

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Evidence pracovní docházky pomocí MS Access

Autor práce: Jan Syřínek, DiS.

© 2013 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra informačního inženýrství

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Syřínek Jan

Informatika

Název práce

Evidence pracovní docházky pomocí MS ACCES

Anglický název

Register of attendance with using MS Access

Cíle práce

Bakalářská práce je tematicky zaměřena na uplatnění MS ACCESS pro účely evidence pracovní docházky a její optimalizace. Cílem práce je:

- navrhnout řešení této problematiky pomocí MS ACCESS se zřetelem na identifikované záležitosti,
- navržené záležitosti ověřit a demonstrovat na konkrétním řešení,
- ověřené záležitosti zobecnit pro další možná použití.

Metodika

Použitá metodika řešení bakalářské práce bude založena na studiu a analýze dostupných informačních zdrojů. Dále budou použity základní metody a postupy relačně databázové technologie. Navrhované řešení bude realizováno formou praktického řešení, které bude respektovat identifikované požadavky na tato řešení a ověřit navrhované záležitosti. Na podkladě syntézy teoretických poznatků a výsledků budou formulovány závěry této bakalářské práce a následně zobecněny pro další možná použití.

Harmonogram zpracování

Vymezení teoretických principů a náležitostí řešení problematiky - předmět 1. zápočtu z BP za 2. ročník: 06/2011 - 09/2011

Identifikování potřeb a náležitostí pracovní docházky a možnost její optimalizace: 09/2011 - 11/2011

Navržením způsobu řešení identifikovaných potřeb a využití potenciálních možností: předmět 1. zápočtu z BP za 3. ročník: 11/2011 - 01/2012

Ověření navrženého způsobu řešení a jeho zobecnění - předmět 2. zápočtu z BP za 3. ročník: 01/2012 - 03/2012

Rozsah textové části

30 - 40 stran

Klíčová slova

Pracovní docházka, optimalizace využívání pracovní doby, relačně databázová technologie, MS ACCESS

Doporučené zdroje informací

JERKE, N.: Microsoft Office Access 2003 pro pokročilé. CP Books. Brno 2005. ISBN 80-251-0723-X
McFEDRIES., P.: VBA for Microsoft Office 2000. UNIS Publishing. Brno 2008. ISBN 80-86097-43-9
SHELDON, R.: SQL začínáme programovat. Grada Publishing a.s., Praha 2005. ISBN 80-247-0999-6
SOLOMON, Ch.: Tvorba aplikací v Microsoft Office 97. Computer Press. Praha 1998. ISBN 80-7226-025-1
VIESCAS, J.: Mistrovství v Microsoft Access 2000. Computer Press. Praha 2000. ISBN 80-7226-274-2

Vedoucí práce

Vostrovský Václav, doc. Ing., Ph.D.

Termín odevzdání

březen 2013



Ing. Martin Pelikán, Ph.D.

Vedoucí katedry



prof. Ing. Jan Hron, DrSc., dr.h.c.

Děkan fakulty

V Praze dne 5.10.2012

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci " Evidence pracovní docházky pomocí MS Access " jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil(a) autorská práva třetích osob.

V Praze dne 15.3.2013

Poděkování

Tímto bych chtěl poděkovat za rady a připomínky při vypracování této bakalářské práce panu doc. Ing. Václavu Vostrovskému Ph.D. Za pomoc patří velké poděkování také Ing. Adéle Neumannové, kamarádce Kataríně Janské, rodině a spolupracovníkům za veškerou podporu při zpracování této práce.

Evidence pracovní docházky pomocí MS Access

Register of attendance with using MS Access

Souhrn

Relačně databázový systém MS Access (dále jen Access) je produktem společnosti Microsoft. Access je pouze částí produktové sady společnosti Microsoft. Pomocí tohoto nástroje bych chtěl názorně demonstrovat využití této produktové sady k vytvoření systému evidence pracovní docházky zaměstnanců společnosti.

Summary

Relational database system MS Access (hereinafter referred to as Access) is a Microsoft product. Access is only part of the product suite from Microsoft. With this tool I would like to clearly demonstrate the use of this product suite to create a working system of records of employee attendance.

Klíčová slova: Pracovní docházka, optimalizace využívání pracovní doby, relačně databázové technologie, MS Access, evidence docházky, SQL

Keywords: work attendance, optimizing the use of working time, relational database technology, MS Access, attendance records, SQL

OBSAH:

1	Úvod	10
2	Cíl práce a metodika	11
2.1	Cíl práce	11
2.2	Metodika	12
3	Pojem databáze	13
3.1	Relační databáze	13
3.1.1	Definice a uložení dat	15
3.1.1.1	DDE	15
3.1.1.2	ActiveX	15
3.1.2	Manipulace s daty	16
3.1.3	Řízení dat	16
3.2	Normalizace databáze	17
3.2.1	Stupně normalizace databáze	17
3.3	Klíče	18
3.3.1	Primární klíč	18
3.3.2	Cizí klíč	18
3.4	Relace	18
3.4.1	Typ relace 1:1	18
3.4.2	Typ relace 1:N	18
3.4.3	Typ relace M:N	19
3.5	Datové typy v aplikaci Access	19
3.5.1	Datový typ „text“ (char)	19
3.5.2	Datový typ „memo“	19
3.5.3	Datový typ „číslo“ (number)	19
3.5.3.1	Bajt (byte)	20
3.5.3.2	Celé číslo (integer)	20
3.5.3.3	Dlouhé celé číslo (long integer)	20
3.5.3.4	Desetinné číslo (single, double nebo real, float)	20
3.5.4	Datový typ „datum a čas“	20
3.5.5	Datový typ „měna“	20
3.5.6	Datový typ „automatické číslo“	20
3.5.7	Datový typ „ano/ne“	21
3.5.8	Datový typ „objekt OLE“	21
3.5.9	Datový typ „hypertextový odkaz“	21
3.6	Architektura Microsoft Access	21

3.6.1	Objekt „Tabulka“	21
3.6.2	Objekt „Dotaz“	22
3.6.3	Objekt „Formulář“	22
3.6.4	Objekt „Sestava“	22
3.6.5	Objekt „Datové stránky“	22
3.6.6	Objekt „Makro“	22
3.6.7	Objekt „Modul“	23
3.7	Zásady databázových návrhů	23
3.7.1	Identifikace úloh	23
3.7.2	Načrtnutí toku úloh	24
3.7.3	Identifikování datových prvků	24
3.7.4	Uspořádání dat	24
3.7.5	Navržení prototypu a uživatelského rozhraní	24
3.7.6	Vytvoření aplikace	24
3.7.7	Testování, přepracování a zdokonalování	25
3.8	SQL	25
3.8.1	Úvod do SQL	25
3.8.2	Vývoj SQL	26
3.8.3	Druhy příkazů jazyka SQL	26
3.8.3.1	Jazyk definice dat (DDL – Data Definition Language)	27
3.8.3.2	Jazyk pro řízení dat (DCL – Data Control Language)	27
3.8.3.3	Jazyk pro manipulaci s daty (DML – Data Manipulation Language)	28
4	Problematika evidence pracovní docházky	29
4.1	Identifikování současných potřeb	29
4.2	Stavy přerušení pracovní doby	30
4.3	Současná evidence pracovní docházky	31
4.4	Tvorba nové evidence pracovní docházky	31
4.4.1	Zaznamenávání informací do tabulek	31
4.4.2	Datová normalizace	32
4.4.2.1	Tabulka „Zaměstnanci“	33
4.4.2.2	Tabulka „Pobočka“	33
4.4.2.3	Tabulka „Svátky“	34
4.4.2.4	Tabulka „Oddělení“	35
4.4.2.5	Tabulka „Přerušení pracovní doby“	36
4.4.2.6	Tabulka „Rok“	37
4.4.2.7	Tabulka „Pracovní fond“	37

4.4.2.8	Tabulka „Stát“	38
4.4.2.9	Tabulka „Docházka“	39
5	Zobecněná funkce nové evidence pracovní docházky	40
6	Závěr.....	43
7	Seznam použitých zdrojů	45
7.1	Tištěné dokumenty	45
7.2	Elektronické zdroje	45
8	Seznam obrázků.....	46
9	Přílohy	47
9.1	Příloha 1. - Zákoník práce „Pracovní doba a doba odpočinku“	47
9.2	Příloha 2. – Přehled tabulek	49
9.3	Příloha 3. – Relationships (Relace).....	50

1 Úvod

V současné době, kdy se s výpočetní technikou setkáváme dnes a denně, jsou pojmy jako databáze a databázové systémy obecně skloňovány a vnímány ve smyslu evidence a uchovávání dat, se kterými dále pracujeme nebo potřebujeme pouze příslušná data do nějakého místa určení pouze zaznamenat či archivovat a pak je využívat i jiným způsobem, např. tato data nebo rovněž také soubor užitečných informací různého typu využívat pro další účely.

Postupem času je ve všech případech v oblasti informačních technologií skoro pravidlem, že se v určitém období stává příslušný informační systém pracující nad nějakou databází již poněkud zastaralým nebo spíše již nevyhovuje takovým potřebám v aktuálním čase, které by si uživatel či správce od dané aplikace představoval. Může i nastat případ či situace, že pokusy o vylepšení stávajícího systému nějakou nadstavbou, případně vylepšením, dopadnou leckdy nešťastně. Ve srovnání s tím je výhodnější se raději pustit do vývoje zcela nové aplikace, která by zprvu byla podrobena srovnávací analýze se stávajícím systémem a byly by vyčleněny některé nadbytečné funkcionality nebo naopak by byla rozšířena nová aplikace, ve které by, oproti starší verzi, potřebnější funkce rozšířily datovou základnu. Základem je vždy vytvořit co nejpodrobnější analýzu, rozebrat potřeby uživatelů a správců do nejmenších detailů, abychom se vyhnuli v budoucnu komplikacím, menšího či většího rozsahu.

Jako krásný příklad, který výše popsanou situaci řeší, může být problematika, která je předmětem této bakalářské práce. Tato práce se zaměřuje na evidenci pracovní docházky zaměstnanců dané společnosti a řešení návrhu praktického využití aplikace pomocí nástroje MS Access a zobecnění pro další možná využití.

2 Cíl práce a metodika

2.1 Cíl práce

Záměrem mé bakalářské práce je uplatnění nástroje Access od firmy Microsoft za účelem evidence pracovní docházky.

V počátečních kapitolách se tato práce zabývá vymezením teoretických principů relačně databázového prostředí MS Access.

Další kapitoly se již zaměřují na nalezení a identifikování potřeb pro evidenci pracovní docházky zaměstnanců společnosti. Řešení popsané problematiky bude vyplývat na základě nevyhovujícího stávajícího zaznamenávání evidence docházky zaměstnanců.

Na základě analýz a potřebných konzultací bude zajištěn postup při tvorbě návrhu řešení, kde budou zohledněny všechny potřebné aspekty, aby bylo vyhověno daný požadavkům.

Dle požadavků, které budou sumarizovány, bude nutné zaznamenávat další stavy přerušení pracovní doby zaměstnance, které ve stávajícím řešení bylo nedostačující a nevyhovovalo současným potřebám. Samozřejmostí bude i tvorba potřebných reportů.

Řešení zanalyzovaného tématu bude zpracováno tedy za pomocí nástroje MS Access, kde budou vytvořeny patřičné tabulky, dotazy, sestavy, vyexportované soubory pro další práci.

Navržené zpracované požadavky budou ověřeny a demonstrovány na konkrétním řešení.

Výsledky tohoto řešení budou zobecněny pro možná další využití, jako například možnosti exportu výsledků docházky do jiných systému či další práce s těmito hodnotami.

2.2 Metodika

Metodika, která byla využita v této bakalářské práci, je založena na základě studia a podrobné analýze dostupných informačních zdrojů.

Zároveň budou uplatněny základní metody, principy a postupy relačně databázové technologie, která zde bude reprezentována pomocí nástroje společnosti Microsoft a to aplikace Access.

Řešení, které zde bude navrženo, bude realizováno formou praktického řešení. Toto řešení bude názorně reflektovat a respektovat identifikované požadavky v této problematice.

Navrhované záležitosti budou poté ověřeny v daném řešení.

Na důsledném podkladě syntézy teoretických poznatků a pomocí výsledků budou formulovány závěry této bakalářské práce a následně zobecněny k dalšímu možnému využití.

3 Pojem databáze

V nejjednodušším smyslu je databáze množina záznamů a souborů, které jsou organizovány za určitým účelem. Ve svém počítači si můžete uchovávat jména a adresy všech svých přátel nebo zákazníků. Můžete soustřeďovat všechny dopisy, které jste napsali, a třídit je podle příjemců. Dokumenty z textového procesu, které třídíte podle tématu, jsou jedním typem databáze. Soubory tabulkového procesoru, které třídíme podle jejich využití, jsou dalším typem databáze (Viescas, 2000).

Pokud jste velmi pečlivý, můžete případně pomocí složek a podsložek spravovat několik stovek tabulek nebo zástupců. Jestliže tak činíte, jste správcem databáze. Co však uděláte, pokud problémy, které se snažíte řešit, příliš narostou? Jak můžete snadno shromažďovat informace o všech zákaznících a jejich objednávkách, když tato data mohou být uložena v několika dokumentech a souborech tabulkových procesorů? Jak můžete udržovat vazby mezi soubory, pokud ukládáte nové informace? Jak zajistíte, aby data byla vložena správně? Co dělat v případě, když potřebujete svoje informace sdílet s mnoha lidmi, ale přitom nechcete, aby se pokusili aktualizovat stejné data dva lidé současně? Pokud jste postaveni před tyto úkoly, pak potřebujete systém pro správu databází (database management systém – DBMS) (Viescas, 2000).

3.1 Relační databáze

Téměř všechny moderní systémy pro správu databází ukládají informace a pracují s nimi za použití relačního modelu řízení databáze.

Termín relační pramení ze skutečnosti, že každý záznam v databázi obsahuje informace vztažené (v relaci) k jedinému subjektu a pouze k tomuto subjektu. Také data o dvou třídách informace (například zákazníci a objednávky) mohou být zpracovávána jako jediná entita založená na vztažených hodnotách. Například by bylo nadbytečné ukládat informace o jméně zákazníka a jeho adrese s každou objednávkou, kterou tento zákazník uskuteční. V relačním databázovém systému obsahuje každá informace o objednávkách pole pro uložení určité hodnoty, například čísla zákazníka, která může být použita pro napojení každé objednávky na informace o zákazníkovi. V systému správy relační databáze, někdy označovaném jako RDBMS, pracuje systém se všemi daty v tabulkách. Tabulky uchovávají informace o subjektu (například adresy zákazníků nebo tituly knih) a

mají sloupce, které obsahují různé druhy informací o tomto subjektu (například data o konkrétním zákazníkovi nebo knize). I tehdy když databázi položíte nějaký dotaz (vyvolání informací z jedné nebo více tabulek), bude výsledek vždy vypadat jako další tabulka (Viescas, 2000).

Můžeme například spojovat data vztahující se k informacím o autorech s informacemi o knihách a tak získáme výpis, který autor napsal, kterou knihu, atd.

Do terminologie relačních databází lze zahrnout tyto pojmy:

Relace (relation) – vazba mezi více množinami dat.

Atribut (attribute) – jedná se o konkrétní informaci o subjektu. Atribut je ve vztahu k tabulce uložen jako datový sloupec nebo pole.

Vztah (relationship) – určuje způsob, jakým jsou informace z jedné relace ve vztahu k informacím z druhé relace. Například jedna ku jedné, jedna ku n či m ku n.

Spojení (join) – jedná se o provázání tabulek či dotazů na tabulky prostřednictvím jejich vztažených datových hodnot.

RDBMS umožňuje plnou kontrolu na tím, jak definovat data, pracovat s nimi a sdílet je s ostatními. Takový systém také zabezpečuje důmyslné funkce, které usnadňují uspořádání a správu rozsáhlých objemů dat v mnoha tabulkách (Viescas, 2000).

RDBMS má tři hlavní typy funkcí: definice dat, manipulace s daty a řízení dat.

Definice dat (data definition) – definujeme, jaká budou data v databázi uložena a popřípadě můžeme zároveň říci jak budou i data formátována

Manipulace s daty (data manipulation) – jednoduše řečeno, jak můžeme s daty pracovat, tj. např. filtrování, řazení, výběr. Data můžeme spojovat, aktualizovat, odstraňovat, kopírovat do jiných tabulek nebo vytvářet tabulky, které budou obsahovat kopírovaná data.

Řízení dat (data control) – zde lze definovat oprávnění uživatele, tj., kdo může data číst, aktualizovat či vkládat.

3.1.1 Definice a uložení dat

RDBMS umožňuje definování druhu ukládaných dat. V RDBMS lze také definovat pravidla pro zajištění integrity ukládaných dat. Pomocí validačního pravidla pak můžeme zajistit, aby nedocházelo k ukládání textových znaků do polí, kde mají být uloženy číselné hodnoty. Dále můžeme také nastavovat různé platné hodnoty pro dané pole či rozsah těchto ukládaných hodnot. Data lze také exportovat nebo importovat z jednotlivých tabulkových nebo textových editorů. Zároveň je zde i možnost přístupu do souborů ostatních databázových prostředků, jako jsou FoxPro, dBASE IV či Paradox. Access zároveň umožňuje i komunikaci s ostatními databázemi, které podporují standard ODBC (Open Database Connectivity) jako je MS SQL Server nebo Oracle. V aplikaci Access lze též využít technologií DDE a ActiveX.

3.1.1.1 DDE

Technologie DDE (dynamická výměna dat) umožňuje vykonávat funkce a posílat datat mezi Accessem a libovolnou další aplikací Windows, která DDE podporuje. Možné také vytvořit DDE spojení do dalších aplikací za pomoci maker nebo jazyka Microsoft Visual Basic for Applications (VBA) (Viescas, 2000).

Jedna z aplikací účastnící se rozhovoru je označována jako server a druhá jako klient. Server vlastně informace poskytuje; klient je aplikace, která proces řídí. Každá aplikace může působit jako server pro více klientů nebo jako klient pro více serverů a také současně jako klient i server (<http://www.sfiles.host.sk/builder/3/CBUILD4.html>).

3.1.1.2 ActiveX

Tato technologie umožňuje do databáze Access připojovat či vkládat objekty. K těmto objektům může patřit například tabulky, grafy, dokumenty nebo obrázky z ostatních aplikací Windows, které ActiveX podporují.

ActiveX jsou založeny na myšlence, že každá nová aplikace nemusí být programována od začátku „na zelené louce“. Programátoři si uvědomili, že mnoho aplikací v sobě obsahuje stejné funkce, které by mohly být mezi aplikacemi sdílené. Například stejná kontrola pravopisu se provádí ve Wordu i Outlook Expressu, proto namísto psaní dvou

samostatných verzí kódu stačí napsat jednu verzi sdílenou oběma aplikacemi. Pomocí ActiveX tedy lze vytvářet stavební bloky, z nichž se pak sestaví větší aplikace (<http://cs.wikipedia.org/wiki/ActiveX>).

3.1.2 Manipulace s daty

Manipulace s daty v tabulkových či textových procesorech je na rozdíl od RDBMS těžkopádná, ikdyž dovoluje vkládání tabulek, grafů či obrázků. Například data uložená v tabulce tabulkového procesoru slouží ke konkrétnímu jednomu účelu a je velmi těžké použít tato data k řešení různých problémů. Řešení je samozřejmě pomocí složitých provazeb. Avšak nástroj Access využívá pro zpracování dat v tabulkách databázový jazyk SQL (Structured Query Language). Pomocí SQL lze definovat skupinu informací, které jsou potřebné k řešení konkrétního případu, včetně dat z možných mnoha dalších tabulek. Access má pro tyto účely vyvinuté jednoduché a účinné prostředky pro grafické definování dotazů, takže je možné během krátké doby pomocí několika kliknutími myši vytvořit i složitý dotaz bez znalosti SQL.

3.1.3 Řízení dat

RDBMS umožňuje práci nad databází jak na jediné stanici, tak i formou sdílení pomocí režimu klient-server v síti. Může sloužit jako například databázový server pro intranet podniku. Access má funkce, které zajistí bezpečnost a integritu dat. Lze definovat uživatele či skupiny uživatelů, kteří budou mít přístup jen k daným objektům. Zároveň také umožňuje pomocí replikace vytvořit kopie vzorové databáze, přičemž každý uživatel může pracovat se svojí replikovanou kopií a přitom budou data synchronizována.

3.2 Normalizace databáze

Pojem normalizace je spjat s relačním modelem. V relačním modelu jsou data uložena v tabulkách, na které má jisté požadavky. Při splnění požadavků je tabulka označována jako normalizovaná. Pokud nejsou tyto požadavky splněny, jsou označovány jako nenormalizované a proces jejich převodu na tabulky se označuje jako normalizace. Při tomto procesu dochází k odstraňování nedostatků tabulek jako je redundance nebo možnost vzniku aktualizací anomálie, tj. nechtěného vedlejšího efektu operace nad databází, při kterém dojde ke ztrátě nebo nekonzistenci dat. Postup normalizace je rozdělen do několika kroků a po dokončení každého z nich se tabulka nachází v určité normální formě. V praxi se většinou normalizuje do Třetí normální formy, vyšší normální formy je vcelku obtížné porušit a vyžadují relativně velké znalosti, stejně jako návrh databází takové velikosti, kde je možné je porušit.

3.2.1 Stupně normalizace databáze

- 1NF: Každý atribut obsahuje pouze atomické hodnoty.
- 2NF: Každý neklíčový atribut je plně závislý na primárním klíči.
- 3NF: Všechny neklíčové atributy musí být vzájemně nezávislé.
- BCNF: Atributy, které jsou součástí primárního klíče, musí být vzájemně nezávislé.
- 4NF: Relace popisuje pouze příčinnou souvislost mezi klíčem a atributy.
- 5NF: Relaci již není možno bezztrátově rozložit.

Pro splnění určité normální formy je nutné splnit i ty předchozí. V praxi se obvykle používají pouze první tři. (http://cs.wikipedia.org/wiki/Normalizace_datab%C3%A1ze, 2012)

3.3 Klíče

3.3.1 Primární klíč

Při vytváření databází se můžeme setkat se dvěma typy klíčů. Primární klíč označuje hodnotu, která je pro danou množinu dat unikátní. Dobrým příkladem primárního klíče je například rodné číslo. Neexistují (resp. neměly by existovat) dvě osoby, které by měly stejné rodné číslo; jedná se tedy o unikátní identifikátor každého člověka (Jerke, 2005).

3.3.2 Cizí klíč

Druhým typem klíče je cizí klíč. Cizí klíč je jednoduše řečeno odkaz na primární klíč jiné množiny dat. Uložení hodnoty tohoto primárního klíče zajistíme propojení obou těchto množin (Jerke, 2005).

3.4 Relace

Základní podmínkou pro správný návrh databází je pochopení vazeb mezi více množinami dat (Jerke, 2005).

Při návrhu databází se lze setkat s několika typy relací. Níže si uvedeme jaké to jsou.

3.4.1 Typ relace 1:1

Jedna ku jedné (1:1) – relace 1:1, jeden konkrétní záznam v první množině dat odpovídá právě jednomu záznamu ve druhé množině dat (Jerke, 2005).

Příklad: při složené zkoušce obdrží student právě jednu známku

3.4.2 Typ relace 1:N

Relace 1:N, jeden konkrétní záznam v první množině dat odpovídá jednomu či více záznamům v druhé množině dat (Jerke, 2005).

Příklad: jeden zákazník má právě více objednávek

3.4.3 Typ relace M:N

Relace M:N, více záznamů jedné množiny dat se váže na více záznamů jiné množiny dat (Jerke, 2005).

Příklad: různé automobily mohou mít různé barvy

3.5 Datové typy v aplikaci Access

Při vytváření návrhu databáze je důležité definovat datová typy, se kterými budeme pracovat. Čísla jako 42, 2.6, nebo -3.14 jsou v databázi ukládány jinak než text, jako například „ABC“, „Kobyla má malý bok“ a další. Pokud je pole definováno jako číslo, lze tato data využívat k matematickým operacím. Access nabízí celkem devět různých datových typů (Jerke, 2005).

Kromě uvedených datových typů obsahuje Access také funkci nazvanou vyhledávání datových typů. Tato vyhledávání slouží k výběru hodnoty z jiné tabulky či ze seznamu hodnot vypsaných v rozbalovací nabídce uživatelského rozhraní. Tato vlastnost je podporována pouze v prostředí aplikace Access, proto by neměla být používána při vývoji aplikací pro jiné prostředí (Jerke, 2005). Dále by se nemělo této funkci využívat při přechodu na databázi SQL.

3.5.1 Datový typ „text“ (char)

V datovém typu text lze uchovat textový záznam o délce 255 znaků.

3.5.2 Datový typ „memo“

Datový typ memo má možnost uchování až 65 535 znaků.

3.5.3 Datový typ „číslo“ (number)

Datový typ číslo se využívá k uložení číselných hodnot nabývajících různých hodnot.

3.5.3.1 Bajt (byte)

Datový typ bajt uchovává číselnou hodnotu o velikosti 0 až 255.

3.5.3.2 Celé číslo (integer)

Datový typ celé číslo ukládá číselné hodnoty od -32768 až 32767.

3.5.3.3 Dlouhé celé číslo (long integer)

Tento datový typ ukládá numerické hodnoty v rozmezí od -2147843648 až 2147483647.

3.5.3.4 Desetinné číslo (single, double nebo real, float)

Jedná se o datový typ s plovoucí datovou čárkou. Např. 3.14 nebo 2.5 (dle anglosaské konvence psané s desetinnou tečkou)

3.5.4 Datový typ „datum a čas“

V tomto datovém typu se uchovává hodnota datumu a času.

3.5.5 Datový typ „měna“

Tento datový typ se využívá k uložení hodnot měnových údajů (např. Kč, atd.).

3.5.6 Datový typ „automatické číslo“

Je využíván většinou jako primární klíč v tabulce. Při každém novém záznamu v tabulce je hodnota automatického čísla navýšena o předepsanou hodnotu.

Příklad: položka brambory má automatické číslo 1 a přidáme další položku mrkev a ta již bude mít hodnotu automatického čísla 2, tj. předepsaná hodnota navýšení je o velikost hodnoty 1.

3.5.7 Datový typ „ano/ne“

Hodnota uložená v tomto datovém typu nabývá pouze hodnot „ano“ či „ne“ nebo také „true“ nebo „false“.

3.5.8 Datový typ „objekt OLE“

V tomto datovém typu lze uchovat objekty, jako např. dokumenty aplikace Word či tabulky aplikace Excel nebo obrázky

3.5.9 Datový typ „hypertextový odkaz“

Jedná se o datový typ uchovávající odkaz na URL adresy.

Při správném návrhu databáze je nutný správný výběr datového typu, který by měl odpovídat povaze ukládaných dat. Tedy je zcela zbytečné ukládání čísla do datového typu „text“ či dvoumístného znakového kódu země jako memo.

3.6 Architektura Microsoft Access

V MS Access vše co má jméno je objektem. Mezi objekty patří tabulky, dotazy, formuláře, sestavy, moduly a makra.

3.6.1 Objekt „Tabulka“

Jedná se o objekt s využitím pro ukládání dat. Tabulka obsahuje informace o předmětu, např. o zaměstnancích. V tabulkách jsou pole, do kterých se ukládají různé typy dat, např. jméno a příjmení zaměstnance. V každé tabulce lze definovat primární klíč, který má pro záznam jedinečnou hodnotu. Dále je zde možné vytvoření indexů pro rychlejší výběr dat.

3.6.2 Objekt „Dotaz“

V tomto objektu lze definovat uživatelský pohled na data z jedné nebo vícero tabulek. Access umožňuje tvorbu dotazů pomocí grafického prostředí, avšak je možné psát i vlastní definice v jazyce SQL, které nám dávají možnost data vybírat, aktualizovat, přidávat či odstraňovat nebo vytvářet nové tabulky.

3.6.3 Objekt „Formulář“

Tento objekt je zejména určen pro vstup nebo zobrazování uložených dat. Formuláře jsou přizpůsobeny pro uživatelský náhled na data, která jsou vybírána pomocí dotazů nebo tabulek a lze je také uzpůsobit pro spouštění maker či VBS skriptů jako reakci na nějakou uživatelskou událost (např. kliknutí myši na tlačítko, atd.). Formuláře poslouží i jako výstup pro tisk.

3.6.4 Objekt „Sestava“

Tento objekt je specifikován pro výpočty, tisky, formátování a souhrny vybraných dat. Před samotným tiskem je možné si sestavu dát k nahlédnutí.

3.6.5 Objekt „Datové stránky“

Jedná se o objekt obsahující HTML s podporou přístupu k datům MS Internet Explorer. Výstupy lze publikovat na intranetu pro zobrazování a úpravu dat v databázi.

3.6.6 Objekt „Makro“

V tomto objektu lze definovat jednu či více akcí, které požadujeme po Accessu, aby vykonal jako reakci na uživatelskou událost. Jako příklad lze uvést otevření nového formuláře po stisknutí tlačítka na původním formuláři. S využitím maker lze otvírat tabulky, spouštět a provádět definované dotazy, tisknout nebo prohlížet sestavy či spouštět procedury napsané ve VBA a nebo také spouštět ještě další makra.

3.6.7 Objekt „Modul“

Je to objekt, v němž jsou obsaženy uživatelem definované procedury, napsané jazykem VBA. Umožňují zachytávat chyby, což u maker toto nelze. Moduly mohou být samostatné objekty obsahující funkce, které lze zavolat z libovolného místa v aplikaci, též mohou být přímo přidruženy k nějakému formuláři nebo sestavě a odpovídat pouze na události tohoto přidruženého formuláře či sestavy.

3.7 Zásady databázových návrhů

Metodologii pro dobrý návrh počítačových aplikací poprvé navrhli v 60. letech uznávaní průmysloví konzultanti, jako byli James Martin, Edward Yourdon a Larry Constantine. Na úsvitu moderního počítačového zpracování bylo vytváření aplikace nebo oprava chybné aplikace tak nákladné, že tito experti často doporučovali, aby se 60 i více procent celkového času návrhu strávilo nad správným vytvořením návrhu, ještě než bude napsán jediný řádek kódu (Viescas, 2000).

V souvislosti se skutečností, že dnešní technologie výrazně pokročily vpřed, je i nadále velmi důležité věnovat prostor pro určení, co by měla v důsledku aplikace dělat, jaké data zpracovávat, jak je případně vyhodnocovat. Podcenění těchto faktů by mohlo mít za následek ztrátu drahocenného času, který by musel být věnován důslednému přepracování ve spojení s dohledáním, identifikací a opravou daného problému či procesu.

3.7.1 Identifikace úloh

Na počátku se jedná o představu, co by se vlastně mělo dělat a co by to vlastně mělo umět. Dále by měla být jednoznačná specifikace hlavních úloh dané aplikace, včetně úloh, které by mohly být realizovány v budoucnu.

3.7.2 Načrtnutí toku úloh

V tomto kroku by mělo dojít k tematickému rozdělení hlavních úloh a zařazení do sekvence úkolů, tak jak by měly jít za sebou. Např. nejprve bude muset být zadaná smlouva a až poté budou vytvořeny propočty hodnot.

3.7.3 Identifikování datových prvků

Po vytvoření seznamu úloh je nutné vytvořit seznamu dat, který bude vyžadován tou danou úlohou, jelikož každá úloha bude vyžadovat nějaká vstupní data. Úloha samozřejmě může dále s daty pracovat, např. data aktualizovat, přidávat nová, mazat nebo pouze prezentovat bez nějaké další akce nad danými daty.

3.7.4 Uspořádání dat

Zde pomůže prvek normalizace databáze, který nás navede jak co nejefektivněji data uložit a uspořádat do tabulek a dotazů využitých v databázi.

3.7.5 Navržení prototypu a uživatelského rozhraní

Po vytvoření struktury tabulek, lze snad napodobit průběh aplikace pomocí prostředků procedur Visual Basicu for Application (VBA). Pro potřeby testu lze vytvořit skutečné formuláře a sestavy, na kterých si lze nasimulovat vzhled.

3.7.6 Vytvoření aplikace

Pro použití Microsoft Accessu existují dva zásadní důvody: vývoj databází pro ukládání dat a výstavba aplikací pro správu dat (Solomon, 1998). U velmi jednoduchých aplikací možná zjistíme, že aplikací je přímo náš prototyp. Většina aplikací však bude vyžadovat, abyste pro plné zautomatizování všech úloh, které jste identifikovali ve svém návrhu, napsaly nějaký kód. Budete pravděpodobně také potřebovat vytvořit určité sjednocující formuláře, které zajistí přechod od jedné úlohy ke druhé. Můžeme například chtít vytvořit přepínací formuláře, které uživatelům vaší aplikace umožní určitou orientaci. Také budete možná potřebovat vytvořit dialogové formuláře, které budou přebírat vstup od

uživatelé a umožní uživatelům snadno filtrovat data, která chtějí použít pro nějakou konkrétní úlohu (Viescas, 2000).

3.7.7 Testování, přepracování a zdokonalování

V případě, že dokončujeme různé části naší aplikace tak bychom měli otestovat každou možnost, kterou nabízíme.

Pokud je to vůbec možné, měli byste nabídnout dokončené části své aplikace uživatelům, aby mohli váš kód otestovat a poskytnout vám zpětnou vazbu ohledně chodu aplikace. Protože Access je relační databáze s dobrými výpočetními a třídícími vlastnostmi, včetně nástrojů nezbytných k vytvoření příjemného uživatelského rozhraní, můžete ji využít pro vývoj snadno použitelných účetních a fakturačních aplikací (Solomon, 1998). Navzdory vašemu největšímu úsilí o identifikaci úloh a uspořádání jejich hladkého toku budou uživatelé po shlédnutí vaší aplikace v akci neustále přemýšlet o nových a lepších způsobech, jakými lze dosáhnout konkrétní úlohy. Uživatelé také často zjistí, že některé funkce, o které vás požádali, nakonec nejsou příliš užitečné. Včasné odhalení vyžadované změny během fáze implementace vám může ušetřit spoustu času s přepracováním (Viescas, 2000).

Proces zdokonalování a přepracování pokračuje i tehdy, když je aplikace zapojena do používání. Většina softwarových vývojářů zjišťuje, že poté co dokončili jednu verzi, musí často provádět změny v návrhu a vytvářet vylepšení. U velkých oprav byste měli začít opět od kroku 1, abyste byli schopni odhadnout celkový dopad požadovaných změn a mohli tyto změny hladce zaintegrovat do dřívější práce (Viescas, 2000).

3.8 SQL

3.8.1 Úvod do SQL

SQL, téměř univerzálně implementovaný relační jazyk, se liší od jiných jazyků jako jsou C, COBOL a Java, které jsou procedurální. Procedurální jazyk definuje to, jak by měly být operace v aplikaci prováděny, a pořadí, ve kterém jsou prováděny.

Neprocedurální jazyk se na straně druhé zabývá více výsledky operace. Základní softwarové prostředí určuje, jak budou operace zpracovány. Nedá se ovšem říci, že by SQL nepodporovalo procedurální funkce. Například uložené procedury, přidané do mnoha produktů RDBMS před mnoha lety, jsou součástí standardu SQL:1999 a umožňují procedurální zpracování (Sheldon, 2005).

3.8.2 Vývoj SQL

Na začátku 70.let, po publikování relačního datového modelu, začala společnost IBM vyvíjet jazyk a databázový systém, který by bylo možno použít pro implementaci tohoto modelu. Ve své první definici byl jazyk označován jako SEQUEL (Structured English Query Language) a po pozdější revizi byl přejmenován na SQL. Když společnost IBM vyvíjela relační databázový systém založený na SQL, jiné společnosti začaly vyvíjet své vlastní produkty založené na SQL. Ve skutečnosti společnost Relational Software, Inc., nyní Oracle Corporation, vydala svůj databázový systém ještě před tím, než společnost IBM uvedla svůj produkt na trh. Jak stále více společností vydávalo své produkty, stával se z SQL standardní jazyk pro relační databáze (Sheldon, 2005).

V roce 1986 vydala ANSI první publikovaný standard pro jazyk (SQL-86). Tento standard byl aktualizován v roce 1989 a znovu v roce 1992. SQL-92 představoval rozsáhlé rozšíření jazyka o některé funkce a vylepšení funkcí oproti předchozím verzím. O sedm let později, v roce 1999, byla vydána poslední verze standardu SQL, SQL:1999, představující další velký krok kupředu, kterým se SQL přiblížilo skutečným implementacím databázových systémů a potřebám jejich uživatelů (Sheldon, 2005).

3.8.3 Druhy příkazů jazyka SQL

Přestože je jazyk SQL považován z důvodu jeho neprocedurální povahy za podjazyk, je kompletním jazykem, v němž je možné vytvářet a spravovat databázové objekty, zabezpečovat tyto objekty a manipulovat s daty v objektech. Běžnou metodou kategorizace příkazů SQL je jejich rozdělování podle funkce, kterou provádí. Pomocí této metody je možno SQL rozdělit na tři druhy příkazů (Sheldon, 2005).

3.8.3.1 Jazyk definice dat (DDL – Data Definition Language)

Příkazy DDL se používají k vytváření, úpravám nebo odstraňování databázových objektů, jako např. tabulek, pohledů, schémat domén triggerů a uložených procedur. Klíčovými slovy SQL nejčastěji spojovanými s příkazy DDL jsou CREATE, ALTER a DROP (Sheldon, 2005).

Příkaz CREATE TABLE byste například použili pro vytvoření tabulky, ALTER TABLE pro změnu vlastností tabulky a DROP TABLE pro odstranění definice tabulky z databáze (Sheldon, 2005).

```
USE northwind
CREATE TABLE customer
(cust_id int, company varchar(40), contact varchar (30),
phone char(12))
GO
```

Obrázek 1 - Příklad kódu DDL

3.8.3.2 Jazyk pro řízení dat (DCL – Data Control Language)

Příkazy DCL umožňují řídit, kdo bude mít přístup k určitým objektům vaší databáze. V jazyce DCL můžete povolovat přístup nebo jej zamezovat pomocí příkazu GRANT nebo REVOKE, jež patří mezi primární příkazy DCL. Příkazy DCL také umožňují řídit typ přístupu každého uživatele k databázovým objektům. Můžete například určit, kteří uživatelé budou moci určitou skupinu dat pouze prohlížet, a kteří budou moci manipulovat s daty (Sheldon, 2005).

```
USE northwind
GRANT SELECT ON products TO public
GO
```

Obrázek 2 - Příklad kódu DCL

3.8.3.3 Jazyk pro manipulaci s daty (DML – Data Manipulation Language)

Příkazy DML se používají k prohlížení, přidávání, úpravám nebo odstraňování dat uložených ve vašich databázových objektech. Primárními klíčovými slovy spojovanými s příkazy DML jsou SELECT, INSERT, UPDATE a DELETE, jež všechny představují druhy příkazů, které budete pravděpodobně používat nejčastěji. Například příkaz SELECT můžete používat k načítání dat z tabulky a příkaz INSERT pro přidávání dat do tabulky (Sheldon, 2005).

```
USE northwind
SELECT categoryid, productname, produktid, unitprice
FROM products
GO
```

Obrázek 3 - Příklad kódu DML

4 Problematika evidence pracovní docházky

4.1 Identifikování současných potřeb

Záměrem mé bakalářské práce je stanovení současné potřeby a problematiky evidence pracovní docházky zaměstnanců dané společnosti. Tato práce by měla blíže specifikovat účelnost, potřebu a popis hodnot vztahujících se k tomuto tématu.

V záznamech evidence pracovní docházky by měly být evidovány hodnoty reprezentující údaje potřebné k vyhodnocování měsíční pracovní docházky jednotlivého zaměstnance společnosti k potřebám finančního a personálního oddělení. Tyto výsledné hodnoty jsou směrodatné pro zpracování mezd a dalších nutných evidencí, jako například záznam o termínu služební cesty či školení zaměstnance atd.

V této evidenci jsou jako hlavní hodnoty zahrnuty údaje o příchodu a odchodu zaměstnance společnosti vztahující se k pracovnímu dni a pracovnímu fondu stanoveným zákoníkem práce ČR ve znění, viz.příloha 1., kde tento zákon stanovuje týdenní pracovní fond zaměstnance firmy na 40 odpracovaných hodin v rámci pracovního týdne.

Tato evidence byla na požadavek finančního a personálního oddělení byla rozšířena o další položky.

Jedná se především o položky přerušení docházky během pracovní doby. Mezi tyto položky jsou zahrnuty následující stavy:

Přestávka na oběd, nepřítomnost z důvod onemocnění zaměstnance společnosti, přerušení ze soukromých důvodů, návštěva u lékaře, OČR, absolvování služební cesty, čerpání dovolené, záznam o návštěvě školení mimo budovu zaměstnavatele i v rámci místa výkonu práce a saldo, které obsahuje sumu odpracovaných hodin daného pracovního měsíce v rámci pracovního fondu daného měsíce.

4.2 Stavby přerušeni pracovní doby

Příslušné stavby přerušeni pracovní doby z různých důvodů:

1. Oběd – v rámci tohoto přerušeni v průběhu pracovní doby je doba na oběd dána v rozsahu 30 min.
2. Nemoc – zahrnuje nepřítomnost zaměstnance společnosti z důvodu onemocnění a na základě potvrzení lékaře o pracovní neschopnosti, podle ustanovení dle zákoníku práce.
3. Soukromě – nepřítomnost zaměstnance společnosti z důvodů soukromé záležitosti, tj. opuštění místa výkonu práce příslušného zaměstnance společnosti.
4. Lékař – opuštění místa zaměstnání příslušného zaměstnance z důvodu návštěvy lékaře
5. OČR – absence zaměstnance z důvodu ošetřování člena rodiny.
6. Služebně – zahrnuje služební cestu, pracovní schůzku, pracovní jednání, výjezd do terénu mimo místo výkonu práce příslušného zaměstnance společnosti.
7. Dovolena – čerpání dovolené nárokované ze zákoníku práce na zotavenou a v rozsahu interních předpisů daných zaměstnavatel (+5 dnů nad rámec stanovený zákoníkem práce v rámci bonusů udělovaných zaměstnanci zaměstnavatelem).
8. Školení – obsahuje informaci o čase, po který byl zaměstnanec společnosti byl přítomný na absolvovaném školení.
9. Saldo – zde jsou zaznamenány souhrnné hodnoty odpracované doby zaměstnancem společnosti. Tento záznam zahrnuje přehled docházky celkově za stanovený/vybraný rok a za aktuální rok, za stanovený/vybraný měsíc a aktuální měsíc.

Výše uvedené stavby rovněž slouží i k informaci o aktuálním místě, kde se daný zaměstnanec společnosti nachází.

4.3 Současná evidence pracovní docházky

Při analýze současného stavu využívané aplikace, která je určena pro evidenci pracovní docházky pracovníků dané firmy bylo zjištěno, že v některých směrech se již stává nevyhovující. Tato aplikace sice byla v určitých obecných směrech postačující, ale postupem času a potřeb evidovat další údaje bylo nutné rozšíření určitých oblastí.

Zásadním problémem, který v této aplikaci byl zaznamenán, a který vyplynul z analýzy, byla evidence služebních cest a práce s časem, který byl v rámci služební cesty vynaložen. Samozřejmě se nejednalo pouze o evidenci a sledování služebních cest, ale i dalších stavů přerušení pracovní doby zaměstnancem.

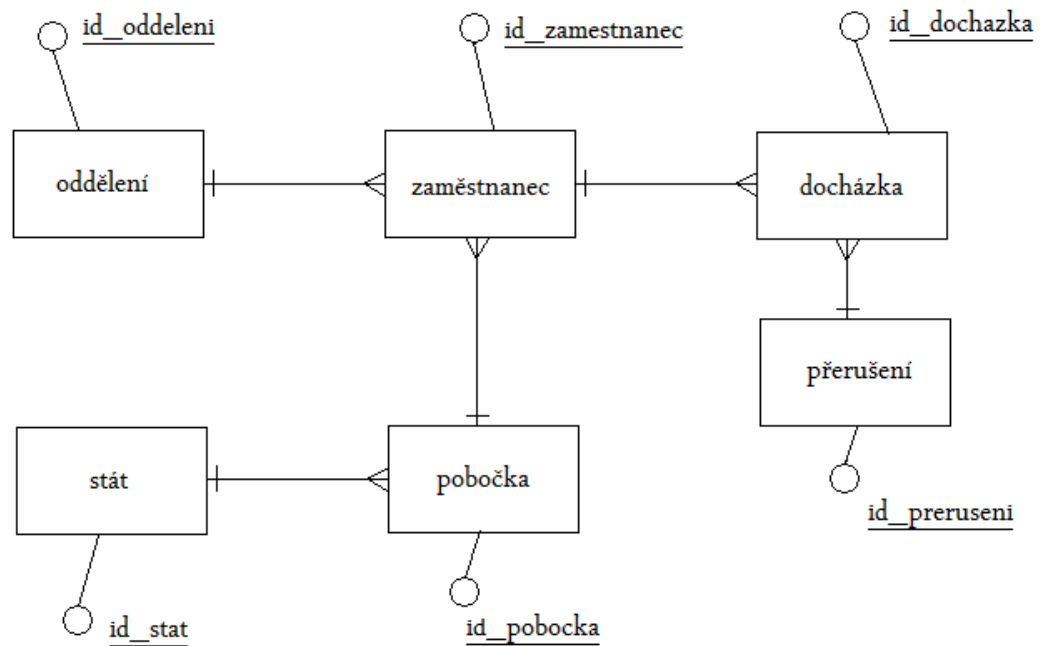
4.4 Tvorba nové evidence pracovní docházky

Před samotným vytvořením nové evidence pracovní docházky bylo nutné zjistit jaké údaje bude nutné evidovat, jaký budou mít tyto údaje další dopad a samozřejmě byla i práce a zpracování těchto údajů.

4.4.1 Zaznamenávání informací do tabulek

Aby bylo možné ukládat patřičné informace, bylo vytvořeno několik tabulek, které jsou popsány níže. Tabulky jsou koncipovány tak, aby pojmuly veškeré potřebné údaje.

Mezi klíčové tabulky, které byly vytvořeny, patří tabulka, do které se budou zapisovat informace o zaměstnanci (tbl_zamestnanci), informace o pobočce (tbl_pobočka), informace o pracovní době a daném přerušení (tbl_dochazka). Dále pak jsou zde tabulky obsahující informace o svátcích či dnech pracovního klidu (tbl_svatek), tabulka specifikující název oddělení (tbl_oddeleni) atd., viz. příloha 2.



Obrázek 4 - E-R model evidence docházky s definicí primárních klíčů

4.4.2 Datová normalizace

Níže je zobrazení ukázky datové normalizace u uvedených tabulek (tbl_zamestnanec, tbl_oddeleni, tbl_pobočka, tbl_stat).

tbl_zamestnanec				
id_zamestnanec	zamestnanec_jmeno	zamestnanec_prijmeni	id_oddeleni	id_pobočka
1	Josef	Novák	1	4
2	Anna	Radová	3	4
(AutoNumber)			0	0

tbl_oddeleni		
id_oddeleni	oddeleni_nazev	oddeleni_zkratka
1	Finanční oddělení	FAD
2	Operační oddělení	OPD
3	Obchodní oddělení	SAD
4	Oddělení IT	IT

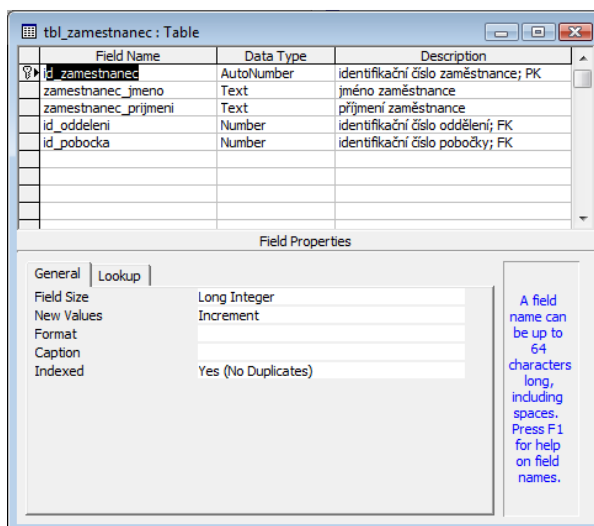
tbl_stat		
id_stat	stat_nazev	stat_zkratka
1	Česká republika	CZ
2	Slovenská republika	SK
3	Rakousko	AT

tbl_pobočka					
id_pobočka	pobočka_nazev	pobočka_adresa_ulice	pobočka_adresa_psc	pobočka_adresa_mesto	id_stat
1	A	Za letištěm 1	11111	Praha	1
(AutoNumber)			0		0

4.4.2.1 Tabulka „Zaměstnanci“

V této tabulce pod názvem tbl_zamestnanec zaznamenáváme informace týkající se zaměstnanců dané společnosti, tj. jméno, příjmení, zařazení do oddělení a na danou pobočku. Jednotlivá evidovaná pole v této tabulce jsou následující:

- id_zamestnanec – automatické číslo; primary key (dále v textu již jen jako *PK*)
- zamestnanec_jmeno – text; velikost 50; jméno zaměstnance
- zamestnanec_prijmeni – text; velikost 50; příjmení zaměstnance
- id_oddeleni – číslo; propojení na tabulku oddělení, foreign key (dále v textu uvedeno pod zkratkou FK)
- id_pobočka – číslo; propojení na tabulku pobočka, FK



Obrázek 5 - Návrhové zobrazení tabulky zaměstnanců

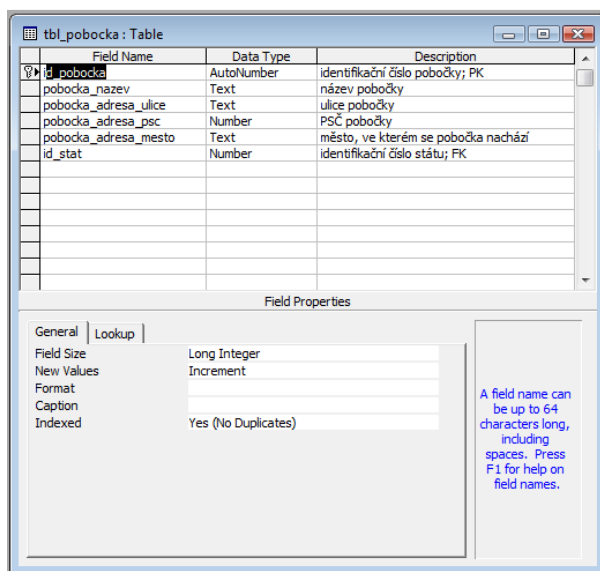
4.4.2.2 Tabulka „Pobočka“

Pod názvem tbl_pobočka je vytvořena tabulka, která uchovává informace o pobočce dané společnosti.

Pole v této tabulce jsou následující:

- id_pobočka – automatické číslo; *PK*
- pobočka_nazev – text; délka 30; název uvedené pobočky
- pobočka_adresa_ulice – text; délka 30; ulice dané pobočky

- pobočka_adresa_psc – číslo; fomát PSČ; PSČ pobočky
- pobočka_adresa_mesto – text; délka 20; uvedení města, ve kterém se pobočka nachází
- id_stat – číslo; propojení na tabulku evidující informace o státu, FK



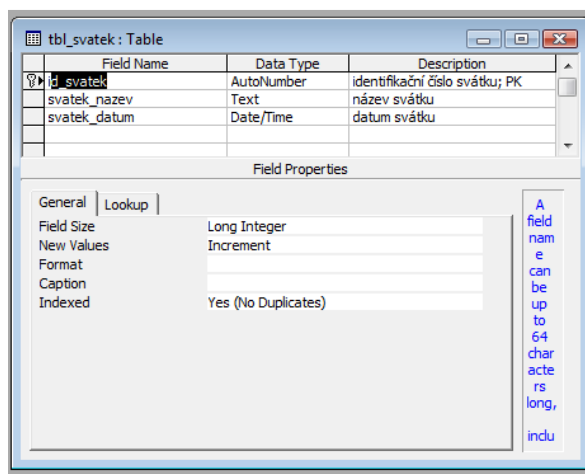
Obrázek 6 - Návrhové zobrazení tabulky pobočky

4.4.2.3 Tabulka „Svátky“

V této tabulce tbl_svatek jsou zapisovány informace o státních svátcích a dnech pracovního klidu k příslušnému datumu.

Pole v tabulce jsou:

- id_svatek – automatické číslo; PK
- svatek_nazev – text; délka 60; název svátku
- svatek_datum – datum a čas; dlouhý formát; datum, ke kterému příslušný svátek přísluší



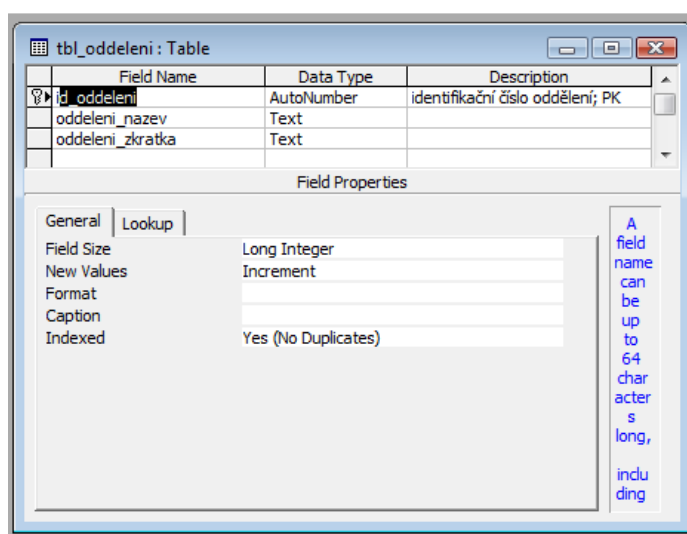
Obrázek 7 - Návrhové zobrazení tabulky svátků

4.4.2.4 Tabulka „Oddělení“

V tbl_oddeleni jsou založeny informace o příslušných odděleních včetně jejich zkratk. Uvažujeme tak, že pro každou pobočku jsou oddělení v organizační struktuře firmy shodná.

Pole jsou:

- id_oddeleni – automatické číslo; *PK*
- oddeleni_nazev – text; délka 30; název oddělení
- oddeleni_zkratka – text; délka 3; zkratka daného oddělení



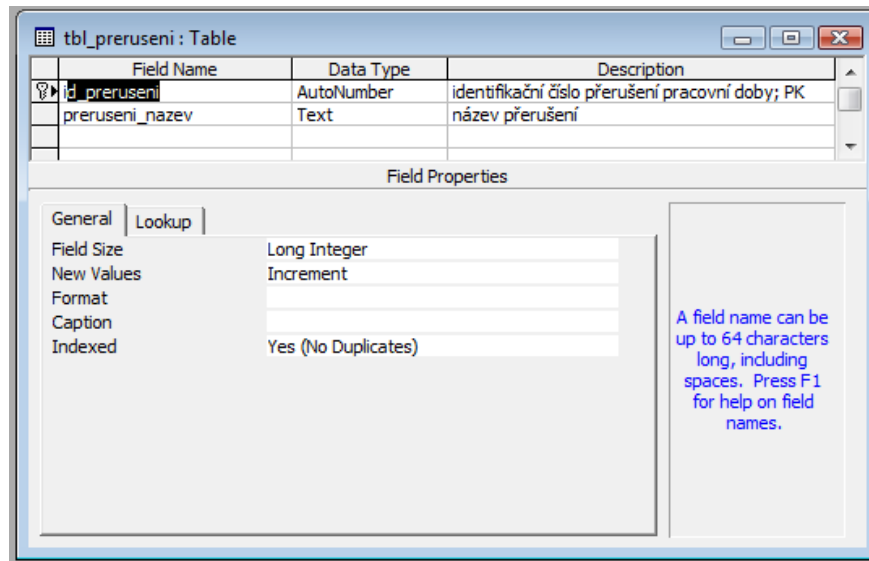
Obrázek 8 - Návrhové zobrazení tabulky oddělení

4.4.2.5 Tabulka „Přerušení pracovní doby“

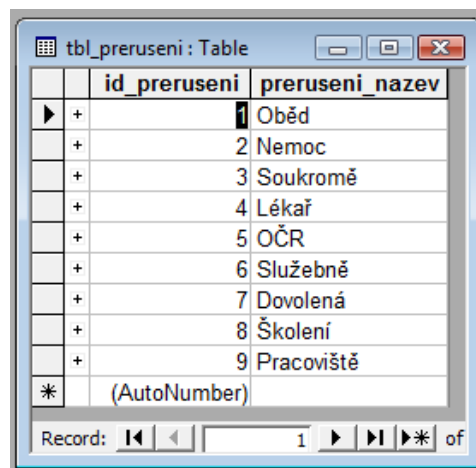
Tabulka tbl_preruseni v sobě zaznamenává popis stavu přerušení pracovní doby.

Pole jsou následující:

- id_preruseni – automatické číslo; *PK*
- preruseni_nazev – text; délka 10; název daného přerušení



Obrázek 9 - Návrhové zobrazení tabulky přerušení pracovní doby

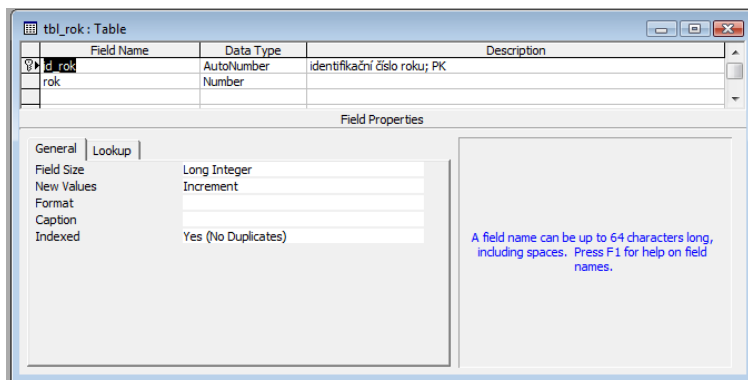


Obrázek 10 - Datový pohled na tabulku přerušení pracovní doby

4.4.2.6 Tabulka „Rok“

V této tabulce tbl_rok se eviduje hodnota roku. Tato tabulka je má vazbu s tabulkou pro určení týdenního pracovního fondu pro daný rok. Pole jsou:

- id_rok – automatické číslo; *PK*
- rok – číslo; hodnota roku

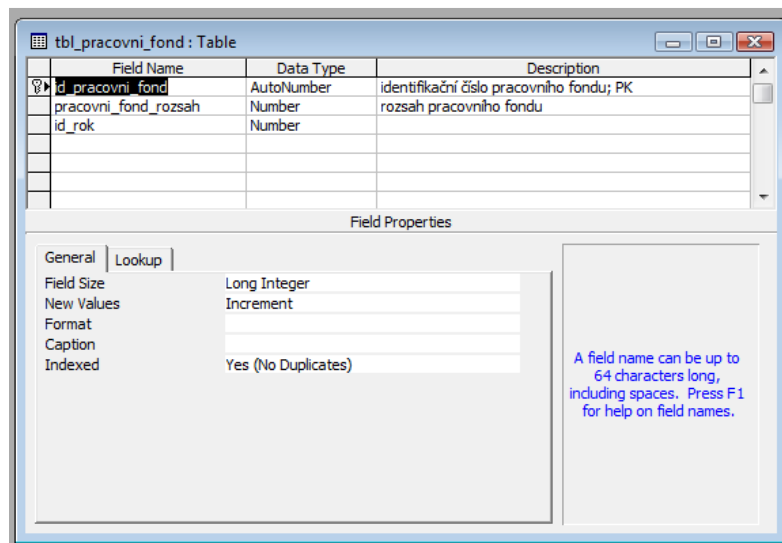


Obrázek 11 - Návrhové zobrazení tabulky evidující roky

4.4.2.7 Tabulka „Pracovní fond“

V tbl_pracovni_fond je zaznamenána evidence pracovního fondu pro daný rok. Pole jsou:

- id_pracovni_fond – automatické číslo; *PK*
- pracovni_fond_rozsah – číslo; udává hodnotu rozsahu daného pracovního fondu
- id_rok – číslo; propojení na tabulku se záznamem roků, FK

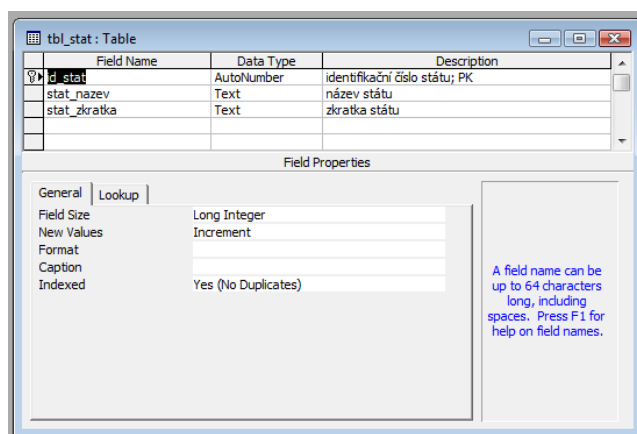


Obrázek 12 - Návrhové zobrazení tabulky pracovního fondu

4.4.2.8 Tabulka „Stát“

Tato tabulka má za úkol zaznamenávat informace o státu. Pole této tabulky jsou následující:

- id_stat – automatické číslo; *PK*
- stat_nazev – text; délka 20; název daného státu
- stat_zkratka – text; délka 2; zkratka názvu státu



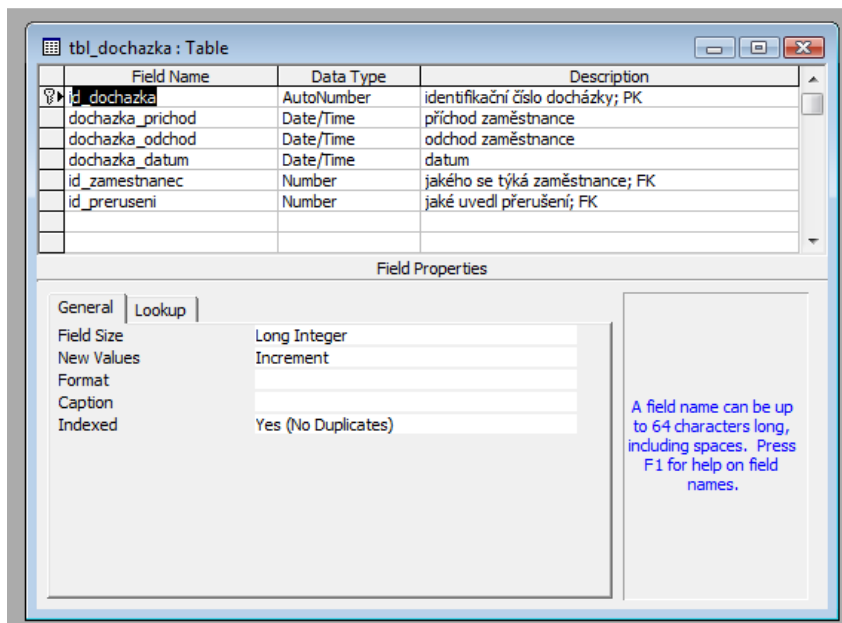
Obrázek 13 - Návrhové zobrazení tabulky států

4.4.2.9 Tabulka „Docházka“

Tato nejdůležitější tabulka, tbl_dochazka ukládá informace o konkrétní pracovní době a jejím stavu a to k danému zaměstnanci.

Pole jsou tato:

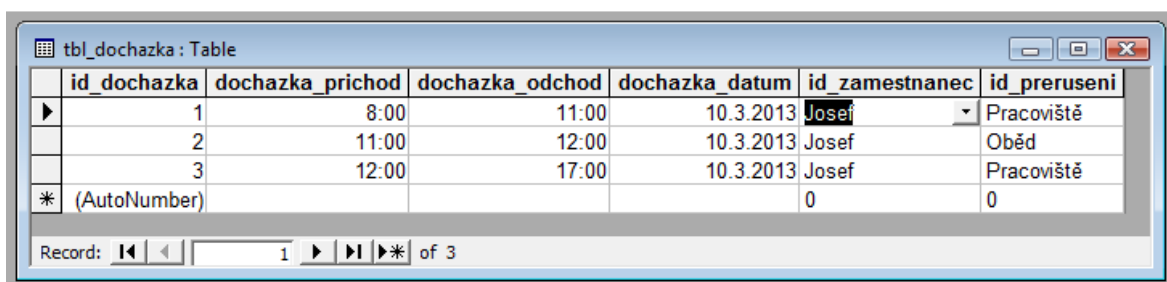
- id_dochazka – automatické číslo; *PK*
- dochazka_prichod – datum a čas; zaznamenává čas příchodu zaměstnance
- dochazka_odchod – datum a čas; ukládání informace o odchodu zaměstnance
- dochazka_datum – datum a čas; evidence příslušného datumu
- id_zamestnanec – číslo; propojení na tabulku zaměstnanců, FK
- id_preruseni – číslo; propojeni na tabulku přerušeni, FK



Obrázek 14 - Návrhové zobrazení tabulky evidující docházku

5 Zobecněná funkce nové evidence pracovní docházky

Data ukládaná do evidence pracovní docházky mají podobu, viz. obrázek 14. V takto podobné struktuře se bude ukládat jakýkoliv záznam daného zaměstnance a informace o jeho pracovní činnosti v daném čase.



id_dochazka	dochazka_prichod	dochazka_odchod	dochazka_datum	id_zamestnanec	id_preruseni
1	8:00	11:00	10.3.2013	Josef	Pracoviště
2	11:00	12:00	10.3.2013	Josef	Oběd
3	12:00	17:00	10.3.2013	Josef	Pracoviště
*(AutoNumber)				0	0

Obrázek 15 - Data evidované v tabulce docházky

Nejpodstatnější částí této evidence je zajištění informací v případě, kdy je zaměstnanec vyslán na služební cestu (tj. mimo místo výkonu práce) a ve většině případech tento stav přerušení pracovní doby přesáhne denní fond 8 odpracovaných hodin. Systémem by mělo být toto ošetřeno tak, že každý den je nastaven na fond 8 hodinové pracovní doby s pauzou na oběd v délce 30min. Tedy nebude mít závislost na odpracované době délka služební cesty, jelikož vždy bude nastaveno dopočtení do 8 hodinové pracovní doby.

Pomocí této evidence je možné určovat i aktuální stav přerušení zaměstnance. Pro vyhledání zaměstnance se v aplikaci použije následující zápis SQL kódu (zobrazí výpis, Jméno, Příjmení, oddělení, datum, apříchod a odchod):

```
SELECT tbl_zamestnanec.zamestnanec_jmeno,  
tbl_zamestnanec.zamestnanec_prijmeni, tbl_oddeleni.oddeleni_nazev,  
tbl_oddeleni.oddeleni_zkratka, tbl_dochazka.dochazka_datum,  
tbl_dochazka.dochazka_prichod, tbl_dochazka.dochazka_odchod,  
tbl_preruseni.preruseni_nazev  
  
FROM tbl_preruseni INNER JOIN (tbl_oddeleni INNER JOIN (tbl_zamestnanec  
INNER JOIN tbl_dochazka ON tbl_zamestnanec.id_zamestnanec =  
tbl_dochazka.id_zamestnanec) ON tbl_oddeleni.id_oddeleni =  
tbl_zamestnanec.id_oddeleni) ON tbl_preruseni.id_preruseni = tbl_dochazka.id_preruseni  
  
WHERE (((tbl_dochazka.dochazka_datum)=Date());
```


Pro vstup dat slouží tento jednoduchý vstupní formulář. Kde uživatel zadá příchod, odchod a příslušné přerušení.

Jméno: Petr Příjmení: Novák
Datum: 27.2.2013 Vyber stav:
Příchod: 7:59
Odchod: 17:11
Nemoc
Soukromě
Lékař
OČR
Služebně
Dovolená
Školení
Pracoviště
Otevřít nový záznam Uložit záznam Ukončit záznam

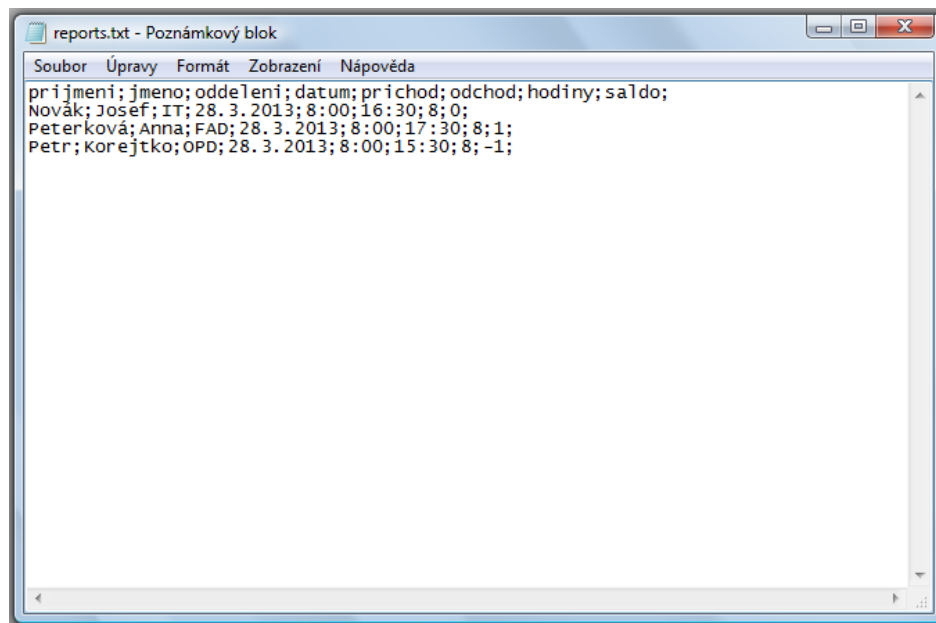
Obrázek 16 - Vstupní formulář

Ve formuláři uvedeném níže se zadávají kritéria pro filtr dat, která se budou dále exportovat.

Dle filtru
Od:
Do:
Oddělení:
Zaměstnanec:
 Vše
Export dat Ukončit export

Obrázek 17 - Vstupní formulář pro určení filtru pro export dat

Jako výstupem je například tento textový soubor, který může sloužit jako vstup pro vyhodnocování v dalších systémech.



```
reports.txt - Poznámkový blok
Soubor  Úpravy  Formát  Zobrazení  Nápověda
pr i jmeni ; jmeno ; oddeleni ; datum ; pr ichod ; odchod ; hodiny ; sa l do ;
Novák ; Josef ; IT ; 28. 3. 2013 ; 8 : 00 ; 16 : 30 ; 8 ; 0 ;
Peterková ; Anna ; FAD ; 28. 3. 2013 ; 8 : 00 ; 17 : 30 ; 8 ; 1 ;
Petr ; korejtko ; OPD ; 28. 3. 2013 ; 8 : 00 ; 15 : 30 ; 8 ; -1 ;
```

Obrázek 18 - Ukázka exportovaných dat

6 Závěr

Cílem této bakalářské práce bylo vymezení problematiky současné evidence docházky zaměstnanců dané společnosti a pomocí nástroje MS Access navrhnout řešení.

V době, kdy bylo stávající řešení implementováno, dostatečně splňovalo kladené požadavky. V aplikaci evidující docházku bylo tedy pro tu dobu zapisováno vše potřebné. Informace se zaznamenávaly v uskupení následujících informací: byl zde zapsán čas příchodu pracovníka na pracoviště, potřebná pauza na oběd a čas odchodu pracovníka z místa výkonu práce.

Tyto záznamy byly zároveň rozšířeny o možnost připojit informaci o přerušení pracovní doby daného zaměstnance či připojit k danému dni nějakou situaci. Mezi situace, které zde bylo možné přiřadit k danému dni patřil například záznam, že se ke jmenovanému dni vztahoval státní svátek či den pracovního klidu, čerpání dovolené, návštěva lékaře, ošetřování člena rodiny nebo služební cesta.

Postupem času a se stupňujícími se dalšími požadavky bylo nutné tento postup a zapisování upravit v rámci dotazů, které vyplynuly z analýz a interních potřeb.

Uživatel evidence pracovní docházky tedy může například vidět, kde se vybraný pracovník zrovna nachází tj. jestli zrovna neabsolvuje školení, není na služební cestě nebo u lékaře.

S těmito stavy bylo rovněž nutné i efektivně pracovat. Další možné zpracování ve stávajícím systému evidence nebylo reálné, jelikož to nebylo v možnostech přístupných touto aplikací.

Bylo tedy nutné i zároveň zmapovat, jak by se dala co nejlépe využít ukládaná data v novém řešení. Konkrétněji se jednalo o zpracování informace o přerušení pracovní doby v rámci služební cesty.

Většinou nastávala situace, kdy byl pracovník absolvující služební cestu nad rámec odpracovaných hodin a to v důsledku nesprávného či nekorektního zadání hodnot do stávajícího řešení.

Nový způsob, který tento záznam uchovává, pracuje s touto hodnotou tak, že záznam je korektní a reflektuje správné hodnoty. K zefektivnění není určen pouze záznam o služební cestě, ale samozřejmě i ostatní evidované stavy pracovní doby u každého zaměstnance.

Výsledkem tohoto řešení jsou i patřičně vydefinované reporty, které lze účinně využít i pro zpracování do jiných systémů.

Dále nesmíme pominout fakt, že evidence pracovní docházky byla řešena pomocí relačně databázového nástroje Access od společnosti Microsoft.

Když shrnu výhody a nevýhody, tak můžu říci, že určitě převládají více výhody než nevýhody. Výhody z pohledu použité nástroje jsou takové, že zde není kladen vysoký důraz na odbornost vývojářů či pracovníků, kteří budou s aplikací pracovat na vývojářské úrovni. Poměrně snadná tvorba dotazů, formulářů, sestav v příjemném grafickém prostředí. Snadná orientace zabezpečena známým a snadno nebo dosti orientovaným prostředím MS Windows. Ovšem, že daný nástroj nemá jen svoje výhody. Mezi jednu z nevýhod patří občasné nekompatibilita ovládacích prvků. Jeho primární určení je jako kancelářský nástroj.

Mezi výhody z pohledu řešené problematiky evidence pracovní docházky patří bezesporu práce se stavy přerušení a možnosti kontroly, kde se daný pracovník v určený čas pohybuje. Samozřejmostí je i vylepšení o vyexportované soubory, se kterými má uživatel možnost nadále pracovat, např. jako import do jiných systému, které mohou tvořit nějakou statistiku atd.

Takovéto řešení v evidenci pracovní docházky lze určitě doporučit do malých či středně malých firem.

7 Seznam použitých zdrojů

7.1 Tištěné dokumenty

JERKE, Noel. *Microsoft Office Access 2003 pro pokročilé*. 1. vydání. Brno: CP Books, 2005. 348 s. ISBN 80-251-0723-X

McFEDRIES, Paul. *VBA for Microsoft Office 2000*. 1. vydání. Brno: UNIS Publishing, 2008. 1013 s. ISBN 80-86097-43-9

SHELDON, Robert. *SQL začínáme programovat*. 1. vydání. Praha: Grada Publishing a.s., 2005. 500 s. ISBN 80-247-0999-6

SOLOMON, Christine. *Tvorba aplikací v Microsoft Office 97*. 1. vydání. Praha: Computer Press, 1998. 598 s. ISBN 80-7226-025-1

VIESCAS, John. *Mistrovství v Microsoft Access 2000*. 1. vydání. Praha: Computer Press, 2000. 809 s. ISBN 80-7226-274-2

7.2 Elektronické zdroje

Normalizace databáze. Wikipedie, otevřená encyklopedie. [Online] Wikimedia Foundation. [Citace: 27.11.2012]

Dostupné z: <http://cs.wikipedia.org/wiki/Normalizace_datab%C3%A1ze>

ActiveX. Wikipedie, otevřená encyklopedie. [Online] Wikimedia Foundation. [Citace: 5.1.2013]

Dostupné z: <<http://cs.wikipedia.org/wiki/ActiveX>>

Dynamická výměna dat.[Online]

Dostupné z: <<http://www.sfiles.host.sk/builder/3/CBUILDL4.html>>

8 Seznam obrázků

Obrázek 1 - Příklad kódu DDL.....	27
Obrázek 2 - Příklad kódu DCL.....	27
Obrázek 3 - Příklad kódu DML.....	28
Obrázek 4 - E-R model evidence docházky s definicí primárních klíčů	32
Obrázek 5 - Návrhové zobrazení tabulky zaměstnanců.....	33
Obrázek 6 - Návrhové zobrazení tabulky pobočky	34
Obrázek 7 - Návrhové zobrazení tabulky svátků.....	35
Obrázek 8 - Návrhové zobrazení tabulky oddělení	35
Obrázek 9 - Návrhové zobrazení tabulky přerušení pracovní doby	36
Obrázek 10 - Datový pohled na tabulku přerušení pracovní doby	36
Obrázek 11 - Návrhové zobrazení tabulky evidující roky	37
Obrázek 12 - Návrhové zobrazení tabulky pracovního fondu.....	38
Obrázek 13 - Návrhové zobrazení tabulky států.....	38
Obrázek 14 - Návrhové zobrazení tabulky evidující docházku.....	39
Obrázek 15 - Data evidované v tabulce docházky.....	40
Obrázek 16 - Vstupní formulář.....	41
Obrázek 17 - Vstupní formulář pro určení filtru pro export dat.....	41
Obrázek 18 - Ukázka exportovaných dat.....	42

9 Přílohy

9.1 Příloha 1. - Zákonník práce „Pracovní doba a doba odpočinku“

Část čtvrtá

Pracovní doba a doba odpočinku

Hlava I: Obecná ustanovení o pracovní době a délka pracovní doby

§ 78

(1) Pro účely úpravy pracovní doby a doby odpočinku je

- a) pracovní dobou doba, v níž je zaměstnanec povinen vykonávat pro zaměstnavatele práci a doba, v níž je zaměstnanec na pracovišti připraven k výkonu práce podle pokynů zaměstnavatele,
- b) dobou odpočinku doba, která není pracovní dobou,
- c) směnou část týdenní pracovní doby bez práce přesčas, kterou je zaměstnanec povinen na základě předem stanoveného rozvrhu pracovních směn odpracovat,
- d) dvousměnným pracovním režimem režim práce, v němž se zaměstnanci vzájemně pravidelně střídají ve 2 směnách v rámci 24 hodin po sobě jdoucích,
- e) třisměnným pracovním režimem režim práce, v němž se zaměstnanci vzájemně pravidelně střídají ve 3 směnách v rámci 24 hodin po sobě jdoucích,
- f) nepřetržitým pracovním režimem režim práce, v němž se zaměstnanci vzájemně pravidelně střídají ve směnách v nepřetržitém provozu zaměstnavatele v rámci 24 hodin po sobě jdoucích,
- g) nepřetržitým provozem provoz, který vyžaduje výkon práce 24 hodin denně po 7 dnů v týdnu,
- h) pracovní pohotovostí doba, v níž je zaměstnanec připraven k případnému výkonu práce podle pracovní smlouvy, která musí být v případě naléhavé potřeby vykonána nad rámec jeho rozvrhu pracovních směn. Pracovní pohotovost může být jen na jiném místě dohodnutém se zaměstnancem, odlišném od pracovišť zaměstnavatele,
- i) práci přesčas práce konaná zaměstnancem na příkaz zaměstnavatele nebo s jeho souhlasem nad stanovenou týdenní pracovní dobu vyplývající z předem stanoveného rozvržení pracovní doby a konaná mimo rámec rozvrhu pracovních směn. U zaměstnanců s kratší pracovní dobou je práci přesčas práce přesahující stanovenou týdenní pracovní dobu; těmto zaměstnancům není možné práci přesčas nařídit. Práci přesčas není, napracovává-li zaměstnanec práci konanou nad stanovenou týdenní pracovní dobu pracovní volno, které mu zaměstnavatel poskytl na jeho žádost,
- ~~j) **týdnem každé období na sebe navazujících 7 dnů,**~~
- j) ~~k)~~ noční práci práce konaná v noční době; noční doba je doba mezi 22. a 6. hodinou,
- k) ~~l)~~ **zaměstnancem pracujícím v noci zaměstnanec, který odpracuje během noční doby nejméně 3 hodiny ze své pracovní doby v rámci 24 hodin po sobě jdoucích v průměru alespoň jednou týdně v období, uvedeném v § 94 odst. 1, zaměstnancem pracujícím v noci zaměstnanec, který během noční doby pravidelně odpracuje nejméně 3 hodiny ze své pracovní doby v rámci 24 hodin po sobě jdoucích.**
- l) rovnoměrným rozvržením pracovní doby rozvržení, při kterém zaměstnavatel rozvrhuje na jednotlivé týdny stanovenou týdenní pracovní dobu, popřípadě kratší pracovní dobu,
- m) nerovnoměrným rozvržením pracovní doby rozvržení, při kterém zaměstnavatel nerozvrhuje rovnoměrně na jednotlivé týdny stanovenou týdenní pracovní dobu, popřípadě kratší pracovní dobu, s tím, že průměrná týdenní

pracovní doba nesmí přesáhnout stanovenou týdenní pracovní dobu, popřípadě kratší pracovní dobu, za období nejvýše 26 týdnů po sobě jdoucích. Jen kolektivní smlouva může toto období vymezit nejvýše na 52 týdnů po sobě jdoucích.

(2) Ustanovení odstavce 1 písm. d) až f) platí i v případě, kdy při pravidelném střídání zaměstnanců ve směnách dojde k souběžnému výkonu práce zaměstnanců navazujících směn, avšak jen po dobu nejvýše 1 hodiny.

§ 79 Stanovená týdenní pracovní doba

(1) Délka stanovené týdenní pracovní doby ~~činí nesmí překročit~~ činí nesmí překročit 40 hodin týdně.

(2) Délka stanovené týdenní pracovní doby ~~činí nesmí překročit~~ u zaměstnanců

a) pracujících v podzemí při těžbě uhlí, rud a nerudných surovin, v důlní výstavbě a na báňských pracovištích geologického průzkumu 37,5 hodiny týdně,

b) s třísměnným a nepřetržitým pracovním režimem 37,5 hodiny týdně,

c) s dvousměnným pracovním režimem 38,75 hodiny týdně.

(3) Zkrácení stanovené týdenní pracovní doby bez snížení mzdy pod rozsah stanovený v odstavcích 1 a 2 může obsahovat jen kolektivní smlouva nebo vnitřní předpis. Zkrácení stanovené týdenní pracovní doby podle věty první nesmí však provést zaměstnavatel uvedený v § 109 odst. 3.

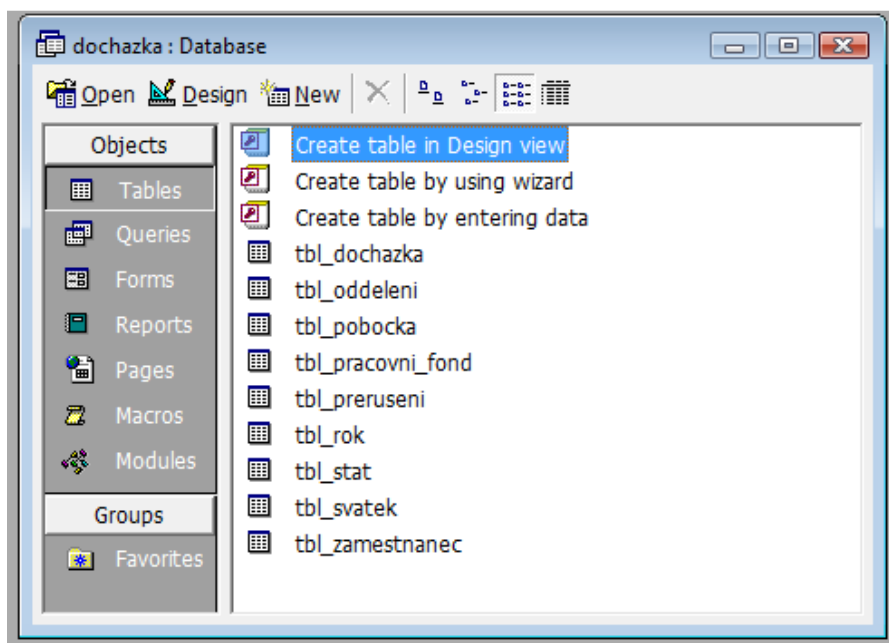
§ 79a

U zaměstnance mladšího než 18 let nesmí délka směny v jednotlivých dnech překročit 8 hodin a ve více **základních** pracovněprávních vztazích podle § 3 věty druhé nesmí délka týdenní pracovní doby ve svém souhrnu překročit 40 hodin týdně.

§ 80 Kratší pracovní doba

Kratší pracovní doba pod rozsah stanovený v § 79 může být sjednána pouze mezi zaměstnavatelem a zaměstnancem. Zaměstnanci přísluší mzda nebo plat, které odpovídají sjednané kratší pracovní době. ~~Byla-li sjednána kratší pracovní doba, přísluší zaměstnanci mzda nebo plat, které odpovídají této kratší pracovní době.~~

9.2 Příloha 2. – Přehled tabulek



9.3 Příloha 3. – Relationships (Relace)

