

Nadstavbový modul ERP systému pro výrobu a řízení

ERP System Upgrade Module for Production and Management

Bc. Michal Lekeš

Diplomová práce
2012



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2011/2012

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Michal LEKEŠ**
Osobní číslo: **A10890**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Bezpečnostní technologie, systémy a management**

Téma práce: **Nadstavbový modul ERP systému pro výrobu a řízení**

Zásady pro vypracování:

1. Analyzujte problematiku a vypracujte literární rešerši na dané téma.
2. Navrhněte a vytvořte strukturu databáze pro nadstavbový modul ERP systému.
3. Navrhněte a vytvořte aplikaci, která bude obsahovat potřebné formuláře, sestavy, funkce, procedury a moduly pro veškerou evidenci dat.
4. Vytvořte další možnosti aplikace – funkce a procedury nepostradatelné pro analýzy, které jsou důležité pro optimalizaci výrobních nákladů.
5. Popište způsob implementace nadstavbového modulu.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. BRUST, Andrew, J. ; FORTE , Stephen. Mistrovství v programování SQL Serveru 2005. Brno : COMPUTER PRESS, 26.11.2007. 848 s. ISBN 978-80-251-1607-4.
2. WALTERS, Robert, E. ; COLES, Michael; RAE, Robert. Mistrovství v programování Microsoft SQL Serveru 2008 : Komplexní průvodce databázového experta. Brno : COMPUTER PRESS, 2009. 864 s. ISBN 978-80-251-2329-4.
3. LACKO, Luboslav . Jak vyzrát na Microsoft SQL Server 2008 : Správa, konfigurace, programování. Brno : COMPUTER PRESS, 28.05.2009. 472 s. ISBN 978-80-251-2101-6.
4. HOTEK, Mike. Microsoft SQL Server 2008 : Krok za krokem. Brno : COMPUTER PRESS, 08.06.2009. 488 s. ISBN 978-80-251-2466-6.
5. HELD, Bernd. Access VBA. Brno : COMPUTER PRESS, 2006. 639 s. ISBN 80-251-1112-1.
6. FEDDEMA, Helen. Mistrovství v Microsoft Access 2002. Brno : COMPUTER PRESS, 2002. 604 s. ISBN 80-7226-725-6.
7. MORKEŠ, David. Microsoft Office Access 2003 Podrobná uživatelská příručka. Brno : COMPUTER PRESS, 2004. 350 s. ISBN 80-251-0179-7.

Vedoucí diplomové práce:

doc. Ing. Zdenka Prokopová, CSc.

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

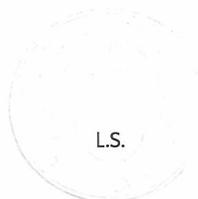
24. února 2012

Termín odevzdání diplomové práce:

15. května 2012

Ve Zlíně dne 24. února 2012


prof. Ing. Vladimír Vašek, CSc.
děkan




doc. RNDr. Vojtěch Křesálek, CSc.
ředitel ústavu

ABSTRAKT

Diplomová práce se zabývá vývojem nového Podnikového informačního systému LEKY, který byl vytvořen jako nadstavbový modul výroby ERP Systému SAP. Podnikový informační systém LEKY usnadní složité výpočty, optimalizuje výrobní materiál s návazností na evidenci skladových zásob, zjednoduší zadání, zprůhlední datový tok, sjednotí kalkulační vzorce, procesy a optimalizuje počty pracovníků přípravy výroby a zákaznického servisu. Teoretická část se zabývá modelací a návrhem databáze na platformě SQL Serveru a aplikace na platformě Microsoft Access.

Praktická část popisuje nejen jednotlivé systémové objekty včetně jejich vlastností, ale také obsahuje ukázky funkcí a procedur zdrojových kódů, které byly použity při vývoji jednotlivých modulů.

Klíčová slova: ERP systém SAP, Microsoft SQL Server 2008, podnikové informační systémy, automatizované informační systémy

ABSTRACT

The presented thesis deal with the development of a new Corporate information system LEKY, which was developed as an overlay module of SAP ERP system. The Corporate information system LEKY facilitates complicated calculations, optimizes manufacturing material in connection with the storing evidence, simplifies assignments, makes data flow more transparent, unifies counting models and processes, and optimizes the number of employees of production preparation and of the customer service.

The theoretical part of the thesis deals with the database modeling and design on the SQL Server platform and application on the Microsoft Access platform.

The practical part of the thesis describes not only individual system objects with their qualities but there are also examples of source code procedure, which were used during the development of individual modules.

Keywords: SAP ERP System, Microsoft SQL Server 2008, Corporate information systems, automated information systems

Rád bych poděkoval doc. Ing. Zdence Prokopové, CSc. za cenné rady v průběhu vývoje projektu a za její odborné vedení a pomoc při vypracování této diplomové práce.

Dále bych chtěl poděkovat společnosti SCA Packaging s.r.o., za důvěru, kterou do mne vložila a umožnila mi implementovat Nadstavbový modul ERP systému pro výrobu a řízení.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	12
1 NÁVRH INFORMAČNÍHO SYSTÉMU	13
1.1 NÁVRH DATABÁZE.....	13
2 PLATFORMA MICROSOFT SQL SERVER 2008	14
2.1 SQL SERVER 2008 STANDARD EDITION.....	15
2.1.1 Databázový modul SQL Server 2008.....	16
2.1.1.1 Relační modul	17
2.1.1.2 Databázové diagramy.....	17
2.2 SQL SERVER MANAGEMENT STUDIO	18
2.3 NÁVRH TABULEK.....	19
2.3.1 Určení datových typů polí tabulky	19
2.3.1.1 Číselná data.....	20
2.3.1.2 Znaková data.....	20
2.3.1.3 Datová a časová data.....	21
2.3.1.4 Binární data.....	21
2.3.2 Dělení tabulek a indexů do oddílů	22
2.3.2.1 Základní dělení tabulek indexů do oddílů:	22
2.4 POHLEDY(VIEWS).....	24
2.5 ULOŽENÉ PROCEDURY	25
2.5.1 Programovací rozhraní	26
2.5.1.1 Konstrukce řízení toku.....	27
2.5.2 Zpracování chyb	28
2.6 PLÁNY ÚDRŽBY	29
2.7 ZABEZPEČENÍ SYSTÉMU SQL SERVER 2008	30
2.8 ZÁLOHOVÁNÍ DAT	31
Modely obnovy databáze.....	32
2.9 POŽADAVKY NA HARDWARE A SOFTWARE.....	33
2.10 NÁVRH APLIKACE.....	34
2.11 PLATFORMA APLIKACE PROGRAM MICROSOFT ACCESS.....	35
2.11.1 Psaní kódu ADO, VBA v Projektu(ADP).....	35
2.11.2 Obecné funkce VBA	35
2.11.3 Formuláře Projektu(ADP).....	37
2.11.3.1 Formulář a jeho oddíly.....	38
2.11.3.2 Ovládací prvky.....	38
2.11.4 Sestavy Projektu (ADP)	39
2.11.4.1 Řízení a seskupování dat v sestavách	39
2.11.5 Procedury a Moduly VBA.....	39
2.11.5.1 Moduly tříd	39
2.11.5.2 Standardní moduly.....	40

2.12	ZABEZPEČENÍ PROJEKTU(ADP).....	40
2.12.1	Uložení v podobě souboru (ADP).....	40
2.12.2	Možnost po spuštění.....	41
2.12.3	Uživatelská úroveň zabezpečení	41
II	PRAKTICKÁ ČÁST	42
3	ANALÝZA DATOVÝCH STRUKTUR.....	43
3.1	SYSTÉM NAVIGATOR	43
3.1.1	Základní pravidla importu dat do databáze Navigator	43
3.2	RELAČNÍ SCHÉMA DATABÁZE NAVIGATOR	45
3.3	DOCHÁZKOVÝ SYSTÉM WATT	46
3.3.1	Výstupní sestavy, export dat	46
3.3.2	Datové typy souboru Docházka.txt	47
3.3.3	Mzdová složka, datové typy	47
3.4	ERP SYSTÉM SAP	47
3.5	EXPORT DAT ZE SYSTÉMU SAP.....	48
3.6	SYSTÉM SAP, DATOVÉ TYPY	49
3.7	VYTVÁŘENÍ TABULEK V DATABÁZI SCA_SQL_2008.....	50
3.8	VYTVÁŘENÍ POHLEDŮ V DATABÁZI SCA_SQL_2008	52
3.9	VYTVÁŘENÍ ULOŽENÝCH PROCEDUR V DATABÁZI SCA_SQL_2008	53
3.10	VYTVOŘENÍ PLÁNU ÚDRŽBY DATABÁZE NAVIGATOR	56
3.11	VYTVOŘENÍ VÝSTRAH V DATABÁZI SCA_SQL_2008	57
3.12	FORMULÁŘE PROJEKTU APL_SCA(ADE)	58
3.12.1	Nevázané formuláře	58
3.12.2	Vázané formuláře	60
3.12.2.1	Funkce formuláře Reklamace	60
3.12.2.2	Ovládací prvek Spreadsheet.....	63
3.12.2.3	Ovládací prvek Indikátor průběhu	65
3.12.2.4	Ovládací prvek Kalendář	65
3.13	SESTAVY PROJEKTU APL_SCA.(ADP).....	66
3.14	MODULY PROJEKTU ADP_SCA(ADP)	69
3.15	SEZNAM MODULŮ PROJEKTU APL_SCA(ADP)	71
3.15.1	Plánování a řízení výroby	71
3.15.2	Výrobní kalkulace	71
3.15.3	Plánování spotřeby materiálu	72
3.15.4	Kapacitní bilance výroby.....	72
3.15.5	Dílenské řízení a evidence výroby	72
3.15.6	Dodavatelské a odběratelské reklamace.....	72
3.15.7	Docházka.....	73
3.15.8	Empirická data.....	73
3.15.9	Bezpečnost a dokumentace (revize)	73
4	IMPLEMENTACE PROJEKTU APL_SCA.ADP.....	74

5 ZÁVĚR	75
ZÁVĚR V ANGLIČTINĚ	77
SEZNAM POUŽITÉ LITERATURY	79
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	81
SEZNAM OBRÁZKŮ	83
SEZNAM TABULEK	84
SEZNAM PŘÍLOH	85

ÚVOD

Podnikový informační systém (ERP) patří mezi nejdůležitější nástroje pro úspěšné řízení podniku, ať řídíme malou či velkou firmu, oddělení nebo velkou nadnárodní korporaci zahrnující výrobní podniky lokalizované v různých státech. ERP systém SAP je informační systém pro střední a malé firmy, který zabezpečuje základní funkcionality finančního a manažerského účetnictví. V systému SAP evidujeme primární data fakturace, materiálové karty, výdeje a příjmy materiálů a výrobků ze skladu, které následně exportujeme do Podnikového informačního systému LEKY.

Diplomová práce si klade za cíl navrhnout Nadstavbový modul ERP systému pro výrobu a řízení pro společnost SCA Packaging s.r.o., která jej využije pro svou potřebu ve Zpracovatelské divizi zahrnující šest výrobních závodů v rámci celé České republiky.

Teoretická část obecně charakterizuje nejen požadavky na podnikový informační systém, ale také popisuje jednotlivé objekty a vlastnosti platformy Microsoft SQL Server 2008 a platformy aplikace Microsoft Access.

Praktická část představuje plně funkční Nadstavbový modul ERP systému pro výrobu a řízení, jehož cílem je sjednocení datových struktur systémů: ERP Systému SAP, Docházkového systému Watt, vytvoření nového výrobního modulu, který zajistí potřebný export dat tak, aby splňoval požadavky systému Navigator. Úkolem Nadstavbového modulu ERP systému pro výrobu a řízení je sloučit a sjednotit data v jednotlivých odděleních tak, aby byla navzájem provázána a rovněž, aby bylo zamezeno duplicitnímu opisování jednotlivých údajů, které poslopně navazují při samotném procesu výroby.

Systémové řešení, které nabízíme, se bude skládat z procedurální a neprocedurální části. Obě systémové části zastřeší odvětví výroby, plánování, objednávání i skladové hospodářství a nahradí známou problematiku s používáním tabulek Excelu, popřípadě starší nekompatibilní programy v zastaralé struktuře DOS.

Procedurální část (aplikace) - bude vyvíjena na modulárním systému platformy MS Access, která využívá v přenosu objektů metodiku OLE v programovacím jazyku Visual Basic Applications (VBA).

Neprocedurální část (databáze) bude vyvíjena v systému SQL Serveru 2008, který obsahuje výkonnou a pružnou bezpečnostní infrastrukturu projektu. Uvedený systém je

navržen tak, aby mohlo pracovat současně více uživatelů, aniž by se nějakým způsobem pracovní omezovali. Jedná se o přenos zabezpečený výměnou informací na úrovni změn databáze vzniklé od poslední výměny aktualizace.

I. TEORETICKÁ ČÁST

1 NÁVRH INFORMAČNÍHO SYSTÉMU

Správně navržený informační systém usnadňuje svým uživatelům orientaci ve velkém množství dat, která souvisejí s jejich každodenními činnostmi. Zmíněný systém tvoří množinu procesů složenou z datové a aplikační platformy, jež obsahuje technické a programovací prostředky sloužící pro sběr, uchování a vyhodnocování dat. Informační systém díky specializovaným modulům má možnost se přizpůsobit a plnit očekávané požadavky potřebných analýz, pro správu dat a pro zdárný průběh tvorby komplexních řešení u jednotlivých podnikových oddělení. Tento souhrn hardwaru a softwaru ve spolupráci se vstupními i výstupními zařízeními a samotnými systémovými daty je dnes nezbytným nástrojem k zajištění chodu dobře fungující firmy.

1.1 Návrh DATABÁZE

Návrh a tvorba databáze vyžaduje nejprve důkladnou přípravu dokumentace a získání potřebných informací pro důkladnou analýzu. Dobrý návrh databáze je základním předpokladem k vytvoření fungující databáze, která bude přesně, účelně a rychle plnit určené požadavky.

První fáze tvorby databáze neprobíhá přímo u počítače, ale za použití tužky a papíru při konstruktivních diskuzích jak se zadavateli, což je v našem případě úzký okruh lidí pod vedením divizního ředitele, tak i samotných budoucích uživatelů, kteří sdělí své představy a vnesou požadavky na funkčnost systému. Je nezbytné seznámit se s názvy polí, datovými typy, relačními vztahy a indexy systémů, které vzájemně propojujeme – z nich totiž čerpáme primární data pro datový sklad informačního systému.

Návrh databáze se skládá:

- Určení účelu databáze
- Vyhledání a uspořádání požadovaných informací
- Rozdělení informací do tabulek
- Převod jednotlivých informací do sloupců
- Zadání primárních klíčů
- Vytvoření relací mezi tabulkami
- Použití normalizačních pravidel

2 PLATFORMA MICROSOFT SQL SERVER 2008

Pro umožnění a potřeby univerzálního přístupu k datům z různých serverů a z různých zařízení do jednotné databázové struktury je nezbytná výkonná serverovou a datovou infrastrukturu. Proto jsme zvolili platformu SQL Server 2008, který byl navržen tak, aby umožňoval kdykoliv spravovat libovolné údaje z jakéhokoliv místa za , že přenosová rychlosti internetového připojení bude dostačující.

Microsoft SQL Server je relační databázový a analytický systém pro řešení datových skladů¹ s podporou nástroje SQL Server Management Studio, a tím tvoří vysoce výkonný nástroj pro správu velkého množství dat. Relační databáze je definována jako soubor nástrojů pro efektivní a spolehlivé ukládání údajů a pro využívání těchto údajů při správné manipulaci s nimi. Obsahuje relační model, který se skládá z datových prvků, entit aplikace a samozřejmě z instancemi navzájem propojených relací tvořících vnitřní strukturu databáze. Pod označením SQL Server se neskrývá pouze databáze, ale komplexní výkonná a spolehlivá serverová platforma pro ukládání a správu údajů. SQL Server je dodáván v šesti edicích, takže nabízí možnost zvolit si vhodnou verzi s ohledem na požadavky aplikace.[4]

„SQL Server 2008 Enterprise Edition

SQL Server 2008 Enterprise Edition je ucelená platforma splňující vysoké nároky podnikových aplikací pro online zpracování transakcí a datové sklady.

SQL Server 2008 Standard Edition

SQL Server 2008 Standard Edition je platforma pro správu dat s jednoduchou zpracovatelností určená primárně pro provoz aplikací firemních oddělení.

SQL Server 2008 Workgroup Edition

Tato verze je určena především pro firemní pobočky. Poskytuje funkce pro zabezpečenou vzdálenou synchronizační správu.

¹ Datový sklad je specifickým typem relační databáze, který řeší analytické úlohy dotazování nad velkým množstvím dat pomocí logických struktur(dimenzí, schémat a faktorových tabulek).

SQL Server 2008 Web Edition

Nabízí nízké náklady, vysokou škálovatelnost pro vysoce dostupné webové aplikace nebo hostovaná řešení a dostupnost internetových prostředí webových služeb.

SQL Server 2008 Compact Edition

Edice Compact je k dispozici bezplatně ke stažení a umožňuje vytvářet samostatné a příležitosti připojené aplikace pro mobilní zařízení.

SQL Server 2008 Express Edition

Edice Express je poskytována bezplatně ke stažení a je ideální pro studium a vytváření aplikací pro klientské počítače a malé servery a pro distribuci nezávislými výrobci softwaru.²

2.1 SQL Server 2008 Standard Edition

Po důkladném zvážení všech možností výběru edicí SQL serveru v závislosti na potřebných požadavcích aplikace jsme se rozhodli pro edici SQL Server 2008 Standard. SQL Server 2008 Standard je relační databázový systém, který je plnohodnotným nástrojem pro správu dat s jednoduchou zpracovatelností. Především se výhodně používá ve firemních aplikacích, kde jsou kladeny požadavky na velký výkon a rychlost zpracování složitých dotazů. Hlavní předností SQL Server 2008 Standard je, že interaktivně vizualizuje uživatelské rozhraní. Tím tvoří přehledné a srozumitelné prostředí pro správu dat. SQL Server 2008 Standard obsahuje výkonné funkce Reporting Services, serverovou platformu pro vytváření a distribuci standardních integrovaných sestav a grafických nástrojů.

Dále SQL Server 2008 Standard poskytuje specifické nástroje, které zajišťují dostupnost a ochranu dat, mezi ně patří instance havarijního clusteringu systému SQL Serveru, které

² LACKO, Euboslav. *Jak vyrát na Microsoft SQL Server 2008: správa, konfigurace, programování*. Vyd. 1. Brno: Computer Press, 2009, s. 21-22. ISBN 978-80-251-2101-6.

jsou vytvořeny na základě služby Windows Clustering Service³. Při nasazení havarijního clusteringu chráníme celou instanci před selháním hardwaru, tím dojde k datovému zrcadlení na úrovni instancí, což vyžaduje podsystém sdílených disků.

Dalším nástrojem, který budeme využívat je zrcadlení databáze, která se spoléhá na interní procesy správy protokolu v rámci modulu úložiště. Nastavením mimořádně nízkého zpoždění se udržuje druhá kopie databáze. Zrcadlení databáze může fungovat i v synchronním režimu, proto nemůže chyba primární databáze nikdy způsobit ztrátu transakce. Pro využití všech možností zrcadlení databáze je zapotřebí tři serverů.

Základem samotného předávání protokolů je modul zálohování a obnovení ve spolupráci s modulem SQL Server Agent, který plánuje automatickou aplikaci záloh transakčního protokolu na sekundární server. Klíčovou vlastností modulu SQL Server Agent je schopnost vytvářet pouze úlohy, které mohou mít více kroků a závislostí mezi kroky úloh a lze je spouštět podle různých plánů. Na základě a pro potřeby různých podmínek lze nakonfigurovat výstrahy, ty mohou buď spouštět automatické akce, nebo odesílat zprávy do protokolů chyb, případně konfigurovaným operátorům.[4]

2.1.1 Databázový modul SQL Server 2008

Nejdůležitějším prvkem SQL Serveru 2008 je databázový modul (Databáze Engine), který zjišťuje ukládání, načítání, zpracovávání a zabezpečení dat. Služby tohoto databázového modulu umožňují vybudovat vysoce výkonné databázové aplikace pro zpracování OLTP (online transaction processing) a podporu zpracování OLAP (online analytic processing). Databáze jsou objekty, se kterými pracuje relační modul i služby SSAS⁴. Ten tvoří výkonnou jednotku pro správu dat tak, že spolupracuje s modulem úložiště (storage engine), což je jádrem systému SQL Server, který řídí ukládání dat na disk a zpřístupňuje tato data aplikacím. Modul úložiště je interní komponentou, se kterou se nedostaneme do

³ Windows Clustering Service poskytuje automatické hardwarové funkce předání služeb při selhání. Snaží se minimalizovat důsledky výpadků systému.

⁴ SSAS (SQL Server 2008 Analysis Service) - tato služba zahrnuje skupinu analytických prvků k získávání informací pro technologické uložení dat v databázi.

přímého kontaktu. Obsahuje však dílčí komponenty, jež jsou klíčové pro ukládání a správu uživatelských dat. Systém SQL Server 2008 zahrnuje sadu systémových databází sloužících pro správu databázového modulu, mezi něž náleží zabudované moduly Master, Model, Msdb, Tempdb, Distribution, bez kterých by Systém SQL Server 2008 nebyl plnohodnotným nástrojem pro správu dat.

Struktura databází systému SQL Server je dána pomocí jednoho fyzického prvku a jednoho logického prvku, který zajišťuje datové seskupení v rámci celé databáze. Datové soubory a soubory transakčního protokolu mají přípony. Přípona souboru .mdf deklaruje primární soubor systému, přípona souboru .ndf vyjadřuje sekundární soubor a přípona souboru .ldf označuje, že se jedná o soubor transakčního protokolu.

2.1.1.1 Relaçní modul

Nejdůležitější částí databázového modulu SQL Serveru je relační databázový modul, který sdružuje data v tabulkách. Pokud jsou v různých tabulkách sloupce stejného typu, pak tyto sloupce mohou vytvářet relace mezi jednotlivými tabulkami. Tabulky se poté naplňují vlastním obsahem - konkrétními daty. Kolekce většího počtu tabulek, jejich funkčních vztahů, indexů a dalších součástí tvoří relační databázi. Relační modul přináší celou řadu výhod, zejména mnohdy přirozeně reprezentuje zpracovává data, dává možnost snadno definovat a zpracovávat vazby. Samotné vazby se definují pomocí databázových diagramů.

2.1.1.2 Databázové diagramy

Databázové diagramy poskytují základní vizualizaci databáze. Při práci s nimi musíme být velmi opatrní, protože databázový diagram je propojen s databází na pozadí. Databázové diagramy umožňují provádět změny relací mezi jednotlivými tabulkami v databázi. Jestliže v diagramu úložiště neupravíme pouze toto schéma, ale i vlastní databázi, může být narušena referenční integrita a poškozena funkčnost databáze. Při návrhu databáze je vhodné používat nástroje datového modelování, jako je např. Visio for Enterprise Architects.

2.2 SQL Server Management Studio

Abychom mohli účelně a efektivně spravovat systém SQL Server, využijeme nástroj pro psaní dotazů, funkcí a uložených procedur. K tomu slouží aplikace SQL Server Management Studio, které je součástí instalačního balíčku MS SQL Server a vypadá obdobně jako Visual Studio⁵. Aplikace SQL Server Management Studio je napsaná v programovacím jazyce Visual C#, který je založen na vizualizaci funkcionalit Visual Studia a usnadňuje svými nástroji psaní kódu. Tuto aplikaci využijeme pro zadávání a ladění příkazů jazyka SQL. Tyto příkazy pak implementujeme do logiky aplikace a databázové struktury datového skladu, když vytváříme procesy instance a konfigurace systému. Kromě klientské konzolové aplikace potřebujeme také nástroje pro správu databáze. Prostřednictvím SQL Server Management Studio můžeme vytvářet nové databáze, tvořit správu a volat účty jednotlivých uživatelů, přidělovat a rušit jejich oprávnění. SQL Server Management Studio je integrované prostředí pro správu databázového serveru SQL Server 2008, kde jeho součástí je i prostředí pro zadávání a ladění SQL příkazů.

Samotný databázový server je koncipován tak, že veškeré nástroje pro správu dat běží na pozadí. Tyto nástroje využívají middlewarové systémy, webové nebo intranetové servery a některé aplikace na klientských počítačích typu klient-server⁶. Proto je nutné s databázovým serverem pracovat na administrátorské úrovni, abychom mohli spravovat systémové nastavení, a navíc vyvíjet nové prvky. [2]

⁵ Visual Studio je vývojové prostředí pro vytváření konzolových nebo objektově orientovaných aplikací

⁶ Klient server je síťová architektura, která odděluje klienta aplikací prostřednictvím grafického rozhraní serveru. Jde o vztah mezi dvěma počítačovými programy (vztah mezi aplikační a datovou částí).

2.3 Návrh tabulek

Nyní již máme definovanou platformu SQL Server 2008 a taktéž nástroj pro správu dat SQL Server Management Studio, a tedy můžeme implementovat potřebné objekty. Koncepce tabulek je jedním z nejdůležitějších kroků při navrhování celé databáze. V okamžiku, kdy jsou již navrženy tabulky, musí být vytvořen základ struktury celé databáze, který lze v budoucnu obměňovat pouze v detailech. Změnit strukturu databáze, která je již naplněna daty a má již vytvořeny relační vztahy, je velmi komplikované a zdlouhavé.

Každá informace by měla být v databázi obsažena pouze jednou. Je třeba zabránit zbytečné duplicitě informací, jak v rámci celé databáze, tak i uvnitř samotné tabulky. Obsah tabulek musí mít předmětově orientovaný charakter. Při návrhu tabulky je potřebné respektovat rozsah dat, která v ní budou nadále obsažena. Každá tabulka by měla být co nejvíce strukturovaná, aby se informace ukládaly v menších celcích, a tím vzrostla možnost filtrovat a analyzovat data pomocí dotazů.

Databázové objekty je třeba vždy identifikovat. Proto má s výjimkou omezení každá databázová tabulka svůj identifikátor, který je objektu přiřazen v průběhu jeho vytváření. Pro identifikátory platí předem definovaná pravidla, která nesmíme při vytváření tabulek ignorovat. Mohou obsahovat nejvýše 128 znaků, prvním znakem musí být písmeno. Identifikátor objektu nesmí obsahovat speciální znaky a nesmí být identický s vyhrazeným slovem jazyka T-SQL.[2]

2.3.1 Určení datových typů polí tabulky

Návrh vhodných datových typů polí databázové tabulky musí být jedním z hlavních úkolů vývoje a implementace Nadstavbového modulu ERP systému pro výrobu a řízení. Sjednocuje totiž data ERP systému SAP s Docházkovým systémem Watt a zajistí potřebný export dat tak, aby splňoval požadavky systému Navigator. Datové typy polí musí odpovídat charakteru dat, která budeme do příslušného pole ukládat. Je proto zcela zbytečné definovat textová pole pro větší počet znaků. Takové tabulky zabírají zbytečně velké množství místa na disku, a trpí tím i celkový výkon databáze. Datový typ je nejdůležitější vlastností pole tabulky. Je ho nutno přesně specifikovat, kvůli efektivnímu ukládání dat a pro rychlou práci s těmito daty. V návrhu tabulek využijeme následné datové typy: Číselná data, Znaková data, Binární data, Časová data.

2.3.1.1 Číselná data

Mezi základní datové typy v systému SQL Server patří číselné datové typy. Nejpoužívanějším datovým typem pro celá čísla je int. V případě, že nám z nějakého důvodu nestačí datový typ Int, pak systém SQL Server 2008 standardně podporuje deset číselných typů, které obsáhnou veškerý rozsah číselných hodnot.[1]

Tabulka č. 1: Popis číselných datových typů[11]

Název	Rozsah	Velikost
Int	-2 147 483 648 až 2 147 483 647 (-231 až 231-1)	4 B
Bigint	-9 223 372 036 854 775 808 až 9 223 372 036 854 775 807 (-263 až 263-1)	8 B
Smallint	-32 768 až 32 767 (-215 až 215-1)	2 B
Tinyint	0 až 255	1 B
Bit	0, 1 nebo null	1 B
Decima/numeric	-1038+1 až 1038-1	17 B
Money	-922 337 203 685 477,5808 až 922 337 203 685 477,5807	8 B
Smallmoney	-214 748,3648 až 214 748,3647	4 B
Float	-1,79E+308 až -2,23E-308, 0 a 2,23E-308 až 1,79E+308	8 B
Real	-3,40E+38 až -1,18E-38, 0 a 1,18E-38 až 3,40E+38	4 B

2.3.1.2 Znaková data

Znakové datové typy spolu s číselnými datovými typy zpravidla představují více než 90 procent všech datových typů, které jsou v databázích definovány. Základní znakové datové typy se dělí na znaková data s pevnou a proměnnou délkou, data v kódování Unicode a množiny znakových dat Varchar(max) nebo Nvarchar(max). Znakový datový typ text je podle dokumentace považován za zastaralý, nyní se místo něj používá znakový datový typ varchar(max).[4]

Tabulka č. 2: Popis znakových datových typů[11]

Název	Rozsah	Velikost
Char	1 až 8 000 znaků	n B
Varchar	1 až 8 000 znaků	m + 2 B
varchar(max)	1 až 231-1 B	m + 2 B

2.3.1.3 Datová a časová data

System SQL Server poskytuje šest základních typů k ukládání data a času. Zachycují okamžik v toku času s předdefinovanou přesností. SQL Server 2008 aktualizoval standardní datový typ datetime o čtyři nové datové typy: datetimeoffset, datetime2, date a time.

Tabulka č. 3: Popis data a časových datových typů[11]

Název	Rozsah	Velikost
Date	0001-01-01 až 9999-12-31 (YYYY-MM-DD)	3 B
Datetime	1753-01-01 až 9999-12-31 (YYYY-MM-DD hh:mm:ss[.nnn])	8 B
Datetime2	0001-01-01 00:00:00.0000000 až 9999-12-31 23:59:59.9999999 (YYYY-MM-DD hh:mm:ss[.nnnnnnn])	6-8 B
Datetimeoffset	0001-01-01 00:00:00.0000000 až 9999-12-31 23:59:59.9999999 (UTC) (YYYY-MM-DD hh:mm:ss[.nnnnnnn] [+ -]hh:mm)	8-10 B
Smalldatetime	1900-01-01 až 2079-06-06 (YYYY-MM-DD hh:mm:ss)	4 B
Time	00:00:00.0000000 až 23:59:59.9999999 (hh:mm:ss[.nnnnnnn])	3-5 B

2.3.1.4 Binární data

Binární data jsou obdobná datovým polím pro znakové řetězce. Deklarují se s parametrem, který určuje rozsah informací. Datové typy pro ně jsou tři, přičemž byl datový typ image označen za zastaralý.

Tabulka č. 4: Popis binárních datových typů[11]

Název	Rozsah	Velikost
Binary	1 až 8 000	n B
Varbinary	1 až 8 000	m + 2 B
varbinary(max)	1 až $2^{31}-1$	m + 2 B
Binary	1 až 8 000	n B
Varbinary	1 až 8 000	m + 2 B

2.3.2 Dělení tabulek a indexů do oddílů

Pokud již máme v tabulkách definované datové typy, můžeme přejít na dělení tabulek a indexů do oddílů. Schopnost rozdělit data do menších skupin je pro zlepšení výkonu běžný požadavek, zejména když nakládáme z více a většími sadami dat SQL Server Management Studio dokáže využívat integrovaný nástroj pro rozdělení už existující tabulky na menší oddíly. Pro požadované rozdělení tabulek slouží dělicí funkce a oddílová schémata využívající generátor náhodných čísel, který je zabudovaný do jádra SQL Serveru. Úlohou oddílového schématu je definovat místo, kde budou údaje z jednotlivých oddílů uloženy.[3]

2.3.2.1 Základní dělení tabulek indexů do oddílů:

Indexování polí tabulek

Indexování polí tabulek je prostředkem, který umožňuje rychleji vyhledávat a řadit záznamy podle indexovaných polí, a tedy vést výkonnější správu a používání jednotlivých tabulek i databáze jako celku. Funkčnost indexu je využitelná pouze tehdy, pokud lze s jeho pomocí najít data velmi rychle, a to bez ohledu na značnou rozsáhlost jednotlivých uložených dat. Vytvoření indexů klade vyšší nároky databázového serveru na operační paměť a na diskový prostor, proto je nutné využívat indexování tabulkových polí co nejefektivněji.

Vyvážené stromy (B-stromy)

SQL Server vytváří a spravuje indexy na principu struktury, jedná se o široce využívanou názornou stromovou strukturu s připojenými uzly nazývanými se „vyvážený strom“ (B-strom). B-strom vychází z kořenového uzlu, který obsahuje jedinou datovou strukturu, nuly nebo více mezilehlých úrovní a listové úrovně. Uzel B-stromu může obsahovat hodnotu, podmínku, ale taky strukturu oddělených dat nebo může obsahovat vlastní strom.

Základním principem je vyváženost a symetrie. V levé i pravé polovině B-stromu se proto vždy nachází nalézá stejný počet stran, což platí pro všechny úrovně B-stromu. Stránky listové úrovně obsahují položky, které jsou seřazeny v definovaném pořadí. Data na listové úrovni mají v sobě obsaženy všechny kombinace hodnot v rámci indexovaných sloupců. Počet řádků indexu na stránce je závislý na tom, jaké úložné místo vyžadují sloupce definované v indexu.

Úrovně indexů

Na jednoduchém matematickém vztahu závisí jak počet stránek indexu, tak i počet stránek na každé úrovni indexu, kdy datová stránka v systému SQL Serveru má velikost 8,192 bajtů a dovoluje uložit až 8060 bajtů vlastních uživatelských dat. S ohledem na počet bajtů potřebných k uložení klíče indexu, což se odvíjí od datového typu, můžeme jednoduchým dělením vyčíslit počet řádků uložených na stránce.

Clusterované indexy

Clusterované indexy fyzicky uspořádávají data v tabulce podle clusterovaného klíče. Index lze definovat pomocí jednoho nebo více sloupců v tabulce. V tabulce je možné definovat index v rozsahu do 16-ti sloupců, maximální velikost klíče je 900 bajtů. Sloupec nebo sloupce definované pro clusterovaný index se označují jako clusterovaný klíč. Vzhledem k tomu, že tabulku nelze seřadit více způsoby, je možné v jedné tabulce definovat jen jediný clusterovaný index.

Neclusterované indexy

Dalším typem relačního indexu, který lze vytvořit, je neclusterovaný index. Obdobně, jako vyhledáváme odkazy v rejstříku knihy, pracují neclusterované indexy. Neclusterované indexy nevynucují určité řazení tabulky, takže jich v jedné tabulce můžeme použít více. Neclusterované indexy vykazují stejná omezení jako clusterované indexy, kdy mohou obsahovat maximálně 900 bajtů a nejvíce 16 sloupců. Kromě toho zde platí, že v tabulce nemůže existovat více než 249 neclusterovaných indexů.

Pokrývající indexy

Pokrývající indexy mohou mít u některých dotazů specifický efekt. Pokud v odpovědi na dotaz stačí vrátit data pouze ze sloupců v indexu, nemusí systém SQL Server vůbec přistupovat k datovým stránkám ve vlastní tabulce. Systém SQL Server totiž vyhledá všechna požadovaná data již při předchozím indexu. Pokud dokáže vytvořit pokrývající indexy pro data, ke kterým uživatelé často přistupují, může urychlit odpovídání na dotazy, protože odpadnou dodatečná čtení tabulky na pozadí.

Filtrované indexy

Filtrované dotazy podléhají omezením, kdy hodnoty sloupců nemohou procházet implicitní nebo explicitní konverzí datového typu, musí se jednat o neclusterovaný index a nelze je

vytvořit pro výpočtová pole. Systém SQL Server při vytvoření indexu generuje strukturu pomocí histogramu, který uchovává informace o relativním rozložení datových hodnot ve sloupci. Míra, se kterou konkrétní hodnoty ve sloupci dovolují vyhledávat malé množiny dat, se označuje jako selektivita indexu.

Vytvoření indexu online

V průběhu vytvoření indexu je nezbytné načíst všechny hodnoty v klíči indexu a následně je použít při generování indexu. Postup načítání všech hodnot a výstavby indexu zabere dobu, jejíž délka závisí na množství dat v tabulce. Proto není vyloučeno, že se data v klíči indexu v této době změní. Systém SQL Server kontroluje změny dat v tabulce, aby byla při vytváření indexu zajištěna konzistence dat se zřetelem na vybrané možnosti. Indexy lze vytvářet online nebo offline. SQL Server uzamkne celou tabulku a zabrání do vytvoření indexu provést jakékoli změny, přičemž využívá úložiště verzí v rámci databáze Tempdb.[1],[4]

2.4 Pohledy(Views)

Další způsob získání dat z databázové tabulky lze uskutečnit pomocí databázového objektu pohled , který poskytuje uživateli data ve stejné podobě jako tabulka. Pohled definuje příkaz SELECT, který je označen názvem a uložen do databáze. Na rozdíl od tabulky pohled nezabírá v paměti téměř žádné místo, protože obsahuje pouze předpis pro získání dat, nikoliv data samotná. Hlavní výhoda pohledu spočívá v tom, že po vytvoření funguje jako tabulka, kterou můžeme využívat pro další libovolné příkazy SELECT.

Pohled může zahrnovat více příkazů SELECT za předpokladu, že využijeme operátory UNION nebo UNION ALL⁷.

Při vytvoření standardního pohledu uloží systém SQL Server pouze jeho definici, kterou pak optimalizátor dotazu nahradí v příkazech SELECT, které s pohledem pracují. Index se vytváří na základě seznamu hodnot ve sloupci. Indexujeme-li pohled, systém SQL Server spouští příkaz SELECT, definovaný pohled uloží sadu výsledků a poté sestaví index.

⁷ UNION je syntaktická konstrukce jazyka SQL, která slouží ke sjednocení výsledku dotazu ze dvou nebo více vstupních množin dotazu SELECT.

Všechny následné příkazy DML odkazují na libovolnou z tabulek, podle kterých je pohled definován, tím způsobí, že systém SQL Server přírůstkově aktualizuje uloženou sadu výsledků a v případě potřeby je samozřejmě schopen zajistit i údržbu indexů. Vytvoříme-li na základě pohledu index, data se tím materializují. Dotazy, které odkazují na indexovaný pohled, sice nenahrazují definici pohledu, ale místo toho vracejí výsledky dotazu přímo z indexovaného pohledu. Výsledky lze vrátit přímo, protože z hlediska úložiště je indexovaný pohled v praxi uložen jako tabulka spravovaná modulem úložiště.[4]

2.5 Uložené procedury

V tomto okamžiku již máme definovány objekty, s nimiž budeme pracovat pomocí jazyka T-SQL v prostředí objektu Uložené procedury. Uložené procedury se vykonávají na pozadí v rámci databázového rozhraní. Každý příkaz spouštěný v systému SQL Server můžeme zapouzdřit do uložené procedury, a tím umístit kromě vnořených údajů i část aplikační logiky, případně její celek. Uložená procedura je proces, který je funkčně oddělený od programového okolí. Komunikuje i s jinými moduly programu pomocí seznamu parametrů. Uložená procedura je v podstatě dávka příkazů jazyka T-SQL, která je označena názvem a uložena do databáze. Uložené procedury můžeme vytvořit pomocí příkazu CREATE PROCEDURE.

Syntaxe kódu jazyka T-SQL v uložené proceduře obsahuje mimo příkazů jazyka T-SQL, taktéž proměnné. Pomocí proměnné je možné manipulovat s daty, můžeme tedy data ukládat, třídít, deklarovat v rámci uložených procedur. Systém SQL Server nabízí dva různé typy proměnných: místní proměnné a globální proměnné. Místní proměnné (global variable), identifikovatelné pomocí dvojitého symbolu @@, lze taky vytvářet, číst a zapisovat do nich na rozdíl od globální proměnné, kterou není možné vytvářet nebo měnit.

I když instanci proměnné můžeme vytvářet pomocí klauzule DECLARE, ve které uvedeme název a datový typ proměnné, můžeme definovat proměnnou libovolného charakteru datového typu s výjimkou typů text, ntext, a image.

Samotnou instanci více proměnných sice lze vytvářet pomocí jednoho jediného příkazu DECLARE, ale instance tabulkové proměnné vyžaduje nutnost příkazu DECLARE. U proměnné můžeme používat statickou hodnotu nebo jedinou hodnotu vrácenou příkazem SELECT nebo SET.

Parametry rozumíme místní proměnné, které mají úkoly pro předání hodnot uložené procedury, jakmile ji spustíme. Při spuštění se všechny parametry chovají stejně jako proměnné a lze je číst, zapisovat nebo editovat. Parametry dělíme na dva základní typy, což jsou vstupní a výstupní parametry. Výstupní parametr je označen klíčovým slovem OUTPUT.

Návratový kód nás informuje o stavu spouštění procedury zpět do aplikace. Účelem návratového kódu není odesílat data, ale slouží jako zdroj informací o tom, jaké výsledky ukazuje probíhající procedura. Převážně je jeho hodnota datového typu Integer.[12],[2]

2.5.1 Programovací rozhraní

Programovací jazyk Transact-SQL (TSQL) je snadno použitelný a výkonný nástroj pro údržbu a vývoj nových prvků. Jestliže syntaxe T-SQL nesplňuje všechny naše požadavky, můžeme možnosti serveru rozšířit o libovolný programovací jazyk s modulem CLR (common language runtime), jako jsou Microsoft Visual Basic nebo Microsoft Visual C#.

Přímo do modulu jsou integrovány funkce jazyka XML (Extensible Markup Language), které dovolují ukládat data XML a dotazovat se na ně, a také vracet sady výsledků v mnoha různých formách XML. Kód lze modularizovat, uložit na server a přistupovat k němu z aplikace pomocí pohledů, uložených procedur, funkcí a spouští. Pomocí fulltextového vyhledávání můžeme vytvářet rozhraní dotazů nad velkými objemy dat.

„Hlavním omezením jazyka SQL je skutečnost, že se jedná o neprocedurální jazyk. V praxi to znamená, že příkazy jazyka SQL se vykonávají sekvenčně, bez možnosti použití klasických programátorských konstrukcí, jako jsou například cykly, podmínky, využití procedur a funkcí.

Proto má téměř každá moderní databázová platforma implementované určité procedurální rozšíření jazyka SQL. Takto rozšířený jazyk SQL se potom stává mocným nástrojem, který umožňuje naprogramovat i ty nejsložitější algoritmy pro práci s údaji, případně pro jejich zpracování.

*Rozšíření jazyka SQL na platformě Microsoft SQL Serveru má název Transact SQL nebo zkráceně T-SQL. Pro tvorbu uložených procedur a triggerů budeme potřebovat alespoň základní přehled o tomto jazyce.*⁸

2.5.1.1 Konstrukce řízení toku

Při práci s daty v různých objektech SQL Server je potřebné využít Konstrukce řízeného toku dat. Mnohdy musíme volit mezi více akcemi podle okamžitých okolností. K těmto účelům slouží konstrukce RETURN; IF...ELSE; BEGIN...END; WHILE; BREAK/CONTINUE; WAITFOR; GOTO.

Příkaz RETURN deklaruje hodnotu funkce jako hodnotu jeho výrazu, umožňuje ukončit řízení i procedury a vrací řízení zpět volající aplikaci. Po příkazu RETURN se již žádné následující příkazy nespustí. Tento příkaz musí být vždy v těle samotné funkce.

Při provádění příkazu IF se povoluje podmíněně spouštět části kódu, který je uložen v konstrukci IF...ELSE. Příkaz IF zkontroluje uvedenou podmínku a spustí následující blok kódu, samozřejmě v případě, že podmínka je pravdivá. Není-li splněna žádná podmínka, provedou se příkazy za ELSE. Volitelný příkaz ELSE umožňuje spustit kód, je-li podmínka vyhodnocena jako nepravdivá.

Při spuštění příkazu WHILE se opakovaně vyhodnocuje podmínka, která je ve větvi příkazu obsažená. Provádění příkazu skončí v případě, jestliže podmínka bude mít hodnotu TRUE.

Mezi největší chyby při psaní bloku kódu s příkazy IF nebo WHILE patří opomenutí faktu, že systém SQL Server podmíněně spouští jen následující příkaz. Chceme-li eliminovat nejčastější programátorské chyby, měli bychom v rámci příkazu IF nebo WHILE vždy zadat konstrukci BEGIN...END i pokud potřebujeme podmíněně spustit pouze jediný řádek kódu.

⁸ LACKO, Euboslav. *Jak vyžrát na Microsoft SQL Server 2008: správa, konfigurace, programování*. Vyd. 1. Brno: Computer Press, 2009, s. 93. ISBN 978-80-251-2101-6.

Příkaz CASE se využívá vybíráme-li jednu z více možných posloupností příkazů. Samotné posloupnosti jsou deklarovány ve větvích příkazu a začínají příkazem WHEN nebo ELSE. V příkazu CASE hledáme hodnotu výrazu po řadě ve větvích mezi hodnotami výrazů. V případě, že nalezneme větev příkazu, spustí se příkazy, které se v této větvi nacházejí.

Příkaz BREAK se používá spolu se smyčkou WHILE. Pokud chceme ukončit provádění v rámci smyčky WHILE, můžeme přerušit iteraci smyčky příkazem BREAK. Po spuštění příkazu BREAK bude provádění kódu pokračovat dalším řádkem kódu za smyčkou WHILE. Příkaz CONTINUE uvnitř smyčky WHILE zajistí pokračování běhu v rámci smyčky.

Pomocí příkazu WAITFOR je možné pozastavit provádění kódu. Příkaz WAITFOR má tři různé varianty a to: WAITFOR DELAY, WAITFOR TIME a WAITFOR RECEIVE. Příkaz WAITFOR TIME pozastaví spuštění kódu do uvedeného času.

Příkaz GOTO umožňuje pokračovat v realizaci kódu od jmenovky, která je uvedena na jiném místě procedury. Konstrukce kódu, jako je GOTO, se však nedoporučuje v žádném programovacím jazyce.[4]

2.5.2 Zpracování chyb

Pokud jsme již seznámeni s konstrukcí řízení toku dat v uložených procedurách, nesmíme opomenout na integraci zpracování chyb. Dřívější SQL Servery zpracovávaly chyby při testování globální proměnné @@error. Nyní můžeme použít strukturované zpracování chyb srovnatelné s ostatními programovacími jazyky. Přitom se využívá blok TRY...CATCH.

Příkaz RAISEERROR představuje další využitelnou možnost, jak získat hlášení o chybách a výjimkách u uložených procedur.

„Zjednodušená syntaxe příkazu je RAISEERR je:

```
RAISERROR ( {msg_id | msg_str } { , uroven_zavaznosti , stav }  
[ , argument [ , ...n ] ] )  
[ WITH option [ ,...n ] ]
```

Úroveň závažnosti od 1 do 15 nevytváří hlášení do systémového protokolu. Úroveň 16 zapíše varování a úrovně 17 až 25 zapíše do systémového protokolu chybu. Úrovně od 1 do 18 může zadávat libovolný uživatel, vyšší úrovně pouze administrátor.“⁹

2.6 Plány údržby

Modul Plán údržby představuje jeden z nejdůležitějších nástrojů zabezpečujících plynulost a funkčnost dat v databázích. Obsahuje pracovní postup úkolů deklarovaný v jazyku T-SQL. V tomto jazyce může být plánován pomocí rozvrhu nebo může být spuštěn na základě předefinovaných požadavků. Plán údržby stojí na bázi stroje SSIS¹⁰, čímž je nám dán jednoduchý způsob vytváření pracovního postupu v rámci všech plánů; umožňuje přidávání vlastních úkolů T-SQL v případě, že uživatelské rozhraní neposkytuje v čase tvorby požadovanou funkčnost nebo sdělování uživatelům, jak bude spouštěný příkaz T-SQL vypadat. SQL Server Management Studio dává možnost vytvářet plány údržby pomocí Maintenance Plan Designer.

Plány údržby pomocí Maintenance Plan Designer, kdy existuje mechanismus grafického provedení umožňují zpracovat níže uvedené úlohy:

- Zálohování databáze a transakčních protokolů
- Zmenšování databáze
- Opakované indexování
- Kontroly konzistence
- Aktualizace statistik

⁹ LACKO, Luboslav. Jak vyžrát na Microsoft SQL Server 2008: správa, konfigurace, programování. Vyd. 1. Brno: Computer Press, 2009, s. 98. ISBN 978-80-251-2101-6.

¹⁰ SSIS je platforma pro integraci dat vybavená rychlou a flexibilní infrastrukturou zabezpečující extrakt, transformaci a načtení dat.

2.7 Zabezpečení systému SQL Server 2008

Nesmíme opomenout bezpečnostní infrastrukturu, která zajišťuje ochranu dat a instancí před neoprávněným přístupem. Systém SQL Server může řídit povolený způsob ověřování uživatelů a vynutit, aby se používala buď pouze pověření systému Windows nebo interní účty systému SQL Server. Lze povolit či zakázat různé funkce modulu SQL Serveru, a tímto způsobem zaručit, že bude disponováno jen podmnožinou funkcí, která je nutná pro činnost aplikace. Zabezpečení serveru a údajů lze rozdělit do tří základních skupin, jimiž jsou: správa uživatelských práv, transparentní šifrování a auditování.

Zabezpečení má za úkol vytvořit dostatečné překážky proti napadení systému nebo proti poškození dat. V případě, že budeme chtít bránit náš systém proti různým metodám útoku, musíme zvolit víceúrovňový systém obrany. Hlubkový systém obrany, který před útočníka klade více překážek, představuje povolený režim ověřování, konfigurace síťových prvků tak, aby bylo povoleno vzdálené připojení. Konfigurace vzdáleného připojení a režimu ověřování představuje základní úroveň bezpečnosti. Další možnosti jsou realizovány pomocí vestavených funkcí, které zabrání útočníkovi vniknout do systému. Systémová funkce v rámci instance lze aktivovat tak, že spustíme uloženou proceduru `sp_configure`.

Velmi účinným nástrojem pro zabezpečení důvěrných dat je šifrování. Šifrovaná data ve sloupcích nelze číst bez příslušného pověření. Pokud jsou data ve sloupcích zašifrována, není možné zašifrovaný sloupec uvádět v argumentech hledání. Sloupce lze šifrovat pomocí hesla, symetrického nebo asymetrického klíče. Z uvedených způsobů šifrování právě symetrické klíče nejméně zatěžují výkon procesoru a dostačující pro šifrování dat.

Funkce SQL Serveru umožňují auditovat přístup do databáze. Jedná se o dotazování a rovněž o modifikaci údajů. Tento nástroj dokumentuje, které údaje byly čteny nebo modifikovány, a nazývá se objekt Audit. Výstupem auditu je možnost směřovat data do souboru nebo protokolu serverových událostí. Samotné výsledky můžeme zpracovat pomocí analytických funkcí nebo zobrazit výsledky pomocí reportu.

„Aby se zajistilo, že systém SQL Server bude ve výchozím nastavení zabezpečený tak, jak nejvíc to bude možné, je ve výchozím nastavení zakázáno množství funkcí, které představují bezpečnostní riziko, a tyto funkce musí být před svým použitím explicitně povoleny. To zahrnuje následující funkce:

- *Vzdálená připojení*

- *Vyhrazená linka správce*
- *Rozhraní .NET Framework*
- *Databázová pošta*
- *Funkce SQLMail*
- *Služba Service Broker*
- *Připojení pomocí protokolu HTTP*
- *Zrcadlení databáze*
- *Služba Web Assistant*
- *Rozšířená procedura xp_cmdshell*
- *Vzdálené dotazy ad hoc*
- *Rozšířené procedury automatizace OLE*
- *Rozšířené procedury objektů SMO a DMO*

V Systému SQL Server 2005 existoval grafický nástroj zvaný Surface Area Configuration Tool pro správu konfigurace těchto funkcí.. V systému SQL Server 2008 byl tento nástroj odstraněn a nahrazení jeho funkcí je možné dosáhnout pomocí správy zásad.¹¹

2.8 Zálohování dat

Nyní se dostáváme k poslednímu oddílu teoretické části, která se zabývá návrhem databáze, a to zálohováním dat a modelem obnovy dat v databázi. Zálohování dat je proces, při kterém se v daném okamžiku vytvoří jedna nebo více kopií požadovaných dat na záložních datových nosičích. Nejpoužívanějšími médii pro vytváření aktuálních záložních kopií jsou pevné disky, magnetická média, NAS¹² a taktéž Vzdálená zálohovací služba, která využívá přenos dat pomocí vysokorychlostního internetového připojení.

Zálohování dat je strukturováno tak, aby tvořilo prevenci pro případ výpadku systému, selhání nebo jiné události. Umožní pak navíc uvést informační systém do původního nepoškozeného stavu. Základní rozdělení tvorby záloh dělíme na provozní zálohování a

¹¹ WALTERS, R. E. et al. *Mistrovství v Microsoft SQL Server 2008*: [kompletní průvodce databázového experta]. Vyd. 1. Brno: Computer Press, 2009, s. 146. ISBN 978-80-251-2329-4.

¹² Network Attached Storage je pevný disk nebo skupina pevných disků, které jsou připojeny k lokální síti. Může se též jednat o jednoúčelové záložní zařízení nebo server, jehož úlohou je skladování dat.

fyzickou archivaci. Podle účelu, objemu zálohovaných dat a podle požadavků na náročnost a trvání obnovy se rozlišujeme několik spolu souvisejících variant zálohování, které se navzájem prolínají:

- Kompletní zálohování (full backup)
- Rozdílové zálohování (differential backup)
- Záloha transakčního protokolu
- Zálohování souborů tvořících databázi

Samotné zálohování dat by mělo být víceúrovňové, jednak pro rychlé vytvoření zálohy a následné obnovy dat z databáze, jednak pro snížení rizika konfliktu při hierarchii úrovně zabezpečení. Při trojstupňovém zálohování dat z databáze by první stupeň mohl být vyřešený pomocí kopie - mirroringu¹³ databáze na záložní server. Druhý stupeň lze realizovat denním přírůstkovým zálohováním na externí server nebo paměťová média. Třetím stupeň je týdenní kompletní zálohování na pásku.

Modely obnovy databáze

Všechny databáze v instanci systému SQL serveru mají předem definovány parametry pro model obnovy. Model obnovy nad databází je možné aktivovat a měnit pomocí nástroje SQL Server Management Studio. Děje se tak v dialogu vlastnosti příslušné databáze ve složce Options. Modul obnovy databáze disponuje třemi modely obnovy (úplné, s hromadným protokolem, jednoduché):

- Full Recovery Model – zabezpečí kompletní datovou ochranu. Transakční protokol obsahuje informace o změnách v datech i ve struktuře a operacích hromadného nahrávání dat (bulk loading). Obnovujeme-li data z transakčního protokolu, je nutné mít log neporušený.
- Bulk – Logged Model – podobně jako u Full Recovery Model obsahuje všechny informace, některé ale ve zjednodušené podobě.

¹³ Mirroring je poměrně jednoduchá, ale velmi efektivní ochrana dat. Provádí se zrcadlení obsahu disků tak, že obsah se současně zaznamenává na dva disky. V případě výpadku jednoho disku se pracuje s kopií, která je ihned k dispozici.

- Simple Recovery Model – obsahuje stejné informace jako Bulk-Logged Recovery Model, ale má rozdílné chování v kontrolních bodech (CHECKPOINT).

Pokud je databáze v režimu obnovy FULL, jsou všechny prováděné změny v knihovných příkazů DDL¹⁴ a DML¹⁵ zapisovány do transakčního protokolu. Je nezbytná konfigurace každé produkční databáze tak, aby pracovala v režimu obnovy Full. Model Full zabezpečí obnovu databáze do libovolného časového bodu v takovém rámci, jak je nastavena struktury zálohy.[3]

2.9 Požadavky na hardware a software

Před samotnou instalací SQL Serveru 2008 Standard musíme prověřit, zda máme dostatečné kapacity pro nepřetržitý provoz systému. Je nutné část prozkoumat, protože případně nedostačující hardwarová nebo softwarová podpora může navýšit negativní uživatelskou odezvu, nebo dokonce může systém zkolabovat.

Požadavky na hardware

„Požadavky na systém SQL Serveru 2008 Standard pro 32bitové i 64bitové architektury je požadovaný procesor kompatibilní s procesory Pentium III nebo rychlejšími. Minimální rychlost procesoru je však 1,0 GHz, přičemž je doporučeno 2,0 GHz. Pro architekturu 164 je vyžadován minimálně procesor Uranium a 512 MB operační paměti RAM. Praktické doporučené minimum je však RAM 2 GB. Verze pro architekturu x64 vyžadují procesor AMD Opteron, AMD Athlon 64, Intel Xeon s podporou Intel EM64T, případně Intel Pentium IV s podporou EM64T. Minimum operační paměti RAM je 512 MB, praktické doporučené minimum je taktéž RAM 2 GB.“¹⁶

¹⁴ DDL - Data Definition Language, knihovna obsahující příkazy pro vytváření struktury databáze jako jsou tabulky, indexy, pohledy a další objekty potřebnou pro definici příkazů (CREATE, ALTER DROP).

¹⁵ DML - Data Manipulation Language, knihovna obsahující příkazy pro získání dat z databáze a pro jejich úpravy (SELECT, INSERT, UPDAT, MERGE, DELETE, EXPLAIN, SHOW).

¹⁶ LACKO, Euboslav. *Jak vyžrát na Microsoft SQL Server 2008: správa, konfigurace, programování*. Vyd. 1. Brno: Computer Press, 2009, s. 24. ISBN 978-80-251-2101-6.

Minimální hodnoty požadavků na hardware jsou uvedeny kvůli instalaci do notebooku nebo stolního PC. Využití slouží pouze pro potřeby programátorů, kteří vyvíjejí aplikace a musí na testovacích datech ověřovat napsané funkce nebo procedury. Pro plnohodnotný provoz systému SQL Serveru 2008 Standard jsou minimální požadavky na hodnoty paměti RAM a procesoru však nedostačující.

Požadavky na software

„Pro architekturu 164 je vyžadován operační systém Windows Server 2008 64-bit Uranium Edition, případně operační systém Windows Server 2003 SP2 64-bit Itanium Data Center nebo Windows Server 2003 SP2 64-bit Itanium Enterprise. Verze pro architekturu x64 mohou být nainstalované a provozované na operačních systémech Windows XP 64-bit x64 Profesional 2003, Windows Server 2003 SP2 64-bit, Windows Vista 64-bit x64, Windows Sever 2008 64-bit x64 (s architektonickou vrstvou Hyper-V nebo bez ní).“¹⁷

2.10 Návrh APLIKACE

Je třeba se seznámit s druhou procedurální částí Návrhu informačního systému, a to samotnou aplikací, která tvoří rozhraní mezi uživatelem a vzdáleným serverem.

Příprava návrhu a tvorby aplikace musí být v souladu s návrhem a tvorbou datové části. Pro tvorbu uživatelského rozhraní představuje příprava aplikace identickou analýzu stejně jako příprava datové části. Zahrnuje detailní seznámení se všemi odděleními obsaženými v infrastruktuře podniku. Je nutné znát základní uživatelské požadavky pro správnou a účelnou funkčnost vyvíjených prvků a nástrojů, které poslouží uživatelům k plnění každodenních rutinních úkolů. Při psaní kódů procedurálního jazyka bylo využito pomocné grafické vizualizace zobrazující tabulky, relační vztahy a jednotlivé identifikátory aplikace. Tato vizualizace nám umožnila rychlou a účelnou orientaci při vývoji dalších větví programu, tím se rozumí propojení formulářů, modulů a tiskových sestav.

¹⁷ LACKO, Euboslav. *Jak vyzrát na Microsoft SQL Server 2008: správa, konfigurace, programování*. Vyd. 1. Brno: Computer Press, 2009, s. 24. ISBN 978-80-251-2101-6.

2.11 Platforma aplikace program Microsoft Access

V minulosti jsme již implementovali aplikace pro potřeby podniku na platformě Microsoft Access. Abychom využili naše dosavadní zkušenosti, rozhodli jsme se i nyní pro vývoj aplikace na této platformě. Program Microsoft Access je součástí Microsoft Office, který dovede s ostatními programy tohoto programového balíku spolupracovat. Tuto spolupráci využijeme v přenosu objektů metodikou OLE mezi jednotlivými aplikacemi, které používají především programovací jazyky - Visual Basic for Applications (VBA) a Structured Query Language (SQL) .

2.11.1 Psaní kódu ADO, VBA v Projektu(ADP)

Programovací jazyk (Visual Basic for Applications)¹⁸ je základním jazykem pro programování událostí Projektu(ADE). Jedná se o událostmi řízený jazyk vývojového prostředí, který obsahuje standardní i specifické funkce pro vývoj aplikací.

2.11.2 Obecné funkce VBA

VBA (Visual Basic for Applications) obsahuje stovky funkcí, které se dají rozdělit do šesti základních skupin.

- Funkce data a času
- Textové funkce
- Funkce ovládání souborů
- Matematické funkce
- Zkušební funkce
- Převáděcí funkce

VBA (Visual Basic for Applications) je navržen tak, aby bylo jednoduché jej ovládnout a používat. Výhodou jazyka VBA je, že obsahuje všechny potřebné funkce, a hlavně umožňuje vytvářet vlastní funkce pro komplexní řešení složitých aplikací.

¹⁸ VBA Visual Basic for Aplikation je událostmi řízený programovací jazyk od společnosti Microsoft pro programovací model COM. Visual Basic umožňuje rychlý vývoj aplikací s grafickým uživatelským rozhraním přístupným k databázím prostřednictvím DAO.

Na základě jejich kombinací můžeme definovat vlastní funkce nebo makra, jež jsou definovány syntaxí: Function nazev_funkce (nazev_promenna1 As ..., nazev_promenna2 As ..., ...). Název musí být bez mezery, bez českých diakritických znaků a také nesmí obsahovat klíčová slova typu Delete, Select, která jsou obsažena v systémových knihovnách DDL a DML.

Funkce data a času

Aby nedocházelo k omylům při zadání data a nemuseli jsme ošetřovat chyby nekompletního textového řetězce kódem, lze na formuláři v textovém poli předdefinovat formát záznamu, který předpokládáme v datovém poli tabulky SQL Serveru. Musíme mít na paměti, že SQL Server umožňuje výběr pouze ze šesti základních typů ukládání data a času v tabulce SQL Serveru. Jednou z dalších možností prevence chyb např. využít uložení data do Znakového pole.

Textové funkce

Hlavní účelem textových funkcí je dostat textový řetězec do podoby očekávaného formátu. Textové funkce například porovnávají délky řetězců, odstraňují číselné hodnoty ze znakových řetězců, rozkládají a skládají cesty k potřebným souborům, mění velká písmena na malá a vykonávají spoustu dalších užitečných činností určených pro řešení zadaných úloh.

Funkce pro ovládání souborů

Nejvíce využívané funkce slouží pro vytvoření, vymazání nebo změnu názvů adresářů. Funkce využíváme pro nové použití standardních adresářů, vybrání aktuálního adresáře i výběr souborů, které jsou v adresářích uloženy. Do této kategorie rovněž náleží funkce kopírování souborů, určení a nastavení přístupového režimu, spouštění externího programu nebo spouštění programu Internet Explorer a stažení například obrázků.

Matematické funkce

Nepostradatelnými pro každou aplikaci jsou matematické funkce, které vypisují absolutní hodnoty, čtou znaménka hodnot, zajišťují celočíselné hodnoty, zaokrouhlují čísla a tvoří náhodná čísla, např. výpočty odpisů degresivní nebo lineární variantou. Vestavěné matematické funkce se využívají také při výpočtu úrokové sazby.

Zkušební funkce

Zkušební funkce slouží k prověření zadání, kontroly datových polí, kontroly tabulek. Zkušební funkce použijeme vždy, když chceme kontrolovat zadání ve formuláři nebo jiném dialogu v uživatelském rozhraní. Prověřujeme jimi kontroly zadání, kontroly datových typů nebo nám poslouží při kontrole existence určitého objektu, jako je tabulka.

Převáděcí funkce

Mnohdy potřebujeme převést jednotlivé datové typy z důvodu výpočtu, odstranění teček, přeměny hodnot data, odstranění míst za desetinnou tečkou, extrakce čísla ze znakového řetězce nebo naopak přeměny čísel na textový řetězec. Pro tyto uvedené příklady využíváme vestavěné převodní funkce.

2.11.3 Formuláře Projektu(ADP)

Dalším důležitým objektem jsou Formuláře Projektu (ADP). Formuláře jsou objekty, které tvoří uživatelské rozhraní objektově orientovaných aplikací. Každý formulář může být propojen s jednou nebo více tabulkami databáze, přičemž obsahuje identická datová pole, jako má propojená tabulka. Grafické uspořádání formuláře je koncipováno tak, aby vyhovovalo požadavkům uživatele a aby zahrnovalo veškeré požadované funkcionality potřebné pro funkčnost aplikace. Formuláře dělíme na vázané a nevázané. Vázané formuláře mohou být založeny na tabulkách, zobrazení, uložených procedurách a příkazech Transact-SQL, které následně vracejí hodnoty. Pokud je formulář v projektu založen na dotazu, je aktualizovaný pouze tehdy, pokud má uživatel potřebná oprávnění, aby mohl zadat hodnoty do tabulky. V případě, že uživatel nemá dostatečná oprávnění, je nutné napsat kód VBA pro aktualizaci nevázaného formuláře.

Vázané formuláře mají určený zdroj dat, přičemž se zobrazí jednotlivá pole v seznamu formuláře. Nevázané formuláře nemají žádný datový zdroj, používají se jako dialogová okna, a dále se také používají pro navigaci pomocí ovládacích prvků, tlačítek, přepínačů, výběrových polí umístěných na formuláři.[7]

2.11.3.1 Formulář a jeho oddíly

Záznamy a zápatí formuláře většinou obsahují nadpisy a ovládací prvky. Tělo formuláře obsahuje záznamy datového pole a v případě, že se nebude záhlaví a zápatí stránky tisknout, je zbytečné záhlaví a zápatí tvořit. Oddíl Seznam polí obsahuje pole datového zdroje. Okno vlastností obsahuje vlastnosti formuláře, kde jsou jeho součástí i ovládací prvky polí, která se dále dělí podle druhu na jednotlivé karty. Posledním oddílem je Sada nástrojů, jejíž ovládací prvky vkládáme do formuláře. „Užitečným využitím zápatí je zobrazení podrobných informací o aktivním záznamu tabulkového zobrazení formuláře. Tělo takového formuláře může obsahovat vybraná nejdůležitější pole záznamu v tabulkovém zobrazení nebo zobrazení datového listu. Zápatí je pak možné využít pro zobrazení všech informací aktivního záznamu (na kterém je umístěn volič záznamů) ve sloupcovém zobrazení.“¹⁹

2.11.3.2 Ovládací prvky

Mezi významnou skupinu nástrojů pro ovládání funkčnosti uživatelského rozhraní patří Ovládací prvky. Do formuláře můžeme přidat mnoho ovládacích prvků, jejichž prostřednictvím budeme přistupovat k datům ve formuláři nebo jejichž prostřednictvím budeme ovládat veškeré procesy, které jsou deklarovány v aplikaci. Ovládací prvky dělíme na nevázané, jenž nemají datový zdroj, a vázané, které datový zdroj mají. Nejpoužívanějším ovládacím prvkem jsou tlačítka, pole se seznamem, přepínače a zaškrťovací políčka. Můžeme použít nejen další ovládací prvky typu Kalendář, ProgressBar, SpinButton, Office Chart, Media Player, TreeView, ale také ovládací prvek Slide a spoustu dalších velmi užitečných ovládacích prvků, které mohou posloužit pro zvýšení uživatelského komfortu při plnění každodenních rutinních úloh.[7]

¹⁹ MORKES, David. *Microsoft Office Access 2003* Podrobná uživatelská příručka. 639 00 Brno : COMPUTER PRESS, 2004. 350 s. ISBN 80-251-0179-7, K0977.

2.11.4 Sestavy Projektu (ADP)

Neméně důležitý nástroje v uživatelském rozhraní jako jsou formuláře, představují tiskové sestavy. Vytváření sestavy se z technického hlediska neliší od vytváření formuláře. Pro sestavy v Projektech(ADP) je nejefektivnějším způsobem použít zobrazení, v nichž jsou vybrány pouze požadované sloupce s daty nebo uložené procedury, které používají parametry a vrací výsledky. Jednou z možností je použít vlastností Input Parameters pro doplnění parametru do uložené procedury, která používá totožný zdroj parametru pro sestavu, jaký je obsažen ve formuláři, ze kterého sestavu otevíráme.

2.11.4.1 Řízení a seskupování dat v sestavách

Data v sestavách můžeme řídit a seskupovat pomocí vestavěných funkcí Podokna řazení sestavy Projektu(ADE). Podokno řazení a součet slouží k identifikaci jednotlivých úrovní různých seskupení k tomu, abychom poznali též pořadí, v jakém jsou řazena data v sestavě Projektu(ADE). Výhodou je, že můžeme data v sestavě řadit i pomocí vypočtené hodnoty, dokonce postačí namísto hodnoty zadat výraz.

2.11.5 Procedury a Moduly VBA

Modul je kolekce deklarácí, procedur a příkazů, které jsou společně uloženy v pojmenovaném bloku. Rozsáhlejší aplikace mají větší množství formulářů a procedur událostí, které využívají určité proměnné a rutiny. Implicitně je každá lokální proměnná pro svoji vlastní proceduru událostí. Tato lokální proměnná se dá číst a měnit jen z té procedury událostí, ve které byla vytvořena. Modulem rozumíme samotný databázový objekt, který je koncipován jako soubor textu. Jedná se o deklaraci jazyka Visual Basic, jehož syntaxe tvoří procedury, funkce a další náležitosti, které představují součásti jazyka Visual Basic. Moduly dělíme do dvou základních typů: moduly tříd a standardní moduly.

2.11.5.1 Moduly tříd

V aplikaci MS Access mohou moduly tříd existovat také nezávisle na formulářích a sestavách. Moduly tříd rozumíme moduly formulářů a moduly sestav, přidružených k určitému jiném formuláři nebo sestavě. Moduly tříd obsahují události procedury, které jsou spuštěny jako odezva na událost ve formuláři nebo sestavě. Pomocí událostních

procesů lze řídit chování formulářů a sestav a jejich odezvy na akce uživatele podle uživatelských potřeb.

2.11.5.2 Standardní moduly

Standardní moduly nalezneme pod položkou Moduly v oblasti Objekty. Standardní moduly obsahují obecné procedury, které nejsou přidruženy k žádnému objektu. Jedná se o procedury, které můžeme spustit na libovolném místě Projektu(ADP), aby nám zabezpečily funkčnost správnosti výpočtů. Název každé procedury je základní jednotkou programových modulů a jejich názvy se řídí zcela přesnými pravidly. Syntaxe názvu může obsahovat argumenty procedur, které se zapisují do závorek za název procedury.[7]

2.12 Zabezpečení Projektu(ADP)

Poslední část oddílu Návrhu aplikace uvádí možnosti zabezpečení Projektu(ADP). Pomocí programování VBA máme přístup k modulům, makrům, sestavám a formulářům v Projektu(ADP). Protože se většinou jedná o důvěrná data, která se využívají pro další zpracování a vyhodnocování, je nutné se zabývat více úrovněmi zabezpečení. Je třeba chránit zdrojový kód Projektu(ADP), můžeme k tomu použít funkce zabezpečení Projektu(ADP) pomocí parametrů spuštění, dále máme možnost nastavit i ochranu Projektu heslem. Lze využít též další způsoby ochrany, jako např. možnost uložení Projektu s přeloženým zdrojovým kódem ve formátu ADE. Ze zkušeností víme, že důraz na zabezpečení jak samotného Projektu(ADE), tak i nastavení přístupu a uživatelských práv na SQL Serveru je nezbytným procesem pro jednoznačné vymezení zadání požadavků kladených na uživatele.

2.12.1 Uložení v podobě souboru (ADP)

Mezi nejbezpečnější metody ochrany přístupu ke kódu VBA je uložení Projektu v podobě souboru (ADE). Tento postup zkompile všechny moduly, odstraní všechny upravitelné kódy a následně přiřadí Projekt k souboru (ADE). Soubor Projektu (ADE) je menší než soubor Projektu (ADP), většinou má i lepší výkon.

2.12.2 Možnost po spuštění

Dialogové okno Po spuštění (Nástroje, Po spuštění) nabízí možnosti pro omezení přístupu uživatelů k různým částem Projektu pomocí zaškrťovacích políček. Znamená to zobrazení okna databáze, povolení pracovat s úplnou nabídkou nebo použít speciálních kláves, označíme-li příslušná zaškrťovací políčka. Uživatelé neuvidí okno databáze a nebudou mít možnosti je obnovit ani klávesovými zkratkami.

Tato možnost je dobrým způsobem, jak zabránit uživatelským chybám v práci s Projektem(ADP) prostřednictvím nabídek a jejich prvků, tohoto pozitiva docílíme díky přednastavené posloupnosti zadání v uživatelském rozhraní.[7]

2.12.3 Uživatelská úroveň zabezpečení

Nejpropracovanějším typem zabezpečení je uživatelská úroveň zabezpečení Projektu(ADP) na bázi procedur a funkcí propojených s implementovanými funkcemi SQL Serveru. Na základě přiřazených uživatelských práv přístupu do objektů Projektu(ADP) docílíme různé úrovně zabezpečení. K nastavení uživatelských práv je třeba definovat všechny objekty pro povolení přístupu a dále závod a user name uživatele. Z důvodu kumulovaných funkcí osob v menších závodech se nám tento způsob osvědčil a již nebylo nutné definovat kategorie skupiny uživatelů.

II. PRAKTICKÁ ČÁST

3 ANALÝZA DATOVÝCH STRUKTUR

Hlavním praktickým úkolem je sloučení několika datových subsystémů, ze kterých vytvoříme jeden celek, který představuje jedinečný systém využívající dvě databáze v prostředí Microsoft SQL Serveru. Vývoj a implementace Nadstavbového modulu ERP systému pro výrobu a řízení s pracovním názvem systému LEKY, Docházkový systém Watt a ERP systém SAP, využívají databáze sca_sql_2008 a systém Navigator využívající databázi Navigator. Data uložená v databázi Navigator budou primárním požadovaným datovým výstupem, kterému se musí přizpůsobit ostatní datové struktury subsystémů. Datové typy (asociativní pole, textové řetězce, zásobníky, seznamy) musí být navrženy tak, aby splňovaly požadavky pro zápis povinných i nepovinných polí jednotlivých tabulek databáze Navigator s nutností zachovat relační vztahy a referenční integritu.

3.1 Systém Navigator

Architektura dat v databázi Navigator je navržena v technologii OLAP, která umožňuje uspořádání velkých objemů dat, jež se již dále nemodifikují, ale nad kterými běží složité vyhodnocovací dotazy. Jedná se o dvojrozměrné tabulkové uspořádání, kdy každá datová dimenze je uložena v jedné ose datové kostky. Samotná struktura databáze Navigator je tvořena relačními tabulkami ve hvězdicovém schématu, přičemž jedním z úkolů je vytvoření upload, externích dat, která zabezpečí požadovanou datovou strukturu.

3.1.1 Základní pravidla importu dat do databáze Navigator

„Všechna data v databázi musí být získána přímo ze zdrojového systému, pokud tomu nebrání žádný jiný důvod. Všechny informace z objednávky zákazníka a z vystavených faktur za vybrané období se budou shromažďovat v databázi Navigator.

Vzhledem k tomu, že ziskovost může být vypočítána, pouze pokud jsou vystaveny všechny faktury, budou veškeré náklady shromážděny pro analýzu ziskovosti. Dále zde budou zahrnuty záznamy vystavených zákaznických objednávek, které budou shromažďovány, aby se mohly porovnávat účty předvýrobních a povýrobních kalkulací.

Případné dodatečné náklady, dobropisy nebo faktury s vícenáklady, které se vztahují k zákaznické objednávce budou aktualizovány k datu vytvoření dokladu. Tyto informace

mohou změnit samotné vyhodnocení zákaznické objednávky, kdy se zakázka vyhodnotí jako zisková, poté se v následném období může zakázka vyhodnotit jako nerentabilní.

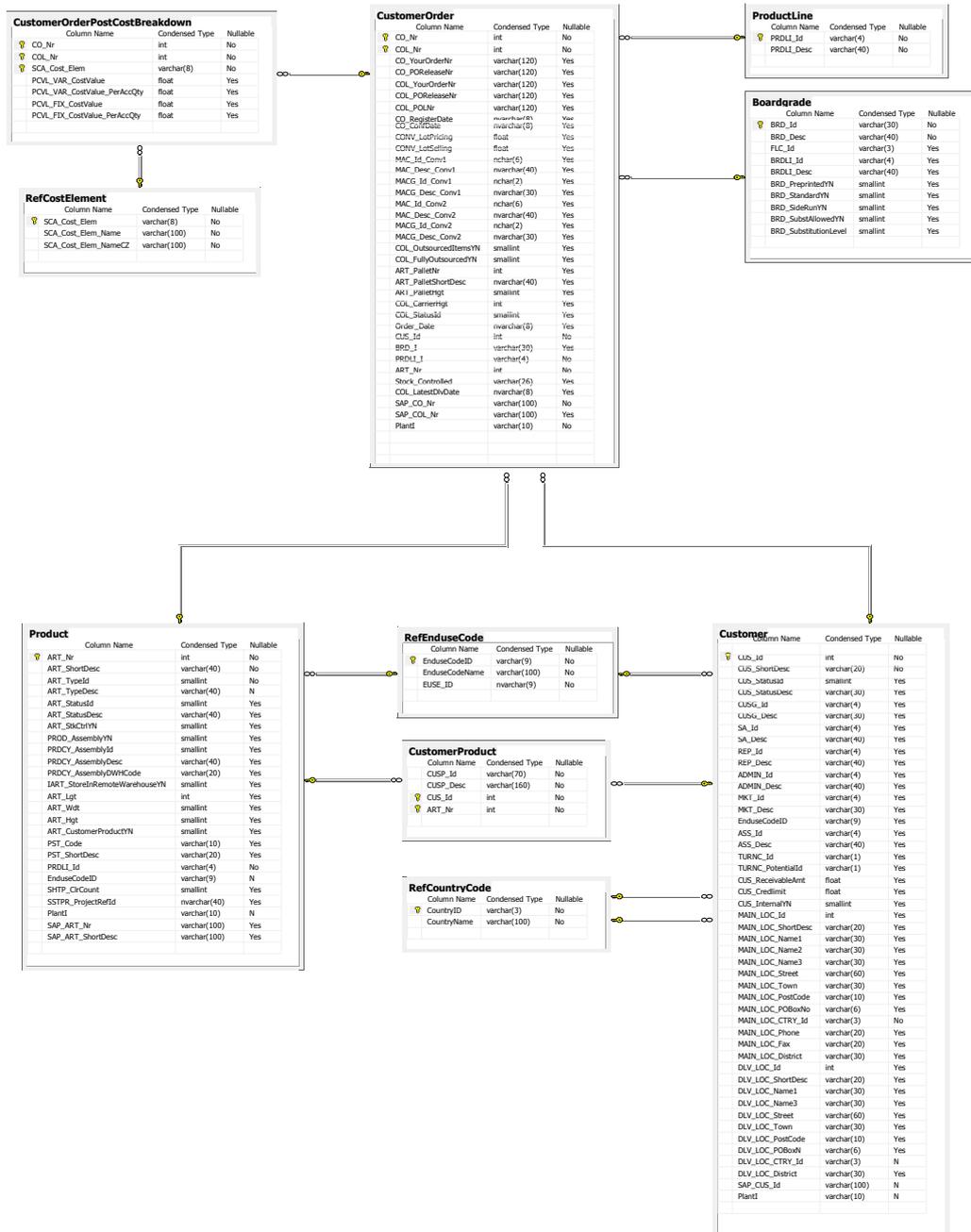
Preferovaný způsob získávání dat do databáze je použití extraktu, který pravidelně přijímá data ze systému ERP. Datová struktura je založena na standardním kalkulačním modelu, který byl odsouhlasen mezi výrobou a ekonomickým oddělením a dále schválen finančním oddělením. Na žádnou jinou strukturu nebude brán zřetel.

Vzhledem k nutnosti získání požadovaných informací, které jsou nezbytné pro výpočet nákladů, se musí vždy odkázat na tabulky dimenzí ("D_" Tabulka). Není-li možné tato data získat, pak je třeba pole označit fiktivními záznamy (například ID: '-1'; popisu: "není-li k dispozici").“²⁰

²⁰ JONES, Martin. *ExtractSpec Navigator_Original V4.0.doc* : Projektové zadání návrhu datové struktury a zadání relačních vztahů v databázi Navigator. [2011]

3.2 Relační schéma databáze Navigator

Relační schéma databáze Navigator znázorňuje zadání očekávané datové struktury, vzájemného propojení jednotlivých tabulek relační databáze a požadovaných datových typů.



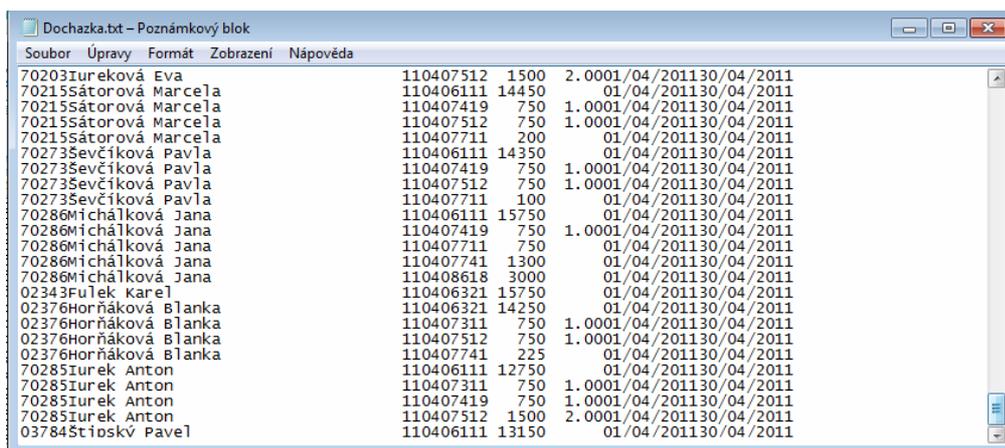
Obrázek č. 1: Relační schéma systému Navigator

3.3 Docházkový systém Watt

Systém docházky slouží k evidenci a automatickému zpracování docházky a odpracované doby s využitím identifikačních karet. Slouží k sledování pohybu zaměstnanců v průběhu pracovní doby pro další zpracování. Identifikační medium je bezkontaktní karta, která při načtení do systému určí identitu zaměstnance. V každém ze závodů je ve vstupní hale umístěn nástěnný přístupový terminál, jehož prostřednictvím se zapisují data o vstupu a výstupu do databáze docházkového systému Watt.

3.3.1 Výstupní sestavy, export dat

V editoru sestav je možno vytvářet vlastní tiskové sestavy a exporty dat. Všechny výstupní sestavy lze ukládat ve zvoleném formátu: txt, csv, rtf, html, xls, jpg, gif, bmp, emf, wmf a pdf. Tabulková data exportovaná do textového souboru, která neobsahují žádné další formátování s výjimkou oddělovače jednotlivých polí, mají výhodu, že tato uložená data jsou čitelná kterýmkoli textovým procesorem a při zadání oddělovače polí je lze načíst prakticky do jakéhokoli databázového systému nebo tabulkového kalkulátoru. Nevýhodou je, že při tomto typu exportu ztratíme veškeré dodatečné informace, které datová tabulka obsahovala.



Dochazka.txt – Poznámkový blok

Soubor	Úpravy	Formát	Zobrazení	Nápověda
70203Iureková Eva	110407512	1500	2.0001/04/201130/04/2011	
70215Sátorová Marcela	110406111	14450	01/04/201130/04/2011	
70215Sátorová Marcela	110407419	750	1.0001/04/201130/04/2011	
70215Sátorová Marcela	110407512	750	1.0001/04/201130/04/2011	
70215Sátorová Marcela	110407711	200	01/04/201130/04/2011	
70273Ševčíková Pavla	110406111	14350	01/04/201130/04/2011	
70273Ševčíková Pavla	110407419	750	1.0001/04/201130/04/2011	
70273Ševčíková Pavla	110407512	750	1.0001/04/201130/04/2011	
70273Ševčíková Pavla	110407711	100	01/04/201130/04/2011	
70286Michálková Jana	110406111	15750	01/04/201130/04/2011	
70286Michálková Jana	110407419	750	1.0001/04/201130/04/2011	
70286Michálková Jana	110407711	750	01/04/201130/04/2011	
70286Michálková Jana	110407741	1300	01/04/201130/04/2011	
70286Michálková Jana	110408618	3000	01/04/201130/04/2011	
02343Fuřek Karel	110406321	15750	01/04/201130/04/2011	
02376Hornáková Blanka	110406321	14250	01/04/201130/04/2011	
02376Hornáková Blanka	110407311	750	1.0001/04/201130/04/2011	
02376Hornáková Blanka	110407512	750	1.0001/04/201130/04/2011	
02376Hornáková Blanka	110407741	225	01/04/201130/04/2011	
70285Iurek Anton	110406111	12750	01/04/201130/04/2011	
70285Iurek Anton	110407311	750	1.0001/04/201130/04/2011	
70285Iurek Anton	110407419	750	1.0001/04/201130/04/2011	
70285Iurek Anton	110407512	1500	2.0001/04/201130/04/2011	
03784Štípský Pavel	110406111	13150	01/04/201130/04/2011	

Obrázek č. 2: Soubor Dochazka.txt, sekvence textových znaků

3.3.2 Datové typy souboru Docházka.txt

Abychom mohli exportovaná data z docházkového systému Watt i dále využívat pro potřebné analytické vyhodnocení v systému LEKY, bylo nezbytné vytvořit adekvátní objekt, tabulku tblDochazka v databázi sca_sql_2008, do které se budou data importovat.

Tabulka č. 5: Popis datových typů tabulky tblDochazka

Column_Name	Data Type	Description
osoba_id	Bigint	ID Zaměstnanec
osoba_jmenoprijmeni	varchar(100)	Jméno a příjmení zaměstnanec
osoba_mzdovokod	varchar(50)	Pomocná data mzdové složky
osoba_pocethodin	decimal(10, 2)	Počet hodin zaměstnanec
osoba_pocetdnu	decimal(10, 2)	Počet evidovaných dnů
osoba_datumod	Smalldatetime	Datum příchodu zaměstnanec
osoba_datumdo	Smalldatetime	Datum odchodu zaměstnanec

3.3.3 Mzdová složka, datové typy

Pro správnou funkci využití exportovaných dat ze souboru Dochazka.txt je nutné nejdříve vygenerovat z docházkového systému všechny identifikátory mzdové složky, které se importují do tabulky tblMzdovaSlozka v databázi sca_sql_2008. Identifikátory mzdové složky určují nestandardní možnosti, jež se po dobu pracovní doby zaměstnanec můžou vyskytnout.

Tabulka č. 6: Popis datových typů tabulky tblMzdovaSlozka

Column_Name	Data Type	Description
MzdovaSlozka_ID	Int	ID Mzdové složky
MzdovaSlozka_Kod	nvarchar(50)	Kód mzdové složky
MzdovaSlozka_Nazev	nvarchar(50)	Název mzdové složky

3.4 ERP Systém SAP

Systém SAP je jedním z největších podnikových informačních systémů na světě využívaný především v automobilovém a elektronickém průmyslu. SAP R/3 je v topologii client/server, přičemž aplikace využívá třívrstvý model. Prezentační vrstva nebo klient komunikuje s uživatelem. V aplikační vrstvě je uložena business logika a databázová vrstva zaznamenává a ukládá všechna data systému včetně transakčních a konfiguračních dat.

„Seznam modulů systému SAP:

- *FI (Financial Accounting) Finanční účetnictví*
- *CO (Controlling) Kontroling*
- *AM (Asset Management) Evidence majetku*
- *PS (Project system) Plánování dlouhodobých Projektů*
- *WF (Workflow) Řízení oběhu dokumentů*
- *IS (Industry Solutions) Specifická řešení různých odvětví*
- *HR (Human Resources) Řízení lidských zdrojů*
- *PM (Plant Maintenance) Údržba*
- *MM (Materials Management) Skladové hospodářství a logistika*
- *QM (Quality Management) Management kvality*
- *PP (Production Planning) Plánování výroby*
- *SD (Sales and Distribution) Podpora prodeje*²¹

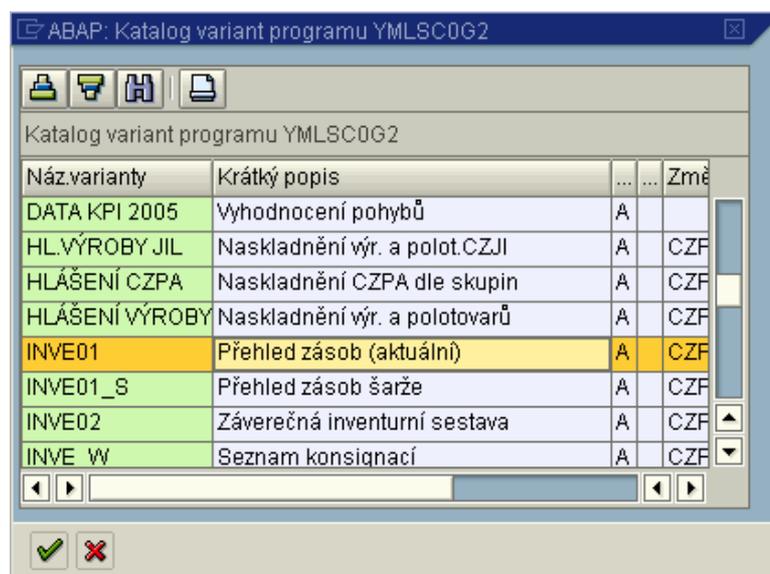
3.5 Export dat ze systému SAP

Pro propojení systémů je třeba rozklíčovat identifikátory evidenčních karet materiálů a výrobků. V systému SAP je tento jedinečný kód nazýván IDOB, jehož struktura charakterizuje evidovaný materiál. Jedná se o 18-ti znakový textový řetězec, jehož kombinace čísel a textu na různých pozicích vyjadřuje jedinečnou identifikaci kódu skladových karet. Pro zjednodušené zadání a zamezení duplicit je vytvořena aplikace, která umožňuje vygenerovat kód materiálové karty. Data aplikace nahradí samotný export dat ze systému SAP a umožní nastavit funkci denního exportu materiálových karet do předdefinovaného adresáře. Z uživatelského rozhraní systému SAP využijeme transakci

²¹ Business info – Oficiální portál ERP Systému SAP Česká republika: Seznam modulů systému SAP [online]. [cit. 2012-04-20]. Dostupné z WWW:

http://www.sap.com/cz/campaign/2009_07_CROSS_ERP_GENERAL/index.epx?campaigncode=CRM-CZ12-RDC-PPC_ABA_01&dna=117800,8,0,95426597,806212279,1335264139,Moduly

YMLSC0G2, varianta INVE01, „Přehled zásob“ - aktuální, která umožňuje export aktuální skladové zásoby. Tato transakce se provádí denně ve čtyřhodinovém intervalu a data budou použita pro pohyb skladových karet. Tyto hodnoty aktuální zásoby materiálu a výrobků poslouží pro evidenci volně použitelné zásoby, rezervaci materiálu a výrobků, sledování nedodaného materiálu.



Obrázek č. 3: Uživatelské rozhraní transakce YN0DCOZIG2

3.6 Systém SAP, datové typy

Ze systému SAP pro potřeby sledování aktuální skladové zásoby a seznamu materiálových karet využijeme dva na sobě nezávislé výstupy. Jedná se o jednoduché formáty .dbf a .txt. Pro aktualizaci skladových karet je nastaven výchozí textový formát (dBase) soubor .dbf a pro data aktuální skladové zásoby je nastaven výchozí formát .txt. Z důvodu potřebného importu dat byly v databázi sca_sql_2008 vytvořeny tabulky tblOBALY_LE a tblLEPENKY_LE, do kterých se ukládají data identifikátorů obalů a lepenek, které obsahují klíče (IDOB) a tabulku tblZustAPLIn, do které se ukládají data aktuální skladové zásoby ERP systému SAP.

Tabulka č. 7: Popis datových typů tabulky tblOBALY_LE

Column_Name	Data Type	Description
OBALY_ID	Int	ID Obalu
IDOB	nvarchar(18)	Individuální číslo skladové karty
TEXT	nvarchar(40)	Název skladové karty
TYP	nvarchar(1)	Typ skladové karty
KONSTR	nvarchar(6)	Konstrukce obalu
DELKA	Float	Délka obalu
SIRKA	Float	Šířka obalu
VYSKA	Float	Výška obalu
VLOZKA	nvarchar(4)	Výrobek s vložkou
MAT1	nvarchar(11)	Materiálová specifikace č.1
MAT2	nvarchar(11)	Materiálová specifikace č.2
MAT3	nvarchar(11)	Materiálová specifikace č.3
TISK	nvarchar(7)	Tisková specifikace
VAHA	Float	Hmotnost výrobku
SKUPINA	nvarchar(10)	Skupina obalů
CENA	Float	Cena obalu za ks
PL_1000KS	Float	Cena za 1000 ks

3.7 Vytváření tabulek v databázi SCA_SQL_2008

Ted, když jsme se již seznámili se strukturou datových polí databázových tabulek, které použijeme pro uplout do databáze SCA_SQL_2008 a taktéž když již známe očekávaná datová pole a relační vztahy tabulek databáze Navigator, můžeme začít vytvářet nové tabulky. Ještě před samotným vytvořením tabulky nesmíme opomenout, že jsme se rozhodli propojit platformu Microsoft Access, projekt (ADP) jako procedurální část a je tedy nutné pro editaci a vytváření nových záznamů, aby každá ze vzdálených vytvořených databázových tabulek na SQL Serveru obsahovala primární klíč tabulky PRIMARY KEY.

Pro vytvoření nové tabulky použijeme příkaz jazyka T-SQL CREATE TABLE a dále využijeme konstrukce pro omezení zápisu datových polí NOT NULL, UNIQUE

Abychom mohli uložit do databáze SCA_SQL_2008 veškerá data potřebná pro funkčnost Nadstavbového modulu ERP systému pro výrobu a řízení, bylo zapotřebí vytvořit 105 tabulek.

Úplná syntaxe pro vytvoření nové tabulky F_Order v databázi SCA_SQL_2008 je:

```
USE [sca_sql_2008]
GO

/***** Object: Table [dbo].[F_Order]      Script Date: 04/25/2012
20:13:54 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[F_Order](
    [CO_Nr] [int] IDENTITY(1,1) NOT NULL,
    [Order_Date] [nvarchar](8) NULL,
    [COL_Nr] [int] NOT NULL,
    [CUS_Id] [int] NOT NULL,
    [BRD_Id] [int] NOT NULL,
    [ART_TypeId] [smallint] NULL,
    [PRDLI_Id] [int] NOT NULL,
    [REPLORD_Nr] [int] NULL,
    [ART_Nr] [int] NOT NULL,
    [Stock_Controlled] [nvarchar](26) NULL,
    [IVCL_GrossSqm] [float] NULL,
    [COL_AcceptedQty] [float] NULL,
    [IVCL_GoodsNetAmt] [money] NOT NULL,
    [DELNL_NrOfCarriers] [int] NOT NULL,
    [COL_LatestDlvDate] [nvarchar](8) NULL,
    [LINK_ExchangeRateDate] [nvarchar](8) NULL,
    [Order_CisloSAPiZak] [varchar](50) NOT NULL,

    CONSTRAINT [PK_F_Order] PRIMARY KEY CLUSTERED
    (
        [CO_Nr] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO

EXEC sys.sp_addextendedproperty @name=N'MS_Orientation', @value=0x00 ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'F_Order'
GO

ALTER TABLE [dbo].[F_Order] WITH CHECK ADD CONSTRAINT
[FK_F_Order_D_Customer] FOREIGN KEY([Objednavky_ID_Zak])
REFERENCES [dbo].[D_Customer] ([CUS_Id])
GO

ALTER TABLE [dbo].[F_Order] CHECK CONSTRAINT [FK_F_Order_D_Customer]
GO
```

3.8 Vytváření pohledů v databázi SCA_SQL_2008

SQL Server převážně obsahuje předdefinované Pohledy(Views), které umožňují přístup k datům pro konkrétní funkci bez přímého přístupu k databázovým tabulkám. Jednoduché databázové objekty Pohledy(Views) jsem převážně využíval k účelům datového propojení tiskových sestav a komplexní databázové objekty Pohledy(Views) pro potřeby analytických datových výstupů pro obchodní nebo ekonomické oddělení. Pohled(Views) v databázi SQL serveru vytvoříme pomocí příkazu CREATE VIEW.

Pro potřeby Nadstavbového modulu ERP systému pro výrobu a řízení bylo nutné vytvořit 101 databázových Pohledů.

Syntaxe pro vytvoření nového komplexního Pohledu qryD_Product_first v databázi APL_SQL_2008 je:

```
USE [sca_sql_2008]
GO

/***** Object: View [dbo].[qryD_Product_first]      Script Date:
05/02/2012 11:58:32 *****/

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[qryD_Product_first]
AS
SELECT      COPY_ART_NR, MIN(Objednavky_DatumVyroby) AS ObjFirst
FROM        dbo.D_Product
GROUP BY   COPY_ART_NR
HAVING      (MIN(Objednavky_DatumVyroby) IS NOT NULL)
GO
```

Úplná syntaxe pro vytvoření nového komplexního Pohledu sysque_LIST_table_columns v databázi SCA_SQL_2008 je:

```
CREATE VIEW [dbo].[sysque_LIST_table_columns]
AS
SELECT      TOP 100 PERCENT o.name AS table_name, c.colorder AS
column_order, c.name AS column_name, t .name AS column_type_name,
           c.length AS column_length, c.xprec AS column_prec,
c.scale AS column_scale, c.iscomputed, CASE WHEN charindex22('char', t
.name)
```

²² Charindex vrátí počáteční pozici v rámci jiného výrazu nalezen jeden výraz.

```

        > 0 THEN 'string' WHEN charindex('text', t.name) >
0 THEN 'string' WHEN charindex('datetime', t.name) > 0 THEN 'datetime'
WHEN charindex('bit',
        t.name) > 0 THEN 'boolean' ELSE 'number' END AS
data_type, c.isnullable
FROM      dbo.sysobjects23 o INNER JOIN
        dbo.syscolumns24 c ON o.id = c.id INNER JOIN
        dbo.systypes25 t ON c.xtype = t.xtype AND
c.xusertype = t.xusertype
WHERE     (o.xtype = 'U') AND (o.name LIKE 'tbl%') and (c.name not
in('pocet_hodin_imp', 'zpozdeni_imp'))
ORDER BY o.name, c.colorder[8]

```

3.9 Vytváření uložených procedur v databázi SCA_SQL_2008

V okamžiku, kdy již známe všechny objekty v rozhraní SQL Serveru, můžeme se pustit do psaní kódů pomocí jazyka T-SQL. Uložená procedura v Systému SQL server obsahuje dávky příkazů jazyka T-SQL, pomocí kterého vypíšeme posloupný sled událostí, které od uložených procedur očekáváme. Uložené procedury vytváříme v databázi SQL Serveru pomocí příkazu CREATE PROCEDURE.

Pro potřeby zajištění veškerých funkcí a procedur Nadstavbového modulu ERP systému pro výrobu a řízení, bylo nutné vytvořit 80 uložených procedur.

Úplná syntaxe kódu pro vytvoření nové uložené procedury sp_UPDT_tblDochazka v databázi SCA_SQL_2008, která zajistí import dat ze systému Watt do databázové tabulky tblDochazka, je:

```

USE [sca_sql_2008]
GO
/***** Object:  Stored Procedure [dbo].[sp_UPDT_tblDochazka]      Script
Date: 04/26/2012 10:07:35 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[sp_UPDT_tblDochazka]
(

```

²³ Sys.Objects obsahuje jeden řádek pro každý objekt vytvořený v databázi, například omezení, výchozí, protokol, pravidla a uložená procedura.

²⁴ Sys.Columns vrátí jeden řádek pro každý sloupec v každé tabulce a rovněž zobrazení a řádek pro každý parametr procedury uložené v databázi.

²⁵ Sys.Types obsahuje řádek pro každý systém a uživatelsky definovaný typ.

```

@dochazka_id int = null,
@osoba_id bigint = null,
@osoba_jmenoprijmeni varchar(100) = null,
@osoba_mzdovokod varchar(50) = null,
@osoba_pocethodin decimal(10, 2) = null,
@osoba_pocetdnu decimal(10, 2) = null,
@osoba_datumod smalldatetime = null,
@osoba_datumdo smalldatetime = null
)
)
as
BEGIN
declare @row int, @id_new_row int
set nocount on
if @dochazka_id is null
BEGIN
    Insert tblDochazka
    (
        [osoba_id],
        [osoba_jmenoprijmeni],
        [osoba_mzdovokod],
        [osoba_pocethodin],
        [osoba_pocetdnu],
        [osoba_datumod],
        [osoba_datumdo]
    )

    Values
    (
        @osoba_id,
        @osoba_jmenoprijmeni,
        @osoba_mzdovokod,
        @osoba_pocethodin,
        @osoba_pocetdnu, @osoba_datumod, @osoba_datumdo
    )
END
ELSE
BEGIN
    if not(exists(select * from tblDochazka WHERE
    [dochazka_id] = @dochazka_id ))
    BEGIN
        raiserror('Chyba pri aktualizaci zaznamu v tabulce
        tblDochazka pro dochazka_id: %d', 15, 1, @dochazka_id)
        Return
    END
    ELSE
    BEGIN
        update tblDochazka
        SET
        [osoba_id] = @osoba_id,
        [osoba_jmenoprijmeni] = @osoba_jmenoprijmeni,
        [osoba_mzdovokod] = @osoba_mzdovokod,
        [osoba_pocethodin]=osoba_pocethodin,
        [osoba_pocetdnu] = @osoba_pocetdnu,
        [osoba_datumod] = @osoba_datumod,
        [osoba_datumdo] = @osoba_datumdo
        WHERE [dochazka_id] = @dochazka_id
    END

```

```
END
set @row = @@rowcount26
if @row = 0
Begin

    raiserror('Chyba pri vkladani/aktualizaci zaznamu do tabulky
tblDochazka.', 15, 1)
    Return
End

ELSE

Begin
select scope_identity27() as newID28, @dochazka_id as updtID,
@row as rwCnt
End
END
```

Potřebujeme dvě funkce potřebné pro import dat z docházkového systému Watt, a to uloženou proceduru sp_UPDT_tblDochazka, která je napsána na vzdáleném SQL Serveru a funkci import_lepenky_le() uloženou v uživatelském rozhraní Projektu(ADP). Tuto funkci představím v sekci Moduly Projektu ADP_SCA(ADP).

Pro zajištění funkčnosti mezi jednotlivými objekty SQL Serveru, Projektu (ADP) a vzdálenými Servery jednotlivých závodů bylo nutné vytvořit 83 Uložených procedur.

Doposud jsme se v praktické části seznámili s vytvořením databázových objektů databáze SCA_SQL_2008, ale teď se musíme zabývat plynulým chodem SQL serveru. Pro tyto účely máme k dispozici nástroje aplikace Management studio, pomocí kterých nastavíme potřebné Plány údržby a taktéž parametry pro volání Výstrah v případě, že vznikne konflikt v instanci systému SQL Server.

²⁶ Funkce @@rowcount vrátí počet řádků, které jsou ovlivněny posledním výrazem.

²⁷ Funkce scope_identity vrací hodnotu posledního záznamu ID, která byla vložena do poslední identity.

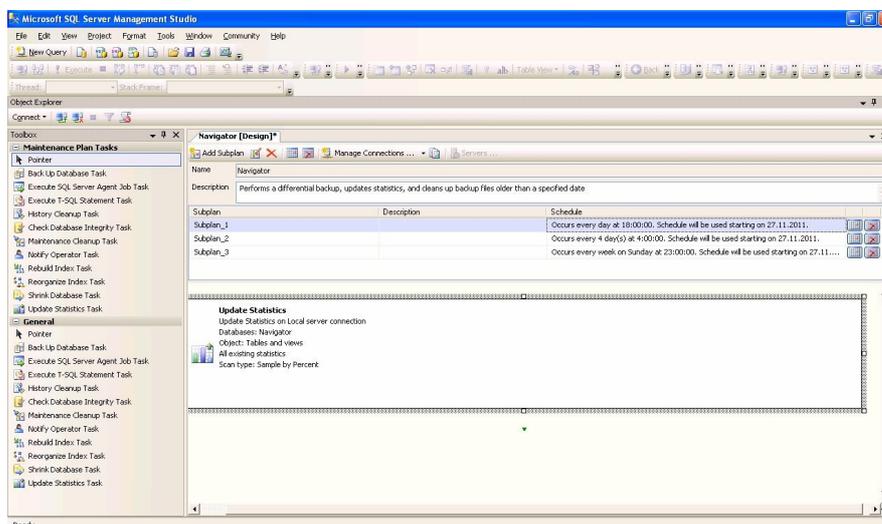
²⁸ NewID vrací hodnotu typu uniqueidentifier.

3.10 Vytvoření plánu údržby databáze Navigator

Plány údržby tvoří každodenní pracovní úkoly, které mimo jiné zajišťují pravidelné zálohování dat databáze Navigator. Nyní si představíme postup, pomocí kterého zajistíme automatické zálohování databáze Navigator, což s pomocí aplikace Management studio dokážeme vytvořit. Nesmíme opomenout, že pro použití Plánu údržby je nutné používat službu SSIS jako modul zálohování. Jinak bychom museli řešit zálohování pomocí vytvoření skriptů v jazyku T-SQL, ze kterého by se daly přímo spouštět Plány údržby.

Vytvoření plánu údržby:

1. Na položce Maintenance Plans vybereme příkaz Maintenance Plan Wizard a klepneme na tlačítko next.
2. Zadáme název plánu a vybereme přepínač Separate scedules for each task a klepneme na tlačítko next.
3. Zaškrkneme políčka Update Statistics, Back Up Databáze(Full) a Maintenance Cleanup Task a klepneme na tlačítko next.
4. Vybereme databázi Navigator s možností ignorovat databáze, které nejsou ve stavu online. Vybereme přepínač All Existing Statistics a vybereme vzorkování po 25 procentech. Ještě musíme na této kartě nastavit plán na 3 A.M. a klepneme na tlačítko next.
5. Nyní zvolíme zálohování databáze Navigator na disk, určíme výchozí složku pro zálohování. Plán záloh nastavíme na každé 4 hodiny a klepneme na tlačítko Next.
6. Na výchozí složce pro zálohování musíme nastavit odstranění záložních souborů, které jsou starší než jeden týden. Nastavíme každodenní spouštění úlohy v 11P.M. a klepneme na tlačítko Finisch.[4]



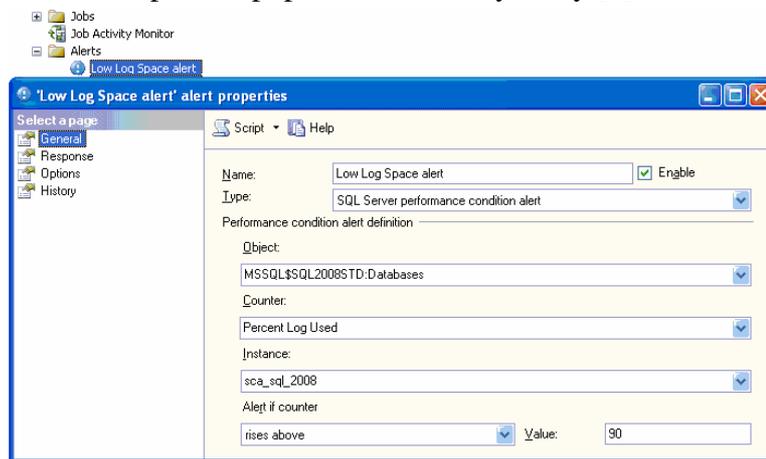
Obrázek č. 4: Grafický návrh Maintenance Plan databáze Navigator

3.11 Vytvoření výstrah v databázi sca_sql_2008

Úkolem Výstrahy (Alert) je odesílat oznámení nebo provádět akce na základě událostí nebo podmínek, které nastanou v instanci systému SQL Server. Můžeme pomocí nástroje Maintenance Plan Wizard nakonfigurovat jeden ze tří typů výstrah: událost systému SQL Server, podmínku výkonu a událost rozhraní WMI²⁹ (Windows Management Instrumentation). Výstraha pro událost systému SQL Server je vyvolávaná na základě čísla chyby nebo úrovně její závažnosti.

Nyní vytvoříme výstrahu, která bude zasílat oznámení, když transakční protokol databáze SCA_SQL_2008 bude zaplněn z více než 90 procent. Podmínky výkonu pro databázi SCA_SQL_2008, čítač Percent Log Used zabezpečíme tak, že nastavíme výstrahu pro situaci, kdy čítač překročí hodnotu 90.

1. Na položce Alerts vybereme příkaz New Alerts.
2. Zadáme název výstrahy a vybereme možnost SQL Server Performance Condition Alert.
3. Vybereme název databáze SCA_SQL_2008 a do pole Value zapíšeme hodnotu 90 pro nastavení podmínky výstrahy, když čítač překročí uvedenou hodnotu.
4. Pak přejdeme na stránku Response a vybereme možnost oznámení operátora, který obdrží emailovou zprávu s popisem události výstrahy.[4]



Obrázek č. 5: Grafický návrh výstrahy podmínky výkonu databáze SCA_SQL_2008

²⁹ WMI (Windows Management Instrumentation) je služba, která poskytuje integrovanou podporu datového modelu. WMI popisuje objekty existující v prostředí správy.

Implementaci neprocedurální části Nadstavbového modulu ERP systému pro výrobu a řízení SQL Server 2008 jsme si již představili a nyní zbývá představit si procedurální platformu Microsoft Access - Projekt APL_SCA(ADP). Projekt APL_SCA(ADP) je propojen s databází SQL Server, a tím tvoří aplikaci typu klient/server prostřednictvím OLE DB³⁰. Rozhraní projektu aplikace Microsoft Access poskytuje devět základních typů objektů aplikace. Tabulky, pohledy, uložené procedury a databázové diagramy patří ke katalogu SQL Serveru, kde jsou fyzicky uloženy. Formuláře, sestavy, datové stránky, makra a moduly jsou uloženy v souboru Microsoft Access - Projekt APL_SCA(ADP).

3.12 Formuláře Projektu APL_SCA(ADE)

Do uživatelského rozhraní Projektu APL_SCA(ADE) jsem implementoval objekty Nevázané formuláře, které nejsou propojeny s datovou částí, a Vázané formuláře, které jsou propojeny s datovou částí SQL Server. a jsou potřebné pro editaci, vkládání nebo odstranění dat zdrojových tabulek. Vázané formuláře byly využity k prohlížení základních dat nebo jejich filtrování, protože s jejich pomocí tvoří výkonný nástroj pro správu dat v uživatelském rozhraní .

3.12.1 Nevázané formuláře

Nevázaný formulář v modulu Docházka určuje v uživatelském rozhraní možnosti zadání Příchodu zaměstnance, Odchodu zaměstnance, Hromadné zadání docházky všech zaměstnanců, Denní přehled docházky, Výstupní sestavu nástrojů pro správu zadání čárovými kódy a Přehled odpracovaných hodin.

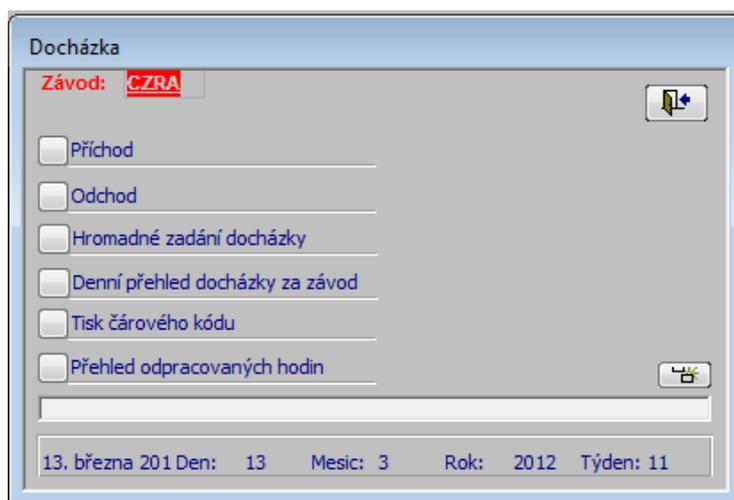
Poklepnutím na některé z tlačítek se spustí funkce, která uživatele směřuje do konkrétní větve programu, tedy spustí se funkce pro otevření příslušného vázaného formuláře. Další funkcí se otestuje, zda uživatel má oprávnění k přístupu do formuláře, a pokud je přístup povolen, spustí se poslední funkce, která přiřadí zdrojová data otevřeného formuláře.

³⁰ OLE DB OleDbConnection Objekt představuje jedinečné připojení ke zdroji dat. Databázový systém typu klient server je ekvivalentní síťové připojení k serveru.

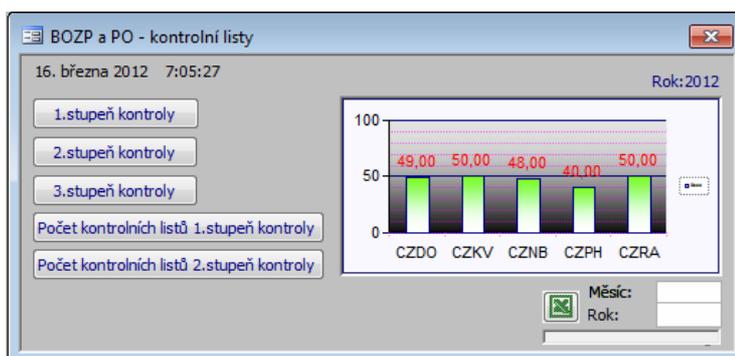
Obdobný scénář využití je i u všech nevázaných formulářů v Projektu APL_SCA(ADE), mimo jiné i formuláře BOZP a PO.

Úplná syntaxe kódu procedury `tl_dochazka_Click()` pro otevření formuláře `frm_Dochazka_hl` je:

```
Private Sub tl_dochazka_Click()  
On Error GoTo Err_tl_dochazka_Click  
  
    Dim stLinkCriteria As String  
  
    stDocName = "frm_Dochazka_hl"  
    DoCmd.OpenForm stDocName, , , stLinkCriteria  
  
Exit_tl_dochazka_Click:  
    Exit Sub  
  
Err_tl_dochazka_Click:  
    MsgBox Err.Description  
    Resume Exit_tl_dochazka_Click  
  
End Sub
```



Obrázek č. 6: Nevázaný formulář Docházka



Obrázek č. 7: Nevázaný formulář BOZP a PO

3.12.2 Vázané formuláře

Vázaný formulář frm_Reklamace v modulu Reklamace je propojen funkcí ke zdrojovým datům tabulky tbl_Reklamace, která je součástí databáze sca_sql_2008. Vázaný formulář Reklamace se využívá k evidenci zákaznických reklamací, dále pro potřeby tvorby analýz a nastavení nápravných opatření k eliminaci opakovaných reklamací.

Obrázek č. 8: Vázaný formulář Reklamace

3.12.2.1 Funkce formuláře Reklamace

Abychom mohli v elektronické podobě evidovat a archivovat korespondenci nebo dokumentaci, která souvisí s reklamacemi, je nutné mít k tomu účelu vhodný nástroj. Proto byla implementována funkce, která vytvoří nový adresář ve stromové struktuře na centrálním serveru s označením ID záznamu reklamace a následně se nabídne uživateli rozhraní, jehož pomocí bude moci vybrat soubor, který chce kopírovat do nově vytvořeného adresáře.



Obrázek č. 9: Detail tlačítka Přidat (tlc_add_dokument_Click()) a seznamu List_files

Syntaxe událostní procedury tlc_add_dokument_Click() je:

```
Sub tlc_add_dokument_Click()
  If IsNull(Me![Reklamace_Zakaznik]) Then
    MsgBox "      Zadej název zákazníka !      "
  Else
    On Error Resume Next
    MkDir31 "\\S0001902\grps\Apl_SCA\Kvalita\" & Reklamace_ID
    Me.Adresar = "\\S0001902\grps\Apl_SCA\Kvalita\" & Reklamace_ID
    Dim prepsat As Boolean, file_to_copy As String, separ_file_to_copy As
    Variant, dest_path As String
    On Error GoTo err_tlc_add_dokument_Click
    file_to_copy = GetFileName()32
    If Len33(file_to_copy) > 3 Then
      separ_file_to_copy = Split34(file_to_copy, "\")
      dest_path = Me("adresar").Value & "\" &
separ_file_to_copy(UBound35(separ_file_to_copy))
      prepsat = vbYes
      If Len(Dir36(dest_path)) > 0 Then
        prepsat = MsgBox("Nahradit soubor:" & Chr(13) & dest_path,
vbYesNo, "Nahradit soubor")
      End If
      If prepsat Then
        FileCopy file_to_copy, dest_path
      End If
    Else
      MsgBox "Není vybrán soubor"
    End If
  End If
End Sub
```

³¹ MkDir je instrukce tvořící nový adresář, v případě, že již adresář existuje, pak instrukce ignoruje opětovné vytvoření.

³² GetFileName je funkce, která vrací název souboru a příponu řetězce zadané cesty.

³³ Len je funkce, jež vrací hodnotu typu Long obsahující počet znaků v řetězci.

³⁴ Split je funkce, co vrací jednorozměrné pole počínající nulou, které obsahuje zadaný počet podřetězců.

³⁵ UBound je funkce, vrací hodnotu typu Long, která obsahuje nejvyšší dostupný dolní index pro uvedený rozměr pole.

³⁶ Dir je funkce, která vrací řetězec představující název souboru, adresáře nebo složky, jenž odpovídá zadanému vzorku, atributu souboru nebo jmenovce svazku jednotky.

```

End If
err_tlc_add_dokument_Click:
If Err <> 0 Then
    MsgBox Error
    Exit Sub
End If
actual_list_files37
End If
End Sub

```

List_files je událostní procedura, která vrací aktuální hodnoty do ovládacího prvku Seznam.

Syntaxe událostní procedury actual_list_files je:

```

Sub actual_list_files()
Dim file_filter As String
Dim file As String

Do Until Me("list_files").ListCount = 0
    Me("list_files").RemoveItem (0)
Loop

If Len(Me("Adresar").Value) > 0 Then

    'file_filter = Replace(Me("Adresar").Value & "\*.*", "\\\"", "\"")
    file_filter = Replace38(Me("Adresar").Value & "\*.*", "\\\"", "\"")

    file = Dir39(file_filter)

    Do Until Len40(file) = 0
        Me("list_files").AddItem file

        file = Dir
    End If
End Sub

```

Abychom mohli nabídnout komfortní uživatelské prostředí, bylo nutné do formulářů Projektu APL_SCA(ADE) implementovat ovládací prvky. Mimo jiné ovládací prvky, které byly využity v Projektu APL_SCA(ADE), bych rád představil Ovládací prvek Spreadsheet, Ovládací prvek Indikátor průběhu a Ovládací prvek Kalendář.

37 List_files je událostní procedura, vrací aktuální hodnoty do ovládacího prvku Seznam.

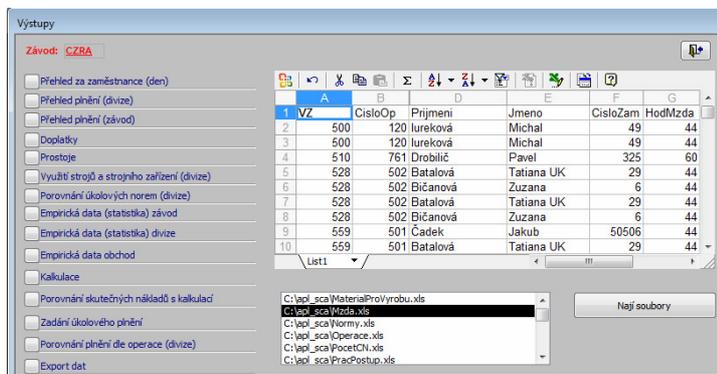
38 Funkce Replace nahradí část řetězce něčím jinými hodnotami.

39 Funkce Dir vrací jméno souboru nebo adresáře.

40 Funkce Len vrací délku řetězce nebo počet bajtů potřebných k uschování proměnné.

3.12.2.2 Ovládací prvek Spreadsheet

Pomocí ovládacího prvku Spreadsheet můžeme na formuláři spravovat kompletní tabulky Microsoft Excel. Ovládací prvek načte určitou oblast tabulky, provede změny a zapíše je zpět do tabulky.



Obrázek č. 10: Ovládací prvek Spreadsheet

Funkce pro načtení dat do ovládacího prvku Spreadsheet se skládá ze dvou částí. První část pomocí uložené procedury `tl_NacistSoubory_Click` načte seznam souborů s příponou `*.xls`, které jsou uloženy v adresáři `"c:\apl_sca\"` do Pole se seznamem a následně po dvojitém poklepnutí na text souboru se načtou hodnoty do ovládacího prvku Spreadsheet.

Syntaxe pro zajištění načtení souboru Excel událostní procedury `tl_NacistSoubory_Click()` je:

```
Private Sub tl_NacistSoubory_Click()
Dim varfile As Variant

With Application.FileSearch
    .NewSearch
    .LookIn = "c:\apl_sca\"
    .FileType = msoFileTypeExcelWorkbooks
    .SearchSubFolders = False

    If .Execute() > 0 Then
        For Each varfile In .FoundFiles
            Seznam.AddItem varfile
        Next varfile
    End If
End With
End Sub[5]
```

Syntaxe pro přenesení hodnot tabulky Excel do ovládacího prvku Spreadsheet událostní procedury Seznam_DblClick () je:

```
Private Sub Seznam_DblClick(Cancel As Integer)

    Dim intRadky As Long
    Dim Sloupce As Integer
    Dim xlApp As Object
    Dim xlSesit As Object
    Dim intZ As Integer
    Set xlApp = CreateObject41("Excel.Application")
    Set xlSesit = xlApp.Workbooks.Open42(Me.Seznam.Value)

    Me.Spreadsheet0.Cells.ClearContents43
    For intRadky = 1 To xlSesit.Sheets(1).UsedRange.Rows.Count44
        For intSloupce = 1 To xlSesit.Sheets(1).UsedRange45.Columns.Count
            Me.Spreadsheet0.Cells46(intRadky, intSloupce) = _
                xlSesit.Sheets(1).Cells(intRadky, intSloupce)
        Next intSloupce
    Next intRadky

    xlSesit.Close savechanges47:=False
    xlApp.Quit
    Set xlApp = Nothing48
    Set xlSesit = Nothing

End Sub[5]
```

⁴¹ CreateObject je funkce, vytvoří odkaz na knihovnu Microsoft Excel a následně získáme přístup ke všem metodám a vlastnostem, které jsou pro sešit k dispozici.

⁴² xlApp.Workbooks.Open je funkce, zapíše všechny soubory Excel do Ovládacího prvku Seznam.

⁴³ xlApp.Workbooks.Open je funkce, odstraní vzorce nebo taky odstraní data z grafů, ale zachová formátování.

⁴⁴ UsedRange.Rows.Count je funkce, vrátí číslo prvního řádku první oblasti, která je v dosahu. Jen pro čtení datového typu Long.

⁴⁵ UsedRange je funkce, vybere obsah buněk na listu Excel.

⁴⁶ Cells je buňka v listu Excel, nebo oblast buněk pro formát zápisu.

⁴⁷ Savechanges je makro, zavírá sešit Excel bez uložení změn.

⁴⁸ Nothing je funkce, vrací k objektu zpět prázdnou hodnotu.

3.12.2.3 Ovládací prvek Indikátor průběhu

V případě, že formulář obsahuje delší makra a uživatel potřebuje být seznámen s průběhem událostí, můžeme do formuláře implementovat ovládací prvek ProgressBar Control.



Obrázek č. 11: Ovládací prvek ProgressBar Kontrol

Syntaxe pro Indikátor průběhu událostní procedura ProgressBar ControlGo()je:

```
Private Sub ProgressBar ControlGo()  
    Dim intZ As Integer  
  
    With Me  
        .ProgressBar0.Min = 0  
        .ProgressBar0.Max = 10000  
  
        For intZ = ProgressBar0.Min To .ProgressBar0.Max  
            .ProgressBar0.Value = intZ  
            intZ = intZ + 1  
        Next intZ  
        .ProgressBar0.Value = 0  
    End With  
End Sub
```

Pomocí vlastností Min a Max můžeme určit počáteční nebo konečnou hodnotu ovládacího prvku. Pokud k vlastnostem Min a Max sestavíme smyčku, bude probíhat tak dlouho, dokud nebude dosaženo hodnoty Max.

3.12.2.4 Ovládací prvek Kalendář

Ovládací prvek Calendar Control 11.0 je pomocník pro rychlé zadání data, využívaný ve všech formulářích Projektu APL_SCA(ADE). Při poklepnutí myši se zobrazí ovládací prvek Calendar Control 11.0 a následně po klepnutí myši na pole s datem kalendáře se kalendář zavře a hodnota data se zapíše v krátkém formátu do požadovaného textového pole.



Obrázek č. 12: Ovládací prvek Calendar Control 11.0

Na základě potřeb uživatelů systémové správy dat rozhraní pro export dat z databáze SCA_SQL_2008 bylo celkem implementováno 213 vázaných nebo nevázaných formulářů do Projektu APL_SCA(ADE).

3.13 Sestavy Projektu APL_SCA.(ADP)

V Projektu apl_sca.adp byly vytvořeny sestavy pro potřebu tisku datových záznamů nebo pro potřebu exportu dat nezbytných pro další vyhodnocení.

Syntaxe pro otevření tiskové sestavy Reklamace událostní procedura tl_tisk_Click() je:

```
Private Sub tl_tisk_Click()
On Error GoTo Err_tl_tisk_Click

Sel_Reklamace PL (Me("Reklamace_ID").Value)

    Dim stDocName As String

    stDocName = "Rep_qryReklamace"
    DoCmd.OpenReport stDocName, acNormal

Exit_tl_tisk_Click:
    Exit Sub
Err_tl_tisk_Click:
    MsgBox Err.Description
    Resume Exit_tl_tisk_Click
End Sub
```

www.scapackaging.cz

Reklamační protokol č.2

Vystaveno: 27.4.2012 Stav: Uzavřeno
 Odpověď do: Stav dobropisu: NEPOŽADOVÁN

Zákazník		Zakázka	
Zákazník:	TELECOMMUNICA	Číslo zakázky:	1411284
Kontaktní osoba:	p.Muller	Produkt:	CZO000000038681
Tel:	602 371 371	Název produktu:	860*791*791
Fax:		Obj. množství:	3200
E-mail:	muller@drew.com	Výrobní zakázka:	
OZ:	Paláček	Dodací list:	81641309
Referent:	Fulek Karel	Datum dodání:	27.4.2012
		Dodané množství:	3200
		Rekl. množství:	358 Znamená: 100,00%
		Náklady celkem:	21 222,00 Kč

Popis reklamace		Reklamací vyřizuje	
Typ:	Reklamace	Výrobní závod:	CZRA
Kategorie:	Výsek	Oddělení / stroj:	Příklopový lis
Podkategorie:	Nedoseknuto	Zodpovědná osoba:	Fulek Karel
Oprávněnost reklamace:	ANO	Směna:	I směna
		Vyřídít do:	27.4.2012

Příčiny	Nápravná opatření

Vrácené zboží		Efektivita nápravných opatření	
Množství k vrácení:		Zavedeno dne:	27.4.2012
Počet palet k vrácení:		Zavedeni kým:	Karel Fulek
Zboží vrat do:		Ověřené dne:	27.4.2012
Místo nakládky:		Ověřeno kým:	Lekš Michal
Místo vykládky:			

27. dubna 2012

Page 1 of 1

Obrázek č. 13: Tisková sestava Reklamační protokol

Sestavu otevřeme pomocí metody OpenReport. Pro parametry data použijeme funkci Sel_Reklamacex_PL, která má již předdefinovanou strukturu parametru selektivního dotazu.

V případě potřeby můžeme pro export sestavy použít argumenty metody OutPutTo. Metodou OutPutTo můžeme zajistit, aby data objektu sestavy měla výstup v různých formátech.

Syntaxe metody OutPutTo je ObjectType, ObjectName, OutPutFormat, OutPutFile, AutoStart, Templatefile, přičemž vstupní argument OutPutFormat definuje, v jakém formátu se mají data přenést. Nejznámější formáty jsou acFormatHTML, acFormatRTF, acFormatTXT, acFormatXLS.[5]

Syntaxe pro otevření tiskové sestavy Cenové nabídky a export dat Cenové nabídky do formátu RTF událostní procedura TiskCN_Click() je:

```

Private Sub TiskCN_Click()
On Error GoTo Err_TiskCN_Click

Dim varText As Variant
Dim intObjednavky_ID As Integer
Dim stDocName As String

varText = "Cenová nabídka č." & Me("Objednavky_ID").Value
intObjednavky_ID = Me("Objednavky_ID").Value
Sel_PracPostx_Tisk intObjednavky_ID

DoCmd.DoMenuItem acFormBar, acRecordsMenu, 5, , acMenuVer70
DoCmd.RunCommand (acCmdSaveRecord)
stDocName = "Rep_CN_M"

DoCmd.OutputTo acOutputReport, _
"Rep_CN_M", acFormatRTF, "C:\apl_sca\Cenove_nabidky\" & varText &
".rtf", True

Exit_TiskCN_Click:
Exit Sub
Err_TiskCN_Click:
MsgBox Err.Description
Resume Exit_TiskCN_Click

End Sub

```

Cenová nabídka č.0001

SCA Packaging Česká republika, s.r.o.
Staré Město

Luční 1867 686 03 Staré Město
00 420 572 503 700
00 420 572 541 519
www.scapackaging.cz

Zákazník: Josef Popelka

Kontakt.osoba: Josef Popelka
Tel.: 608 999 999
Fax:
E-mail: info@fatrapark.cz

Číslo výrobku	Název výrobku	FEFCO	Materiál	Počet barev	Počet ks/pal	Množství	Prodejní cena	PN:
00001	152 x 450	0110	SVL HH FC 32 EB	0	1500	3000	0,08 Kč	2-42937
00002	255 x 368	0901	SVL HH FC 32 EB	0	1500	3000	0,92 Kč	7609657

- Uvedené ceny jsou BEZ dopravy.
- Uvedené ceny jsou bez DPH, které činí 20%.
- Cena platí pouze pro výše uvedené množství, po dobu jednoho měsíce.
- Pokud by se výrazně změnily ceny vstupů nebo bude objednávka uplatněna po této lhůtě, cena bude překalkulována.
- Je nutné počítat s uvedenými vícenásledky :
- Zboží je dodáváno na paletách: EURO-B (světla, použitá) 180,- Kč
- Platební podmínky - tak jako doposud/zálohová faktura/platba v hotovosti - budou předmětem dalšího jednání..
- Ostatní podmínky vyplývající z Obchodního styku se řídí Všeobecnými prodejními podmínkami č. 1101/SCA, které Vám budou zaslány spolu s kupní smlouvou.

Věříme, že budete s touto cenovou nabídkou spokojeni. V případě Vaší objednávky, uvádějte prosím číslo výrobku.
Děkují za Vaši poptávku a jsem s pozdravem.

Zpracoval : Michal Lekeš

27. dubna 2012

Obrázek č. 14: Export tiskové sestavy Cenová nabídka do souboru RTF

Ve výpisu, který deklaruje proceduru otevření tiskové sestavy a její následné převedení do formátu RTF, je znázorněno pořadí potřebných parametrů zadání metody OutputTo. Název sestavy "Rep_CN_M", výstupní formát acFormatRTF, název složky "C:\apl_sca\Cenove_nabidky\" a název souboru " & varText & ". VarText &" je v našem případě proměnná, která nese hodnotu záznamu ID.

Pro potřeby tisku nebo exportu dat z databáze SCA_SQL_2008 bylo vytvořeno v Projektu APL_SCA(ADP) 36 tiskových sestav.

3.14 Moduly Projektu ADP_SCA(ADP)

V Projektu ADP_SCA(ADP) byly implementovány pro řešení procedurální části Moduly třídy a Standardní moduly. Obecně modulem rozumíme samostatný databázový objekt, který je konstruován jako soubor textu programovacího jazyka Visual Basic.

Modul třídy obsahuje událostní procedury, pomocí nichž lze řídit chování formulářů nebo sestav podle požadavků uživatele.

Událost Private Sub Form_Current() ve formuláři frm_REKL_uzivatel vypíše hlášení v případě, že při procházení jednotlivých datových záznamů formuláře se dostaneme na poslední záznam.

Syntaxe kódu události Private Sub Form_Current() je:

```
Private Sub Form_Current()  
If Me.CurrentRecord <> 1 Then  
    If Me.RecordsetClone.RecordCount = Me.CurrentRecord Then  
        MsgBox "    Bylo dosaženo posledního záznamu!    ", vbInformation  
    DoCmd.GoToRecord acDataForm, "frm_REKL_uzivatel", acLast  
    End If  
End If  
End Sub
```

Tělo procedury tvoří příkazy událostní procedury Private Sub a End Sub.

Standardní modul deklaruje událostní procedury, které nejsou přidruženy k žádnému objektu. Nalezneme zde procedury, které se dají zploštit na libovolném místě Projektu ADP_SCA(ADP).

Výpis syntaxe kódu Standardního modulu M_fn, který obsahuje Proceduru Sub actual_aplBar(), která zabezpečuje po celou dobu spuštěného Projektu ADP_SCA(ADP) informaci o Loginu do Windows uživatele a středisku, ve kterém se uživatel právě nachází je:

```
Sub actual_aplBar()  
    CurrentProject.Properties("apptitle").Value = "SCA - [Uživatel: " &  
    Environ("username") & "]" - [Středisko: " & get_SEL_stredisko() & "]"  
    Application.RefreshTitleBar  
End Sub
```

Nyní na závěr popisu procedurální části Projektu ADP_SCA(ADP) se můžeme vrátit k doplnění funkce potřebné pro import dat z docházkového systému Watt, která je spouštěcí funkcí Událostní procedury sp_UPDT_tblDochazka.

Pomocí funkce import_lepenky_le() importujeme aktuální seznam identifikátorů materiálových systému SAP ze vzdáleného serveru (Link =\\S0001902\grps\Apl_SCA\SQL\SAP\IDOB\) do tabulky tblTMPlepen_LE.

Funkce import_lepenky_le() je bez argumentu a není součástí denního vykonávacího plánu v systému SQL Server.

Úplná syntaxe kódu VBA funkce import_lepenky_le() je:

```
Function import_lepenky_le()  
  
Dim adoDB_DBaseConn As New ADODB.Connection  
Dim adoDB_DBaseRecordset As New ADODB.Recordset  
  
Dim adoDB_SqlRecordset As New ADODB.Recordset  
Dim fie As ADODB.Field  
  
adoDB_DBaseConn.Open "Driver={Microsoft dBASE Driver (*.dbf)};" & _  
    "DriverID=277;" & _  
    "Dbq=\\S0001902\grps\Apl_SCA\SQL\SAP\IDOB\  
  
CurrentProject.Connection.Execute "delete from tblTMPLEPEN_LE"  
    adoDB_DBaseRecordset.Open "Select * From lepen_le.dbf",  
adoDB_DBaseConn  
    adoDB_SqlRecordset.Open "Select * From tblTMPlepen_LE",  
    CurrentProject.Connection, adOpenDynamic, adLockOptimistic  
  
Do Until adoDB_DBaseRecordset.EOF  
  
    adoDB_SqlRecordset.AddNew  
  
    For Each fie In adoDB_DBaseRecordset.Fields  
        adoDB_SqlRecordset.Fields(fie.Name).Value = fie.Value  
    Next  
  
    adoDB_SqlRecordset.Update  
  
    adoDB_DBaseRecordset.MoveNext  
Loop  
  
MsgBox "Import do tabulky IDOB_LEPENKY je dokončen !"  
  
End Function
```

3.15 Seznam modulů Projektu APL_SCA(ADP)

Vzhledem k rozdělení přístupových uživatelských práv jednotlivých zaměstnanců bylo nutné vytvořit oddíly, které byly pojmenovány dle pracovního zařazení nebo pracovních funkcí v úkolech samotných uživatelů.

- Evidence výrobních zakázek
- Evidence cenových nabídek
- Evidence materiálů
- Evidence výrobků
- Evidence setů
- Evidence litografií
- Evidence výsekových desek
- Evidence reklamací a stížností
- Evidence dodavatelských reklamací a stížností
- Evidence pracovních úrazů
- Evidence bezpečnosti práce a požární ochrany
- Evidence materiálových karet (IDOBů)
- Evidence ceníků
- Evidence norem - úkolového plnění, doplatků, prostojů, výkonů strojů a strojního zařízení
- Evidence dodavatelských objednávek
- Evidence strojů a strojního zařízení
- Evidence docházky zaměstnanců
- Historie výroby
- Fakturace
- Výstupy - empirická data, vyhodnocení divize, závodu, obchodu

3.15.1 Plánování a řízení výroby

Modul Plánování a řízení výroby tvoří spolu s moduly Zakázky a Sklad materiálové hospodářství. Procesy jsou do značné míry přizpůsobeny specifickým provozním požadavkům, zejména zohledňují různé výrobní možnosti sériové a velkosériové výroby. Systém umožňuje plánovat a řídit tok materiálu, poskytuje aktuální informace.

3.15.2 Výrobní kalkulace

Kalkulace nákladů výroby je zajištěna vazbami kmenových dat modulu Plánování a řízení výroby na modul Sklad, který je propojen se systémem SAP. Je tak zabezpečeno provádění efektivní a flexibilní stanovení nákladů na jednotlivé výrobky, které jsou dále exportovány do systému Navigator.

3.15.3 Plánování spotřeby materiálu

Na základě primárních potřeb jsou stanoveny sekundární požadavky pro jednotlivé výrobky nebo komponenty. Tvoří se seznamy pro možnost následného objednání v systému SAP.

3.15.4 Kapacitní bilance výroby

Tato skupina funkcí zajišťuje na všech úrovních plánování přehled o požadavcích na kapacity zdrojů (pracovišť) resp. jejich skupin tak, aby bylo možné zasahovat do průběhu jednotlivých výrobních zakázek a pružně řešit problematiku tzv. úzkých míst ve výrobě.

3.15.5 Dílenské řízení a evidence výroby

V této skupině úloh obsahuje výrobní zakázka údaje jako termíny, náklady a zdroje (tj. kapacity strojů a lidí, výrobní pomůcky a nástroje, materiál a dokumentaci). K dispozici jsou funkce pro postupné uvolňování výrobních zakázek s kontrolou disponibility materiálů, výrobních zdrojů a přípravků (výsekových desek, litografií), dále tisk výrobních dokladů pro zjednodušené zpětné hlášení a odvádění vyrobených produktů a polotovarů do skladů a v neposlední řadě také řešení zmetků vznikajících ve výrobním procesu. Evidence je zajištěna pomocí čteček čárových kódů

Informační systém výroby umožňuje analyzovat a monitorovat průběh výrobního procesu, rovněž sledovat stav výrobních zakázek a dodržování termínů.

3.15.6 Dodavatelské a odběratelské reklamace

Modul Reklamace slouží k posouzení problematiky reklamačního řízení ve všech jeho souvislostech. Slouží nejen k evidenci reklamací, ale umožňuje sledovat oběh reklamované položky od vazby na dodavatele, zakázku, dodací list, fakturu přes příjem/výdej reklamovaného zboží na reklamační sklad, převody na jiné závody Zpracovatelské divize a jeho odeslání a přijetí. Rovněž poskytuje informace o dodatečných nákladech a ztrátách vznikajících při reklamaci. Modul v sobě zahrnuje i užitečné přehledy a vyhodnocovací sestavy, jež slouží k analýze objemu a vývoje reklamací.

3.15.7 Docházka

Modul docházky umožňuje uživatelům zaznamenávat čas, který strávili prací na konkrétním projektu či příležitosti, a to i s popisem konkrétní vykonávané činnosti.

Eviduje dokonalý přehled o činnosti zaměstnanců a zároveň o průběhu práce. Díky záznamům v modulu Docházka resp. jeho struktuře čárových kódů je nenáročné evidovat úkolové plnění pracovníky.

3.15.8 Empirická data

Významný je modul zajišťující sběr, přenos, uchování, zpracování a prezentaci dat s cílem tvořit a poskytovat informace pro vyhodnocení výroby, obchodu, závodu, divize. Modul napomáhá při podpoře obchodu a marketingu, kdy na základě vizualizace nebo exportu dat znázorňuje úspěšnost zakázek a množství zpracovaných cenových zakázek i výrobních příkazů.

3.15.9 Bezpečnost a dokumentace (revize)

Modul pro bezpečnost umožňuje tvorbu různorodých seznamů. Do jeho obsahu náleží základní povinnosti zaměstnavatele v oblasti BOZP, dokumentace, školení zaměstnanců, pracovní úrazy, pracoviště a pracovní prostředí, podmínky používání strojů a technických zařízení, kontrola dodržování podmínek BOZP.

Mezi základní povinnosti zaměstnavatele v oblasti BOZP patří zpracování dokumentace, kategorizace jednotlivých prací z hlediska rizikových faktorů, způsoby ochrany zdraví zaměstnanců (závodní lékařská péče, lékařské prohlídky), školení v oblasti BOZP, potřeby spojené s používáním osobních ochranných pracovních prostředků, postupy evidence pracovních úrazů, potřeby pracovního prostředí výrobních a nevýrobních provozů, definice podmínek používání technických zařízení a strojů, rozdělení technického zařízení (elektrické, plynové, zdvihací, tlakové ...), kontroly dodržování podmínek BOZP.

4 IMPLEMENTACE PROJEKTU APL_SCA.ADP

Vývoj a implementace podnikového informačního systému znamená činnost, jejímž cílem je navrhnout ucelený systém sběru, zpracování a distribuce informací o organizaci výkonných funkcí, které nejlépe vyjadřují požadavky na objekty, případně soubory objektů určitého typu (tabulky, dotazy, funkce, procedury, diagramy, pohledy, formuláře, sestavy nebo moduly).

První část vývoje Projektu APL_SCA(ADP) proběhla ve specifickém okruhu lidí. S budoucími uživateli byly detailně prodiskutovány jejich představy a vytvořen model vizualizace uživatelského rozhraní. V návaznosti na zjištěné informace bylo nutné si ujasnit strukturu dat, seznámit se s požadovanými výstupy do systému Navigator a navrhnout strukturu nové databáze SCA_SQL_2008.

Bylo nezbytné, aby systém splňoval určité standardy, které zajistí vyžadovanou úroveň kvality a proto byl vytvořen tým lidí, který měl zabezpečit podporu a nastavení procesů potřebných pro první úroveň testování, abychom odladili nedostatky v podnikovém informačním systému LEKY před samotným spuštěním. Což odstranilo převážné množství vzniklých chyb a zaručilo úspěšný průběh v samotném provozu. Testování před implementací proběhlo ve dvou fázích testování, kdy v první fázi kvalita testování závisela především na jednotlivcích týmu, s jakou zodpovědností se časově náročného úkolu zhostí. Po ukončení první fáze testování, instalaci a nastavení přístupových práv do podnikového informačního systému LEKY jednotlivým uživatelům jsme započali druhou část testování. Protože se jedná o poměrně náročný proces kontroly, zvolili jsme paralelní provoz zadávání dat všech uživatelů. Pro samotné uživatele to znamená duplicitní zadávání dat, ale pouze takto jsme schopni odladit zbytek systémových chyb. Abychom využili testování, stanovili jsme plán testu na dobu jednoho měsíce s následným překlopením zkušebních dat do ostré verze systému. Paralelně s testováním běžel plán proškolení a seznámení všech uživatelů výrobního závodu s novým uživatelským prostředím. Tento způsob implementace jsme použili ve všech závodech Zpracovatelské divize.

Vzhledem k tomu, že jsme s jednotlivými pracovišti byli v každodenním kontaktu a vzájemně jsme probírali jednotlivé fáze vývoje podnikového informačního systému LEKY, proběhl nespočet testů zadání, při kterých jsme většinu chyb odladili, a proto jsme nenašli v testu žádná zásadní pochybení.

5 ZÁVĚR

Cílem mé diplomové práce bylo vytvořit výrobní modul, který bude splňovat požadavky pro export dat do databáze Navigator, a taktéž sjednotí kalkulační vzorce ve všech závodech Zpracovatelské divize. Dále bude obsahovat požadované funkce, které usnadní náročné výpočty s průhledností vkládaných údajů, zachová vypovídající schopnost a výrazně zjednoduší zadávání dat s omezením duplicit. Jeho vysoký stupeň variability, tvorba technologického postupu se odvíjí od zvolené míry podrobností při zadávání pracovních postupů a operací, které umožní všechny dříve vytvořené postupy doplňovat.

Systém LEKY podporuje všechny druhy výpočtů konstrukcí FEFCO a je vhodný pro zakázky sériového charakteru. Pro zakázkovou výrobu jsou vytvořeny výrobní příkazy bez předem stanovených technologických postupů. Postupy jsou tvořeny přímo na průvodkách včetně požadovaného výrobního materiálu. U opakované výroby jsou průvodky vytvořeny automaticky spolu s vytvořením průvodek pro upřesnění technologického postupu.

Systém LEKY zahrnuje nejen nástroje výrobního charakteru nebo nástroje pro tvorbu podrobných analýz, ale i také veškeré součásti potřebné pro řízení podniku v dnešní rozsáhlé legislativě. Zejména je kladen důraz na nástroje obsahující prvky zvyšující bezpečnost práce a funkce pro kontrolu platnosti revizí strojů a strojního zařízení. Všechny funkce byly implementovány tak, abychom se mohli rychle a přehledně orientovat ve velkém množství dat s možností snadno v nich provádět změny.

Předpokládám, že implementovaný systém splnil očekávané požadavky, zprůhlednil datový tok i technologické postupy, uspořádal a zprůhlednil výrobní dokumentaci, optimalizoval počty zaměstnanců ve všech odděleních, snížil odpadovost, zvýšil výkony úkolového plnění, optimalizoval využití strojů a usnadnil zadání výrobní dokumentace pomocí čárových kódů ve všech závodech Zpracovatelské divize.

Při vyvíjení systému LEKY jsem využil zkušeností a připomínek mých kolegů, kteří významně napomohli zajistit posloupnost a návaznost jednotlivých technologických procesů. Taktéž mi byli významnou pomocí při testování funkcí a procedur, kdy celý tým testoval systém na více úrovních, aby bylo možné jednoznačně zkontrolovat jejich funkčnost.

Lze se domnívat, že předložený projekt splnil kritéria zadání, neboť praktická část mé diplomové práce je již plně využita v praxi. Systém LEKY napomáhá již dva měsíce v závodech Zpracovatelské divize společnosti SCA Packaging s.r.o. Vzhledem k tomu, že se osvědčil na všech odděleních výrobních závodů, se vedení společnosti rozhodlo rozšířit software o další modul řízení výroby pomocí pohybových a infračervených senzorů. Doufám, že nově požadovaný modul bude stejně účinným příspěvkem pro lepší a efektivnější práci v dalších odděleních.

ZÁVĚR V ANGLIČTINĚ

The objective of this thesis was to create production module that will meet the requirements for data export to database Navigator and that will also unite the calculation formulas in all the factories of the manufacturing division. It will also contain required functions that will make difficult calculations easier and the imported data transparent, preserve adequate efficiency and will make entering of the data easier with limitation of duplications. Its high level of variability and creating of technological methods depend on the degree of given details when entering the working procedures and operations. It will be possible to add data to the previously created methods.

System LEKY supports all kinds of calculations of FEFCO and it is suitable for orders of serial production. For custom-made production there are created production orders without predefined technological methods. The methods are created straight on the dispatch forms including the required material. When the production is repeated the dispatch forms are created automatically together with creation of dispatch forms for specification of the technological method.

System LEKY consists not only of tools for production or tools for creating of detailed analyses but also of all components necessary for company management in today's complex legislation. Stress is put on tools containing features increasing safety at work and functions for checking of inspection validity for machines and machinery. All the functions were implemented in the way that we are able to be well informed about big amount of data and make changes easily.

I suppose that the implemented system met the requirements, made the data stream and technological methods transparent, set the production documentation in order, optimized the number of employees in all departments, reduced waste, increased results of task work, optimized use of machinery and made entering of production documentation easier by means of barcodes in all the factories of the manufacturing division.

When developing the system LEKY, I used experience and comments of my colleagues who significantly assisted to get sequence and continuity of particular technological methods. They also helped me with testing of functions and procedures; the whole team tested the system at different levels so that the functionality of these levels could be clearly checked.

I guess that the presented project met the criteria of the task because the practical part of my thesis has been put into practice. System LEKY has been used in factories of manufacturing division of company SCA Packaging s.r.o. for two months. Based on the fact that the system proved its worth in all departments of production factories, the leadership of the company decided to extend the software and add another module – module for control of production by means of motion detectors and infrared sensors. I believe that the newly requested module will be effective at better and more efficient work in other departments.

SEZNAM POUŽITÉ LITERATURY

Knihy:

- [1] LACKO, Ľuboslav. *Jak vyzrát na Microsoft SQL Server 2008: správa, konfigurace, programování*. Vyd. 1. Brno: Computer Press, 2009, s. 472. ISBN 978-80-251-2101-6.
- [2] J.BRUST, Andrew ; FORTE , Stephen. *Mistrovství v programování SQL Serveru 2005*. 639 00 Brno : COMPUTER PRESS, 26.11.2007. 848 s. ISBN 978-80-251-1607-4, K1443.
- [3] E.WALTERS, Robert ; COLES, Michael; RAE, Robert. *Mistrovství v programování Microsoft SQL Serveru 2008 : Komplexní průvodce databázového experta*. 639 00 : COMPUTER PRESS, 2009. 864 s. ISBN 978-80-251-2329-4, K1660.SS
- [4] HOTEK, Mike. *Microsoft SQL Server 2008 : Krok za krokem*. 639 00 Brno: COMPUTER PRESS, 08.06.2009. 488 s. ISBN 978-80-251-2466-6, K1686.
- [5] HELD, Bernd. *Access VBA*. 639 00 Brno : : COMPUTER PRESS, 2006. 639 s. ISBN 80-251-1112-1, K1258.
- [6] FEDDEMA, Helen. *Mistrovství v Microsoft Access 2002*. 639 00 Brno : COMPUTER PRESS, 2002. 604 s. ISBN 80-7226-725-6, K0595.
- [7] MORKEŠ, David. *Microsoft Office Access 2003 Podrobná uživatelská příručka*. 639 00 Brno : COMPUTER PRESS, 2004. 350 s. ISBN 80-251-0179-7, K0977.
- [8] FUCHS, Joachim a BARCHFELD, Andreas. *Visual Basic: velká kniha řešení*. Vyd. 1. Brno: Computer Press, 2010. 722, xxii s. Programování. ISBN 978-80-251-2212-9.
- [9] KOCICH, Pavel a SPILKA, Ondřej. *1001 tipů a triků pro Microsoft Visual Basic*. Vyd. 1. Brno: Computer Press, 2010. 520 s. ISBN 978-80-251-2118-4.SSS
- [10] MEIER, Marco, SINZIG, Werner a MERTENS, Peter. *Enterprise management with SAP SEM business analytics*. 2nd ed. Berlin: Springer, 2005. xi, 219 s. SAP excellence. ISBN 3-540-22806-3.

Internetové zdroje:

- [11] Knihovna MSDN [online]. [cit. 2012-03-15]. Dostupné zWWW: [http://msdn.microsoft.com/cs-cz/library/bb677243\(v=sql.100\).aspx](http://msdn.microsoft.com/cs-cz/library/bb677243(v=sql.100).aspx)
- [12] Knihovna MSDN [online]. [cit. 2012-03-16]. Dostupné zWWW: [http://msdn.microsoft.com/en-us/library/aa174792\(v=sql.80\).aspx](http://msdn.microsoft.com/en-us/library/aa174792(v=sql.80).aspx)
- [13] Business info – Oficiální portál ERP Systému SAP Česká republika: Seznam modulů systému SAP [online]. [cit. 2012-04-20]. Dostupné zWWW: http://www.sap.com/cz/campaign/2009_07_CROSS_ERP_GENERAL/index.epx?campaigncode=CRM-CZ12-RDC-PPC_ABA_01&dna=117800,8,0,95426597,806212279,1335264139,Moduly
- [14] KRISHNASWAMY, Jayaram. Learning SQL server 2008 reporting services : a step-by-step guide to getting the most of Microsoft SQL server reporting services

2008 [online]. Birmingham, U.K.: Packt Pub., 2009 [cit. 2012-03-21]. Dostupné zWWW: <<http://site.ebrary.com/lib/natl/Doc?id=10442957>>.

[15] FNIELSEN, Paul. Microsoft SQL server 2008 bible [online]. Indianapolis, IN: Wiley, ©2009 [cit. 2012-03-21]. Dostupné zWWW: <<http://site.ebrary.com/lib/natl/Doc?id=10330068>>.

[16] Knihovna MSDN [online]. [cit. 2012-03-16]. Dostupné zWWW: <http://msdn.microsoft.com/en-us/library/ms187752.aspx>

Články:

[17] HÉGR, Michael. APS systém nenahradí funkcionalitu ERP systému. IT Systems, 2010, 12(10), s. 18-19. ISSN 1802-002X.

[18] KOREJS, Martin. Kvalitní ERP systém se dokáže přizpůsobit. IT Systems, 2011, 13(3), s. 28-29. ISSN 1802-002X.

[19] PETRJANOŠ, Vít. ERP systémy pro malé a střední podniky: Podaří se odblokovat trh?. Computerworld, 2011, 22(20), s. 13-20. ISSN 1210-9924.

[20] NOVÁK, Daniel. Technologie v pozadí vývoje informačních systémů ERP. Computerworld, 2011, 22(15), s. 26-27. ISSN 1210-9924.

[21] KLČOVÁ, Hana a SODOMKA, Petr. Představujeme ERP systémy na českém trhu: QI - systém s moderní technologickou koncepcí. IT Systems, 2009, 11(9), s. 18-21. ISSN 1802-002X.

[22] BEREZIN, Vladlen. Podobné jak vejce vejci: jak vybrat ERP systém, aby dobře sloužil. Connect!, 2008, 13(9), s. 20-22. ISSN 1211-3085.

[23] ZETLINOVÁ, Minda. Nová pravidla pro podnikové aplikace. Computerworld, 2012, 23(8), s. 22-26. ISSN 1210-9924.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

- ADE Soubor, programové vybavení generující projekt, který tvoří uživatelské rozhraní.
- DDL Jazyk obsahující příkazy pro vytváření struktury databáze.
- DML Jazyk obsahující příkazy manipulující s daty v databázi.
- ERP Informační systém, který integruje a automatizuje velké množství procesů souvisejících s produkčními činnostmi podniku.
- FEFCO Mezinárodní názvosloví konstrukce vlnitých lepenek.
- IDOB 8-ti znakový textový řetězec jehož kombinace čísel a textu na různých pozicích vyjadřuje jedinečnou identifikaci kódu skladových karet.
- NAS Pevný disk nebo skupina pevných disků, které jsou připojeny k lokální síti.
- OLAP Jednorázové nahrávání dat, nad kterými jsou prováděny složité dotazy.
- OLTP Technologie pro uložení dat v databázi.
- DDL Jazyk obsahující příkazy pro vytváření struktury databáze.
- DML Jazyk obsahující příkazy manipulující s daty v databázi..
- RAM Typ elektronické paměti, která umožňuje přístup k libovolné části v konstantním čase bez ohledu na její fyzické umístění..
- ADE Soubor, programové vybavení generující projekt, který tvoří uživatelské rozhraní.
- SQL Standardizovaný dotazovací neprocuderální jazyk používaný pro práci s daty v relačních databázích.
- SSIS Platforma pro integraci dat vybavena rychlou a flexibilní infrastrukturou zabezpečující extrakt, transformaci a načtení dat.
- T-SQL Proprietární rozšíření využívaný pro psaní kódu v produktu Microsoft SQL Server.
- VBA Událostmi řízený procedurální programovací jazyk používaný pro tvorbu aplikací.

WMI Služba, která poskytuje integrovanou podporu datového modelu.

SEZNAM OBRÁZKŮ

Obrázek č. 1: Relační schéma systému Navigator	45
Obrázek č. 2: Soubor Dochazka.txt, sekvence textových znaků	46
Obrázek č. 3: Uživatelské rozhraní transakce YN0DCOZIG2	49
Obrázek č. 4: Grafický návrh Maintenance Plan databáze Navigator	56
Obrázek č. 5: Grafický návrh výstrahy podmínky výkonu databáze SCA_SQL_2008	57
Obrázek č. 6: Nevázaný formulář Docházka	59
Obrázek č. 7: Nevázaný formulář BOZP a PO	59
Obrázek č. 8: Vázaný formulář Reklamace	60
Obrázek č. 9: Detail tlačítka Přidat (tlc_add_dokument_Click()) a seznamu List_files	61
Obrázek č. 10: Ovládací prvek Spreadsheet	63
Obrázek č. 11: Ovládací prvek ProgressBar Kontrol.....	65
Obrázek č. 12: Ovládací prvek Calendar Control 11.0.....	66
Obrázek č. 13: Tisková sestava Reklamační protokol	67
Obrázek č. 14: Export tiskové sestavy Cenová nabídka do souboru RTF.....	68

SEZNAM TABULEK

Tabulka č. 1: Popis číselných datových typů[11]	20
Tabulka č. 2: Popis znakových datových typů[11]	20
Tabulka č. 3: Popis data a časových datových typů[11]	21
Tabulka č. 4: Popis binárních datových typů[11]	21
Tabulka č. 5: Popis datových typů tabulky tblDochazka.....	47
Tabulka č. 6: Popis datových typů tabulky tblMzdovaSlozka.....	47
Tabulka č. 7: Popis datových typů tabulky tblOBALY_LE	50
Tabulka č. 8: Seznam tabulek databáze SCA_SQL_2008.....	86
Tabulka č. 9: Seznam pohledů databáze SCA_SQL_2008.....	86
Tabulka č. 10: Seznam uložených procedur databáze SCA_SQL_2008	87
Tabulka č. 11: Seznam formulářů Projektu APL_SCA(ADP)	88
Tabulka č. 12: Seznam tiskových sestav Projektu APL_SCA(ADP).....	89

SEZNAM PŘÍLOH

P I: OBJEKTY NADSTAVBOVÉHO MODULU ERP SYSTÉMU PRO VÝROBU
A ŘÍZENÍ

PŘÍLOHA P I: OBJEKTY NADSTAVBOVÉHO MODULU ERP SYSTÉMU PRO VÝROBU A ŘÍZENÍ

Tabulka č. 8: Seznam tabulek databáze SCA_SQL_2008

Název tabulky	Datová pole tabulky (počet)	Název tabulky	Datová pole tabulky (počet)	Název tabulky	Datová pole tabulky (počet)
D_Board_Grade	15	TblDruhPrace	2	tblPRDCY	3
D_Customer	60	TblDruhUrazu	2	tblPrepocet	3
D_Customer_Product	4	TblDruhZraneni	2	tblProstoj	3
D_Customer_Product	43	TblEUSE	3	tblPST	3
D_Product	110	TblFakturace	60	tblRefCostElement	4
D_Product_Line	2	TblFEFCO	53	tblReklamace	129
D_Product_Type	2	TblFLC	2	tblReklamaceKat	5
D_Replenishment_Order	6	TblHistorie	7	tblReklamaceOdd	4
F_Delivery_Note	3	TblChyby	2	tblReklamaceNak	20
F_Invoice	4	TblIdob	29	tblReklamaceNakTyp	3
F_Order	33	TblKurz	3	tblReklamaceNav	8
F_Order_PCC_Cost_Breakdown	17	tblLEPEN_LE	29	tblReklamaceStav	4
tblADMIN	2	TblLide	20	tblReklamaceStroj	4
tblAktualizace	4	TblLimit	6	tblReklamaceUz	7
tblAktualizace	2	TblLitografie	23	tblReklamaceZav	7
tblART_Pallet	2	TblMAC	2	tblREP	6
tblART_Status	2	TblMACG	2	tblRevize	8
tblASS	2	TblMaterial	52	tblRevizeSel	2
tblAtribut	2	tblMatOst	14	tblRozpisDod	22
tblBOZP	46	tblMatOstSez	2	tblRozpisObj	22
tblBRDLI	2	tblMatSpec	22	tblSA	3
tblCastTela	2	tblMKT	2	tblSAPOZ	4
tblCostElements	6	tblMzdovaSlozka	3	tblSety	9
tblCountry	3	tblNakladyNaZakazku	2	tblSklad	9
tblCus_Status	3	tblNorma	20	tblSSTPR	4
tblCUSG	2	tblOBALY_LE	17	tblTURNC	2
tblCUSG	15	tblObjednavka	22	tblURAZY	52
tblDodavatel	2	tblObjekt	3	tblVADA	3
tblDodavatel	17	tblOperace	30	tblVLNA	3
tblDodObj	22	tblOZ	3	tblVyrobyFefco	96
tblDodReklamace	117	tblPalety	2	tblVysekDesky	35
tblDodReklamaceNak	11	tblPlanovani	8	tblZakazkySAP	4
tblDocházka	7	tblPlans	3	tblZakaznik	19
tblDodReklamaceNav	8	tblPobocka	10	tblZakaznikTel	13
tblDochazkaDate	4	tblPodminka	4	tblZasobaZast	7
tblDochazkaSCA	10	tblPojistovna	7	tblZustAPLIn	26
tblDochazkaZam	5	tblPoznamka	4		
tblDoplatky	3	tblPracPostup	367		

Tabulka č. 9: Seznam pohledů databáze SCA_SQL_2008

Název pohledu	Datová pole pohledu (počet)	Název pohledu	Datová pole pohledu (počet)	Název pohledu	Datová pole pohledu (počet)
qry BOZP_Skladovani_text	4	qryKalkulace_hlSum	34	qryReklamace_Nap	4
qry BOZP_Uklid_text	4	qryKalkulace_hlZ	12	qryReklamace_prod	8
QryAdmin	4	qryKalkulace_ObjMat	21	qryReklamace_sum	2
qryArt_type	2	qryLide	18	qryREP	5
QryAss	2	qryLimit	6	qryRevize	2
QryBOZP	4	qryLimit_ex	7	qrySA	3

qryBOZP_DodrzBezPred_text	4	qryLimit_xeD	5	qrySAP_Oz	4
qryBOZP_DodrzPracPost_text	4	qryMATERIAL_ned	12	qrySklad_mat	22
qryBOZP_Komunikace_text	4	qryMaterial_sum	3	qrySumOZ	5
qryBOZP_Kryty_text	4	qryMKT	2	qryTURNC	2
qryBOZP_OchrPomc_text	4	qryMzda	11	qryVyr_VD	8
qryBOZP_Ostatni_text	4	qryNavigator_fa	7	qryVyroba_hl	12
qryBOZP_Poradek_text	4	qryNavigator_fa_r	6	qryVyroba_ku	17
qryBOZP_Pristup_text	4	qryNavigator_SAP_CO	2	qryVyroba_ku_Null	6
qryBOZP_Zajisteni_text	4	qryNavigator_SAP_hl	28	qryVyroba_kum	10
qryCost_Element	4	qryNorma	14	qryVyroba_list	24
qryCost_ElementVP	6	qryNorma_dochazka	6	qryVyroba_ObPI	12
qryCountry	3	qryNorma_dopl	11	qryVyroba_pl	27
qryCus_Status	3	qryNorma_hl	14	qryVyroba_pro	14
qryCUSG	2	qryNorma_prac	11	qryVyroba_roz	21
qryD_Customer_Dupl	3	qryNorma_sel	11	qryVyroba_rozOp	28
qryD_CustomerIsNull	7	qryNorma_selDiv	8	qryVyroba_rozOpU	24
qryD_F_OrderIsNull	7	qryNorma_selOp	10	qryZakaznik	8
qryD_Product_first	2	qryNorma_SortZamD	14	qryZasoba_hl	8
qryD_ProductIsNull	8	qryOperace	21	qryZasoba_kon	13
qryDLV_LOC	14	qryOstatni	4	qryZasoba_konA	15
qryDochazkaZam	3	qryPlanovani	7	qryZasoba_konSA_it	12
qryEUSE	3	qryPocetCN	4	qryZasoba_konZ	13
qryF_D_Order	35	qryPozadavek	7	qryZasoba_konZ_lit	13
qryF_Order	4	qryPozadavekV	13	qryZasoba_konZ_li	14
qryFakturace	6	qryPrepocet	3	qryZasoba_rez	8
qryKalkulace_cost	7	qryRekalkulace	6	qryZustAPLinA	7
qryKalkulace_costZ	8	qryReklamace	27	sysque_LIST_rekla	10
qryKalkulace_hl	11	qryReklamace_Nak_s	5		

Tabulka č. 10: Seznam uložených procedur databáze SCA_SQL_2008

Název procedury	Parametry (počet)	Název procedury	Parametry (počet)	Název procedury	Parametry (počet)
sp_DELETE_ReklNak_sel	1	sp_INS_URAZY_T_op	1	sp_INS_Vyrobek_plP	3
sp_INS_aaa_Correct	0	sp_INS_Vyroba	2	sp_INS_Vyrobek_plV	3
sp_INS_BOZP	2	sp_INS_Vyroba_A	2	sp_INS_Vyrobek_pr	1
sp_INS_CN_new	2	sp_INS_Vyroba_T_op	3	sp_INS_vyrobek_update	1
sp_INS_CN_sel	3	sp_INS_Vyroba_T_opA	2	sp_INS_vyrobek_updateA	2
sp_INS_DodReklamace_new	2	sp_INS_Vyroba_T_opB	1	sp_INS_VZ_cost	1
sp_INS_Form_no	1	sp_INS_Vyroba_T_pal	1	sp_INS_VZ_sel	3
sp_INS_IDOB	1	sp_INS_Vyroba_T_sk	1	sp_INS_Zakazka_new	2
sp_INS_IDOB_akt	0	sp_INS_Vyroba_T_vp	2	sp_INS_Zakaznik_new	2
sp_INS_IDOB_v	1	sp_INS_Vyroba_T_vpA	1	sp_INS_Zasoba_list	1
sp_INS_Material	2	sp_INS_Vyroba_uk	1	sp_INS_Zasoba_sel	0
sp_INS_Objednano	1	sp_INS_Vyroba_uk_sel	1	sp_LIST_szn	1
sp_INS_Pozadavek_new	5	sp_INS_Vyrobek_ce	2	sp_UPDT_Board	0
sp_INS_PracPostup_copy	1	sp_INS_vyrobek_copy	2	sp_UPDT_D_Customer	0
sp_INS_Reklamace_new	2	sp_INS_vyrobek_copyA	2	sp_UPDT_DochazkaLide	0
sp_INS_Revize_new	2	sp_INS_Vyrobek_do	2	sp_UPDT_DochazkaZam	0
sp_INS_sum_tblOperace	5	sp_INS_Vyrobek_hl	2	sp_UPDT_Fakturace	0
sp_INS_sum_tblPracPostup	5	sp_INS_Vyrobek_idob	2	sp_UPDT_IDOB	0
sp_INS_tblCustProd	0	sp_INS_vyrobek_new	2	sp_UPDT_Nav01	0
sp_INS_tblTMPZustAPLin	8	sp_INS_Vyrobek_os	2	sp_UPDT_Nav02	0
sp_INS_tblZustAPLin	0	sp_INS_Vyrobek_pal	2	sp_UPDT_Norma_sk	0
sp_INS_tblZustAPLinA	0	sp_INS_Vyrobek_palT	2	sp_UPDT_Plans	0
sp_INS_tblZustAPLinB	0	sp_INS_Vyrobek_pl	3	sp_UPDT_SapOz	0
sp_INS_tblZustAPLinx	0	sp_INS_Vyrobek_plK	3	sp_UPDT_tblDochazka	0
sp_INS_TISK_CN	1	sp_INS_Vyrobek_plL	3	sp_UPDT_Vyroba	0
sp_INS_UPDATE_IS_NULL	0	sp_INS_Vyrobek_plO	3		

Tabulka č. 11: Seznam formulářů Projektu APL_SCA(ADP)

Název formuláře	Název formuláře	Název formuláře	Název formuláře
frm_BOZP_gr	frm_LITOGRAFIE_subform	frm_REVIZE_hl	frm_Vyroba_zadCarKod
frm_BOZP_hl	frm_MATERIAL_DOD	frm_REVIZE_sel	frm_VYROBEK_car
frm_BOZP_hl_pr	frm_MATERIAL_DOD_subform	frm_REVIZE_subform	frm_VYROBEK_ce
frm_BOZP_hIA	frm_MATERIAL_hl	frm_REVIZE_subformVyr	frm_VYROBEK_do
frm_BOZP_pr	frm_MATERIAL_OBJ	frm_RozpisDod_subform	frm_VYROBEK_dod
frm_BOZP_roz	frm_MATERIAL_obj_saz_subform	frm_RozpisObj_subform	frm_VYROBEK_hl
frm_BOZPT_hl	frm_MATERIAL_Obj_subform	frm_RozpisVyr_subform	frm_VYROBEK_hlx
frm_BOZPT_hl_pr	frm_MATERIAL_Poz	frm_SEL_hl_aktual	frm_VYROBEK_idob
frm_BOZPT_pr	frm_Material_pozV	frm_SEL_hl_tools	frm_VYROBEK_idob_subform
frm_CENIK_hl	frm_Material_pozV_subform	frm_SEL_stredisko	frm_VYROBEK_lit
frm_CENIK_Subform	frm_MATERIAL_sel	frm_SEL_uvod	frm_VYROBEK_mat
frm_CN_det	frm_MATERIAL_subform_ts	frm_SEL_vysekdesky	frm_VYROBEK_nak
frm_CN_HL	frm_NORMY_prac_subform	frm_SETY_sel	frm_VYROBEK_pal
frm_CN_roz	frm_NORMY_cn_oz_subform	frm_SETY_subform	frm_VYROBEK_pr
frm_CN_roz_subform	frm_NORMY_cn_oz_subform_graf	frm_SETY_vyr	frm_VYROBEK_prac
frm_CN_seznam	frm_NORMY_cn_oz_subform_grafV	frm_SETY_vyr_subform	frm_VYROBEK_sled
frm_CNCus_HL	frm_NORMY_cn_oz_subformG	frm_SKLAD_sel	frm_VYROBEK_stav
frm_CNCus_subform	frm_NORMY_cn_oz_subformG1	frm_SKLAD_vy	frm_VYROBEK_stav_subformN
frm_CNZ_HL	frm_NORMY_cn_subform	frm_SKLAD_vy_subform	frm_VYROBEK_subform_Cosls
frm_CNZ_subform	frm_NORMY_dat	frm_TMPCOPY	frm_VYROBEK_subform_Fakur
frm_Dodavatel	frm_NORMY_dop	frm_URAZY_hl	frm_VYROBEK_subform_Lid
frm_DodObj	frm_NORMY_kum_subform	frm_URAZY_hl_zal	frm_VYROBEK_subform_Lit
frm_DodReklamacie_hl	frm_NORMY_kumZ	frm_URAZY_pr	frm_VYROBEK_subform_Mat
frm_DodReklamacie_sel	frm_NORMY_kumZ	frm_VYPFEFCO_hl	frm_VYROBEK_subform_Matr
frm_DodReklamacie_sel_subform	frm_Normy_op_subform	frm_VYROBA_do_subform	frm_VYROBEK_subform_Mate
frm_DodReklamacie_subform	frm_NORMY_Por	frm_VYROBA_hl	frm_VYROBEK_subform_PraP
frm_Dochazka_hl	frm_NORMY_pro	frm_VYROBA_hl_subform	frm_VYROBEK_subform_PraL
frm_Dochazka_hl_CK	frm_NORMY_pro_subform	frm_VYROBA_kum_subform	frm_VYROBEK_subform_Pras
frm_Dochazka_hl_CK_subform	frm_NORMY_sel	frm_VYROBA_no	frm_VYROBEK_subform_Preo
frm_Dochazka_hl_D	frm_NORMY_stat	frm_VYROBA_no_subform	frm_VYROBEK_subform_SST
frm_Dochazka_hl_Kon	frm_NORMY_stat_subform	frm_VYROBA_ObPl_subform	frm_VYROBEK_subform_TISK
frm_Dochazka_hl_Kon_subform	frm_NORMY_stat0	frm_VYROBA_op_subform	frm_VYROBEK_subform_VD
frm_Dochazka_hl_N	frm_NORMY_statD	frm_VYROBA_pl	frm_vyrobek_vd
frm_Dochazka_hl_N_subform	frm_NORMY_statD_subform	frm_VYROBA_pl_subform	frm_VYROBEK_vy
frm_Dochazka_hl_O	frm_NORMY_vyk	frm_VYROBA_plo	frm_VYROBEK_zas
frm_Dochazka_hl_O_subform	frm_NORMY_zad	frm_VYROBA_plo_subform	frm_VYROBEK_zas_subform
frm_Dochazka_hl_P	frm_NORMY_zad_sel	frm_VYROBA_pro	frm_VYROBKY_CN_seznam
frm_Export_sel	frm_NORMY_zad_subform	frm_VYROBA_pro_null_subform	frm_VYROBKY_CN_seznamV
frm_FAKTURACE_hl	frm_Normy_zadCarKod	frm_VYROBA_pro_subform	frm_VYROBKY_CN_subformE
frm_FAKTURACE_hl_graf	frm_NORMY_zav_subform	frm_VYROBA_roz	frm_VYROBKY_CN_subformM
frm_FAKTURACE_hl_graf1	frm_OBJ_HL	frm_VYROBA_roz_subform	frm_VYROBKY_CN_subformD
frm_FAKTURACE_hl_subform	Frm_Objednavky	frm_VYROBA_rozN_subform	frm_VYROBKY_CN_subformS
frm_FAKTURACE_zak_subform	frm_OSTMAT_hl	frm_VYROBA_rozOp	frm_VYSEKDESKY_dat
frm_IDOB	frm_OSTMAT_subform	frm_VYROBA_rozOp_subform	frm_VYSEKDESKY_hl
frm_Idob_hl	frm_Porovnani_graf	frm_VYROBA_rozOPN	frm_VYSEKDESKY_seznam
frm_Idob_pr	frm_Porovnani_grafSum	frm_VYROBA_rozOpN_subform	frm_VYSEKDESKY_subformP
frm_Idob_pr_subform	frm_Porovnani_hl	frm_VYROBA_rozOpU	frm_Vystupy_sel
frm_IDOB_prid	frm_Porovnani_hlNorm	frm_VYROBA_rozOpU_subform	frm_VZ_HL
frm_IDOB_pridV	frm_Porovnani_norm	frm_VYROBA_Rozpac	frm_VZ_roz
frm_IDOB_prV	frm_Porovnani_normGraf	frm_VYROBA_RozpracVyr	frm_VZ_roz_subform
frm_IDOB_prV_subform	frm_Porovnani_normSum	frm_VYROBA_rozV_subform	frm_VZ_seznam
frm_IDOB_r	frm_Porovnani_plan	frm_VYROBA_rozVZ	frm_VZ_Subform
frm_IDOB_r_subform	frm_Poznamka_subform	frm_VYROBA_rozVZ_subform	frm_ZAK_hl
frm_IDOB_skl	frm_PracPOSTUP_hl	frm_VYROBA_sel	frm_ZAK_SEL_hl
frm_IDOB_skl_subform	frm_PracPOSTUP_hl2v	frm_VYROBA_set_subform	frm_ZAK_SEL_Subform
frm_IDOB_sl	frm_PracPOSTUP_mat	frm_VYROBA_sk	frm_ZAKAZKA_hl
frm_IDOB_sl_subform	frm_Product_subform_det	Frm_VYROBA_sk_subform	frm_ZAKAZKA_VP_subform

frm_IDOB_subform	frm_REKL_akt	Frm_VYROBA_skl	frm_ZAKAZNIK_hl
frm_Kalkulace_hl	frm_REKL_kat	frm_VYROBA_skl_subform	frm_ZAKAZNIK_seznam
frm_KALKULACE_hlPo	frm_REKL_kategorie	frm_VYROBA_sklA	frm_ZAKAZNIK_subform
frm_Kalkulace_subform	frm_REKL_oddeleni	frm_VYROBA_SKLA_subform	frm_ZakaznikTel
frm_Kalkulace_subform_sel	frm_REKL_seznam	frm_VYROBA_sklI	frm_ZakaznikTel_subform
frm_Kalkulace_subformG	frm_REKL_seznamR	frm_VYROBA_sklN_subform	frm_ZASOBA_hl
frm_Kalkulace_subformGZ	frm_REKL_seznamZ	frm_VYROBA_sklr	frm_ZASOBA_JARA
frm_Kalkulace_subformN	frm_REKL_stroj	frm_VYROBA_sklr_subform	frm_ZASOBA_sazava
frm_Kalkulace_subformNZ	frm_REKL_uvod	frm_VYROBA_slkN	frm_ZASOBA_SAZAVA_subfm
frm_Kalkulace_subformZ	frm_REKL_uzivatel	frm_VYROBA_slkNa	frm_ZASOBA_sl
frm_LITOGRAFIE_dat	frm_REKL_zavod	frm_VYROBA_slkNa_subform	frm_ZASOBA_subform
frm_LITOGRAFIE_hl	frm_REKLAMACE_subform_ts	frm_VYROBA_vy	frm_ZASOBA_zas
frm_LITOGRAFIE_seznam	frm_ReklamaceNav_subform	frm_VYROBA_Zadano	frm_ZASOBA_zas_subform

Tabulka č. 12: Seznam tiskových sestav Projektu APL_SCA(ADP)

Název sestavy	Název sestavy	Název sestavy	Název sestavy
rep_BOZP_nebezpeci	rep_CN_subreport	rep_Reklamace_vReklamProt	rep_VYROBA_pr
rep_BOZP_seznam	rep_Inspection List	rep_tblLide_kod	rep_VYROBA_vp
rep_BOZPT_nebezpeci	rep_Order	rep_URAZ_hl	rep_VYROBA_vp_subreport
rep_BOZPT_seznam	rep_qryNormy_vyk	rep_VYROBA_inpa	rep_VYROBA_paB
rep_CN	rep_qrySklad_mat	rep_VYROBA_inpa_subreport	rep_VYROBA_vpA
rep_CN_M	rep_qrySklad_vd_lit	rep_VYROBA_op	rep_VYROBA_vy
rep_CN_M_subreport	rep_qrySklad_vyr	rep_VYROBA_opA	
rep_CN_M_UVN	rep_REKL_navratka	rep_VYROBA_opB	
rep_CN_M_UVN_subform	rep_REKL_protokol	rep_VYROBA_pa	