřednictvím takzvaného transakčního log souboru, do kterého jsou zapisovány veškeré příchozí požadavky a v případě jejich vyřízení i veškeré výstupní výsledky. Tento soubor, který dále umožňuje této službě v případě jakékoli kolizní situace návrat k předchozímu stavu databáze nebo k efektivnější obsluze více požadavků jednoho uživatele najednou.), je proto databázový systém MS Access relativně nevhodným pro více uživatelskou obsluhu databáze informačního systému.

5.4. Access ve spolupráci s jinými programy

Propojení MS Access s jinými aplikacemi kancelářské sady Office firmy Microsoft (kterými jsou například MS Word, MS Excel, MS Outlook, MS PowerPoint nebo MS Project) je velice těsné a je realizováno především prostřednictvím technologie nazývané *Automation*.

Tato technologie je dostupná v MS Office od verze 97 a slouží k manipulaci s různými objekty²⁵ napříč všemi aplikacemi sady MS Office. Z aplikace MS Access lze pomocí této technologie, například kliknutím na jediné tlačítko, spustit MS Excel, vytvořit v něm nový sešit a naplnit jej daty. Z těchto dat je poté možné v daném sešitu vytvořit graf a soubor MS Excelu uložit a následně odeslat prostřednictvím aplikace MS Outlook například jinému pracovníkovi firmy.

Mezi velké výhody produktu MS Access patří také schopnost spolupráce i s jinými oddělenými databázovými systémy. MS Access umožňuje prostřednictvím tzv. *propojených tabulek* pracovat s daty uloženými například v databázi dBASE, Paradox,

²⁵ Zde je zapotřebí opět nevytvářet vazbu mezi výrazem „objekt“ a současným trendem uplatňujících se objektových metod v oboru IT. Slovo „objekt“ je zde použito v obecném významu.

Btrieve, FoxPro, Microsoft SQL Server²⁶ nebo jakoukoliv jinou databází podporující standard *ODBC*²⁷.

5.5. Náhled na Microsoft Office Access 2007

Tato práce je především věnována produktu MS Access a to ve verzích 8.0, 9.0, 10.0, případně 11.0. Protože však firma Microsoft již uveřejnila termín uvolnění své poslední verze balíku MS Office 2007²⁸ (12.0), jehož součástí je také *MS Office Access 2007 (12.0)*²⁹, přiblížíme si nyní v této kapitole MS Access 2007 tak, jak o něm mluví oficiální informační zdroje firmy Microsoft [16].

Vylepšení aplikace Microsoft Office Access 2007 dle publikace [19]:

- *Nové snazší a intuitivnější uživatelské rozhraní* – Office Access 2007 poskytuje zcela nový styl práce, jehož základem je přepracované uživatelské rozhraní orientované na výsledky, nové navigační podokno a zobrazení oken se záložkami.
- *Nové propojení s aplikací MS Office InfoPath a MS Office Outlook* – Office Access 2007 může vytvářet formuláře s použitím aplikace MS Office InfoPath 2007 nebo HTML pro sběr informací pro databázi. Takový formulář může být následně odeslán prostřednictvím aplikace MS Office Outlook 2007 například zákazníkům a jejich případné odpovědi použity pro naplnění tabulky databáze MS Office Access 2007.

²⁶ MS Access má s produktem MS SQL Server velmi úzkou vazbu, neboť vývojové tendence firmy Microsoft směřují k přímé spolupráci s tímto produktem na úrovni klientské části serveru MS SQL Server. Již nyní lze v MS Accessu vytvářet speciální projekty databáze, která je umístěna právě na tomto databázovém serveru.

²⁷ ODBC (Open Database Connectivity) je standard firmy Microsoft, který má představovat jednotné SQL rozhraní k různým databázovým systémům.[9]

²⁸ Tento termín se odhaduje na začátek roku 2007.

²⁹ V nové verzi kancelářského balíku Microsoft Office 2007 jsou všechny jeho součásti nově pojmenovány, a proto se také MS Access nově oficiálně nazývá *Microsoft Office Access 2007*.

- ***Zlepšené návrhové prostředí pro vytváření tiskových sestav*** – rozhraní WYSIWYG obsahuje nové podokno seskupování a filtrování a vylepšené možnosti třídění zobrazených informací.
- ***Nové přímé propojení s MS Office SharePoint Server 2007*** – aplikace MS Office Access 2007 a MS Office SharePoint Server 2007 jsou spolu nově více propojeny a lépe spolupracují. Proto lze MS Office Access 2007 například využít jako klientské rozhraní serveru Office SharePoint Server 2007 (využít tak snadné a intuitivní klientské prostředí Office Accessu 2007), nebo umístit standardní databázi MS Office Access 2007 přímo na serveru MS Office SharePoint Server 2007 a získat tak lepší možnosti správy a přístupu k datům.
- ***Nové typy polí*** – uživatelé Office Access 2007 mohou nyní připojit v podstatě libovolný dokument, obrázek či tabulku programu MS Excel k libovolnému záznamu v aplikaci (prostřednictvím nového rozšířeného pole pro vkládání souborů do databáze). *Vícehodnotové pole* také umožňuje zadání více hodnot do každé buňky (je tedy možné například přiřadit úkol více než jednomu člověku prostřednictvím jediného pole).
- ***Nové přednastavené návrhy řešení a šablony vzhledů*** – Office Access 2007 obsahuje také nové přednastavené řešení pro vytváření formulářů, sestav, nebo celých řešení tabulek, upravitelných podle specifických potřeb uživatele. Samozřejmostí jsou také nové šablony vzhledů formulářů a tiskových sestav.

6. Závěr

V průběhu zkoumání možnosti uplatnění MS Access v informačním zabezpečení firemních subjektů, bylo zjištěno několik rozhodujících faktorů, které si nyní zrekapitulujeme:

- Uživatelské rozhraní produktu MS Access je vytvořeno tak, aby umožňovalo velmi jednoduché a intuitivní ovládání programu a to i pro uživatele neovládající znalost tvorby databází (viz kapitola 5.1).
- V souboru základních vývojových součástí produktu MS Access jsou obsaženy velmi účinné prvky, které se při procesu tvorby databází vzájemně doplňují a vytvářejí tak jednu z nejsilnějších stránek produktu MS Access (viz kapitola 5.1).
- MS Access je díky technologii *Automation* a standardu *ODBC ovladačů* schopen velmi obstojně spolupracovat jak s ostatními produkty sady MS Office tak i s jinými běžně používanými databázovými systémy (viz kapitola 5.4)
- Produkt MS Access je finančně velmi snadno dostupný díky možnosti jeho zakoupení prostřednictvím kancelářského balíku programů MS Office.
- V souvislosti s některými omezeními databáze MS Access, popsanými v kapitole 5.1.1, je tento program neschopen obsloužit rozsáhlejší bázi dat (Mnohdy vyžadovanou kvalitními informačními systémy firem.).
- Databáze produktu MS Access mají velmi nízkou úroveň zabezpečení proti neoprávněnému přístupu uživatelů (viz kapitola 5.2).
- V databázi produktu MS Access je díky přímému přístupu uživatelů k souboru databáze prakticky umožněno pracovat pouze nízkému počtu uživatelů (3 – 4 uživatelé). Přičemž tento víceuživatelský přístup do databáze je častou příčinou kolizních stavů (viz. kapitoly 5.2.8 a 5.3).
- Řešení, případných havarijních situací databázového systému MS Access, je díky přímému přístupu uživatelů do databáze zcela nedostačující. Ve srovnání s jinými relačními databázovými systémy typu Oracle, DB2 nebo MS SQL

Server, totiž MS Access neobsahuje žádné potokolační soubory, které v případě potřeby slouží k zamezení ztráty dat (viz kapitola 5.3).

V závislosti na požadavku informačního systému na bez-kolizní a spolehlivý provoz databázového systému s velkým důrazem na bezpečnost dat uložených v jeho databázi, nemůžeme po zhodnocení výše uvedených rozhodujících faktorů nyní doporučit produkt MS Access pro obecné využití v informačním zabezpečení firemních subjektů. Toto nedoporučení však neplatí na využití produktu MS Access v informačním zabezpečení všude tam, kde není vyžadována potřeba víceuživatelského přístupu k databázi, vysokého zabezpečení dat proti neoprávněnému přístupu do databáze, nebo práce s velkou bází dat. V takovém případě (například u velmi malých firem, nebo v personálním provozu databázového systému), se jistě četné kladné stránky produktu MS Access stanou výraznou výhodou zásadním důvodem pro jeho využití.

7. Seznam použité literatury

- [1] CASSELL P., EDDY C., PRICE J.: *Nauč se sám Microsoft Access 2002 za 21 dní*. Praha: SoftPress, 2003
- [2] HNOJIL, J.: *Operační a databázové systémy*. Praha: ČVUT, 1999
- [3] MELICHAR, B.: *Textové informační systémy*. Praha: ČVUT, 1996
- [4] VANÍČEK, T.: *Informační systémy 2*. Praha: ČVUT, 2001
- [5] POKORNÝ, J.: *Databázové systémy a jejich použití v informačních systémech*. Praha: Academia, 1992
- [6] POKORNÝ, J.: *Počítačová databáze*. Praha: 1991
- [7] RIORDAN, R. M.: *Vytváříme relační databázové aplikace*. Praha: Computer Press, 2000
- [8] VOSTROVSKÝ, V.: *Relační databázové systémy*. Praha: ČZU PEF, 2001
- [9] MERUNKA, V.: *Objektový přístup v databázových systémech*. Praha: ČZU PEF, 2002
- [10] VOGLOVÁ, B.: *Access v kanceláři – typické činnosti krok za krokem*. Praha: Grada Publishing a.s., 2002
- [11] PÍSEK, S.: *Access 2002 snadno a rychle*. Praha: Grada Publishing a.s., 1999
- [12] PÍSEK, S.: *Access 2003*. Praha: Grada Publishing a.s., 2004
- [13] DOBSON, R.: *Programování v Microsoft Access 2000*. Praha: Computer Press, 2000
- [14] Náповěda dostupná v aplikaci MS Access 2003
- [15] ŠIMŮNEK, M.: *SQL – kompletní kapesní průvodce*. Grada, 1999, ISBN 80-7169-629-7.
- [16] MICROSOFT TEAM: *Microsoft Corporation*. Poslední aktualizace 30.6.2006, URL <<http://www.microsoft.com>>
- [17] MICROSOFT TEAM: *Pracujeme s programem Microsoft Access 97 – praktická příručka pro vaši práci*. Microsoft, 2000

- [18] MICROSOFT TEAM: *Tvorba aplikací pomocí programu Microsoft Access 97 – příručka pro vytváření, správu a distribuci databázových řešení*. Microsoft, 1997
- [19] MICROSOFT OFFICE REVIEWS TEAM: *Microsoft® Office systém verze 2007 – příručka recenzenta – květen 2006*, Microsoft, 2006
- [20] VALÁŠEK, M. A.: *Představení uživatelských instancí SQL Serveru 2005*. poslední aktualizace 12.4.2006, <URL:<http://www.aspnet.cz/Articles/86-predstaveni-uzivatelskych-instanci-sql-serveru-2005.aspx>>
- [21] LAZECKÝ, P.: *Jak na to - Zabezpečení MS Access databáze*. poslední aktualizace 21.6.2005, <URL:<http://blog.vyvojar.cz/lazo/archive/2005/06/21/6193.aspx>>