



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ  
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT  
INSTITUTE OF INFORMATICS

## VYUŽITÍ BUSINESS INTELLIGENCE PRO HLEDÁNÍ ZÁVISLOSTÍ V DOTAZNÍKOVÝCH ŠETŘENÍCH

SEARCHING FOR DEPENDENCIES IN THE QUESTIONNAIRE SURVEYS USING BUSINESS  
INTELLIGENCE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MICHAL JANATA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ KŘÍŽ, Ph.D.

BRNO 2012

# ZADÁNÍ DIPLOMOVÉ PRÁCE

**Janata Michal, Bc.**

---

Informační management (6209T015)

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

**Využití Business Intelligence pro hledání závislostí v dotazníkových šetřeních**

v anglickém jazyce:

**Searching for Dependencies in the Questionnaire Surveys Using Business Intelligence**

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení, přínos návrhů řešení

Závěr

Seznam použité literatury

Přílohy

Seznam odborné literatury:

BRUST, J. A., FORTE, S. Mistrovství v programování SQL Serveru 2005. Tvorba databázových aplikací a řešení pro BI. Brno: Computer Press, 2007. 833 s. ISBN 978-80-251-1607-4.

LACKO, L. Business Intelligence v SQL Serveru 2008: reportovací, analytické a další datové služby. Brno: Computer Press, 2009. 456 s. ISBN 978-80-251-2887-9.

NOVOTNÝ, O., POUR, J., SLÁNSKÝ, D. Business Intelligence: jak využít bohatství ve vašich datech. Praha: Grada, 2005. 256 s. ISBN 80-247-1094-3.

PRICE, J. C#: programování databází. Praha: Grada Publishing a.s., 2005. 623 s. ISBN 80-247-0982-1.

Vedoucí diplomové práce: Ing. Jiří Kříž, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2011/2012.

L.S.

---

Ing. Jiří Kříž, Ph.D.  
Ředitel ústavu

---

doc. RNDr. Anna Putnová, Ph.D., MBA  
Děkan fakulty

V Brně, dne 12.05.2012

## **Abstrakt**

Tato diplomová práce se zabývá vytvořením aplikace pro analýzu dotazníků. V první části jsou uvedeny teoretické základy z přímého marketingu a Business Intelligence. Další kapitola je zaměřena na dostupné nástroje pro analýzu dat a BI. V této části je také zahrnuta analýza a příprava vstupních dat. Na závěr je předložen vlastní návrh řešení.

## **Klíčová slova**

Business Intelligence, Data Mining, přímý marketing, integrační služby, analytické služby, SQL server.

## **Abstract**

This master's thesis is focused on creating an application for survey analysis. In the first section are theoretical foundations of a direct marketing and Business Intelligence. The next chapter focuses on available tools for data analysis and BI. Analysis and preparation of input data are also included in this part. Finally, the solution proposal is presented.

## **Key words**

Business Intelligence, Data Mining, Direct Marketing, Integration Services, Analysis Services, SQL server.

## **Bibliografická citace práce**

JANATA, M. *Využití Business Intelligence pro hledání závislostí v dotazníkových šetřeních*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2012. 111 s.  
Vedoucí diplomové práce Ing. Jiří Kříž, Ph.D.

### **Čestné prohlášení**

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně 17. května 2012

.....  
Podpis

## **Poděkování**

Chtěl bych poděkovat vedoucímu práce, panu Ing. Jiřímu Křížovi, Ph.D., za čas a cenné rady poskytnuté při odborném vedení práce. Velké poděkování také patří mé rodině za podporu při studiu.

# OBSAH

ÚVOD.....	11
1. VYMEZENÍ PROBLÉMU A CÍLE PRÁCE .....	12
1.1. Vymezení problému .....	12
1.2. Cíle práce .....	12
1.3. Metodika práce.....	12
2. TEORETICKÁ VÝCHODISKA .....	13
2.1. Marketing .....	13
2.1.1. Direct marketing (přímý marketing).....	13
2.1.2. Výzkum trhu .....	14
2.1.3. Marketingový výzkum.....	15
2.1.3.1. Význam marketingového výzkumu pro rozhodování manažerů.....	15
2.1.3.2. Proces marketingového výzkumu.....	15
2.1.4. Marketingový informační systém .....	20
2.2. Business Intelligence.....	22
2.2.1. Architektura Business Intelligence .....	23
2.2.1.1. Datová transformace.....	23
2.2.1.2. Databázové komponenty .....	24
2.2.1.3. Analytické komponenty.....	24
2.2.1.4. Koncové nástroje .....	25
2.2.2. Datový sklad .....	25
2.2.2.1. Metody budování datového skladu.....	26
2.2.3. Integrované služby.....	26
2.2.3.1. Extrakce.....	27
2.2.3.2. Transformace .....	27
2.2.3.3. Loading.....	28
2.3. Analytické služby – OLAP kostky.....	28
2.3.1. „Dvanáctero“ původních pravidel OLAP .....	28
2.3.2. Fakta a dimenze .....	29



2.4.	Analytické služby – Data mining.....	29
2.4.1.	Algoritmy pro data mining (9).....	30
2.5.	Reportovací služby.....	31
2.5.1.	Životní cyklus reportu.....	31
3.	ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE .....	33
3.1.	Analýza vstupních dat.....	33
3.2.	Příprava vstupních dat.....	36
3.3.	Software pro analýzu dat.....	39
3.3.1.	IBM SPSS Statistics.....	39
3.3.2.	StatSoft STATISTICA 10.....	41
3.4.	Software pro Business Intelligence řešení .....	43
3.4.1.	Pentaho Business Analytics .....	43
3.4.2.	Microsoft Business Intelligence Development Studio.....	45
4.	VLASTNÍ NÁVRH ŘEŠENÍ, PŘÍNOS NÁVRHŮ ŘEŠENÍ.....	50
4.1.	Integrační služby .....	50
4.1.1.	Vytvoření cílových tabulek na SQL serveru.....	50
4.1.2.	Načítání dat ze souborů Flat File .....	51
4.1.3.	Načítání dat ze souborů Excel .....	54
4.2.	Analytické služby – Data Mining .....	57
4.2.1.	Analytický projekt .....	57
4.2.2.	Data Source.....	58
4.2.3.	Data Source View .....	58
4.2.4.	Mining Structure .....	59
4.2.5.	Mining Model .....	59
4.2.6.	Výsledky Mining Modelu.....	60
4.2.6.1.	Sít' závislostí (Dependency Network) .....	60
4.2.6.2.	Lift Chart .....	61
4.2.6.3.	Rozhodovací stromy.....	62
4.2.7.	Predikce .....	63
4.3.	Vytvořená aplikace.....	64

4.3.1. Okno SQL_Server.....	64
4.3.2. Okno New_Table.....	65
4.3.3. Hlavní okno programu.....	65
4.4. Interpretace výsledků – přínosy .....	66
4.4.1. HR - Hledají oslovené firmy lidské zdroje u konkurence a na univerzitách? 66	
4.4.2. INOVACE - Mají dotazované firmy zájem při inovacích spolupracovat s inovačními a výzkumnými centry, univerzitami a Akademií věd?.....	67
ZÁVĚR .....	69
SEZNAM POUŽITÉ LITERATURY .....	70
SEZNAM ZKRATEK .....	72
SEZNAM OBRÁZKŮ.....	74
SEZNAM TABULEK .....	75
SEZNAM GRAFŮ .....	75
SEZNAM PŘÍLOH.....	76

# ÚVOD

V současné době je pro podnikatelské subjekty největší výzvou a také rozhodujícím faktorem pro udržení konkurenceschopnosti umět správně využít získaná nebo dostupná data. Nestačí tedy pouze data vlastnit nebo je umět získat, ale je důležité, jaké informace a znalosti z dat získáme. Manažeři se denně rozhodují o tom, jakým směrem povedou svou firmu, a často se spoléhají na získané informace a s nimi spojené závislosti. Využívané informace však nejsou vždy relevantní a ve správném formátu. Takto nekvalitní informace mohou vést k nesprávným rozhodnutím a ke zvyšování nákladů za jejich zpracování, ať už za pomoci interních či externích zdrojů. Díky lidské povaze, která má potřebu si vše zjednodušovat a zpříjemňovat, se pro tyto účely v oblasti informačních technologií dynamicky rozvíjí využívání nástroje Business Intelligence.

Business Intelligence je složena z integračních, analytických a reportovacích služeb. Integrační služby slouží pro přípravu dat z nejrůznějších zdrojů. Pro analýzu dat je nejčastěji využívána OLAP kostka a data mining, který je v současné době nejvíce se rozvíjející metodou. Reportovací služby mají za úkol poskytovat získané informace ve správnou dobu a v přijatelné formě tak, aby manažerům napomohly ke správnému rozhodování.

Business Intelligence je využitelná v mnoha oblastech, jako jsou například informační systémy, výzkum a vývoj či prodej a marketing. V mé diplomové práci se soustředím na využití BI v oblasti marketingových procesů, kde je často využívanou metodou právě data mining.

Pomocí data miningu společnosti identifikují potenciální zákazníky a také drží své profitabilní zákazníky. Využíváním této metody vedoucí manažeři získají znalosti o svých zákaznících, ale i informace k efektivnějšímu řízení a minimalizování nákladů.

# **1. VYMEZENÍ PROBLÉMU A CÍLE PRÁCE**

## **1.1. Vymezení problému**

V současné době je prováděno mnoho marketingových průzkumů. I přes velké množství důmyslných analytických nástrojů, které trh v současnosti nabízí, je jen málo těch, které podrobují data zkoumání pomocí nějakých pokročilých statistických metod. Pokud si firmy na analýzu dat najmou odborníky, je to velice finančně náročné, a tak se manažeři často spokojí se základními agregačními funkcemi.

## **1.2. Cíle práce**

Hlavním cílem této diplomové práce je vytvoření aplikace pro analýzu dotazníků zaměřených na stakeholdery.

Pomocí aplikace bude možné nahrát data z různých vstupních souborů na SQL server a následně identifikovat závislosti a predikovat výstupní veličiny. K tomuto účelu bude využito integračních a analytických služeb architektury Business Intelligence.

V diplomové práci budou popsány základní teoretická východiska z oblasti přímého marketingu a Business Intelligence.

## **1.3. Metodika práce**

K vypracování diplomové práce a naplnění hlavního a dílčích cílů jsou využity pokročilé metody v rozhodování – konkrétně metoda využívající algoritmus rozhodovacích stromů.

Pro sběr dat je v této práci použita technika analýzy dokumentů. Jako vstupní data byly použity výsledky dotazníků, které nebyly sestaveny za účelem této práce.

Diplomová práce se zaměřuje na vytvoření .NET aplikace, která identifikuje ve vstupních datech závislosti, na základě kterých bude predikovat výstupní veličiny, jejichž uplatnění může pomoci společnosti v získávání nových zákazníků.

Charakteristika metod je popsána v kapitole zabývající se teorií.

## 2. TEORETICKÁ VÝCHODISKA

V této kapitole budou popsány základní teoretické poznatky, na kterých se bude dále v diplomové práci stavět. Teoretická východiska jsou rozdělena do dvou základních kapitol – Marketing a Business Intelligence.

### 2.1. Marketing

*„Marketing definujeme jako společenský a manažerský proces, jehož prostřednictvím uspokojují jednotlivci a skupiny své potřeby a přání v procesu výroby a směny produktů a hodnot.“ (6, s. 39)*

*„Cílem marketingu je poznat a pochopit zákazníka natolik dobře, aby mu výrobek nebo služba padla jako ušitá na míru a prodávala se sama.“ (5, s. 44)*

#### 2.1.1. Direct marketing (přímý marketing)

*„Přímý marketing (direct marketing) představuje přímou komunikaci s pečlivě vybranými individuálními zákazníky s cílem získat okamžitou odezvu a vybudovat dlouhodobé vztahy se zákazníky.“ (6, s. 928)*

Současným trendem v tržním prostředí je odpoutání se od masové komunikace. Společnosti proto budují propracovanější metody a cílenou komunikaci. K direct marketingu je využívána podrobná databáze, která obsahuje přizpůsobené nabídky dle jednotlivých segmentů či zákazníků. Hlavní výhodou direct marketingu je okamžitá měřitelná zpětná vazba zákazníků. Jako komunikační kanály se využívají direct mail, telemarketing, katalogový marketing, televize a jiná media, sociální sítě, e-marketing aj. (5; 6; 15)

Direct marketing patří mezi nejrychleji se vyvíjející metodu obsluhy zákazníka. Je to zapříčiněno tím, že snahou firem je minimalizování nákladů na oslovení cílových trhů prostřednictvím prodejních sil.

### **Charakteristika direct marketingu:**

- Neveřejný – sdělení je obvykle adresováno konkrétní osobě,
- okamžitý – sdělení lze připravit velmi rychle,
- přizpůsobivý – sdělení je možné připravit tak, aby bylo pro každého přitažlivé,
- interaktivní – umožňuje během dialogu zprávy upravovat dle reakcí spotřebitele. (5)

Přímý marketing je vhodný pro budování individuálních vztahů se zákazníky.

#### **2.1.2. Výzkum trhu**

Pojem výzkum trhu je definován v mnoha literaturách. Jednotliví autoři tento pojem chápou různě, a tak je možné najít nespočet definic.

Výzkum trhu je disciplína, která využívá poznatky z řady mnoha vědních oborů, jako jsou například psychologie, sociologie, kulturní antropologie, ekonomie a statistika. Výzkum trhu také silně ovlivnil vývoj v IT.

Výzkum trhu může být součástí informačního systému a pro účelovou analýzu získaných dat z informačního systému využívá specifickou metodologii. Dále rozšiřuje zdroje informačního systému i mimo podnik, a tím získává podstatné informace od cílového subjektu, které jsou potřebné pro adresnost marketingové aktivity. (20)

Výzkum trhu je možné vymežit dle jeho činností a cílů:

- Sběr informací,
- analýza informací – zjištění aktuálního stavu a určení vlivů, které na trh působí,
- sestavení obecných informací o trhu produktů – slouží k marketingovému rozhodování. (20)

Mezi klíčové prvky výzkumu trhu patří:

- Zákazník, spotřebitel,
- produkt (služba, výrobek a idea),
- prostředí (dodavatelé, konkurence, ekonomika, sociální situace aj.). (20)

### 2.1.3. Marketingový výzkum

V dnešní době by se málokterá firma obešla bez marketingového výzkumu, protože výstupy z něj jsou nezbytnou součástí celkové strategie společnosti. Marketingový výzkum se netýká pouze velkých společností, které budují marketingová oddělení a vynakládají na jejich chod 1 – 2 % ze zisku, ale také malých firem, které mohou využívat finančně méně náročné alternativy. Marketingový výzkum pomáhá zodpovědět otázky, které se týkají například nákupního chování zákazníků, účinnosti reklamy, cen a kvality distribuce výrobků. (5; 7)

*„Definujeme marketingový výzkum jako systematické plánování, shromažďování, analyzování a hlášení údajů a zjištění jejich důležitosti pro specifickou marketingovou situaci, před níž se ocitla firma.“ (5, s. 140)*

#### 2.1.3.1. Význam marketingového výzkumu pro rozhodování manažerů

V posledních letech marketingové prostředí v důsledku technologického pokroku, deregulace a globalizace zaznamenává velké změny. Tyto změny ovlivňují tržní chování a vytvářejí pro vedení společností nové výzvy. Proto musí manažeři pro vytváření stále efektivnějších strategií hledat a zpracovávat mnohem více adekvátních informací. (7)

#### 2.1.3.2. Proces marketingového výzkumu

Stejně jako existuje mnoho definic marketingového výzkumu, tak existuje i mnoho popisů procesu marketingového výzkumu.

Obrázek 1 - Proces marketingového výzkumu



*Zdroj: (5, s. 229), vlastní zpracování*

## **Proces marketingu (5)**

### **1. Definování problému a cíle výzkumu**

Problém by měl být definován tak, aby marketingový výzkumník věděl, na co se při sestavování dotazníku zaměřit. Na správně definovaném problému je pak postaven celý výzkum a výsledky z něj.

### **2. Vytvoření plánu výzkumu**

Vytvořený plán musí být co nejefektivnější na získávání dat. Je důležité rozhodnout se, jaká zdrojová data, výzkumné nástroje a přístupy a jaký soubor respondentů bude použit. Důležitá je také finanční stránka – pokud se během vytváření zjistí, že náklady na realizování výzkumu převyšují užitek, z realizace výzkumu sejde.

Při vytváření plánu výzkumu jsou využívány dva typy dat – sekundární a primární.

#### **▪ Sekundární data**

*„Sekundární data jsou data shromážděná k jinému účelu a již někde existují.“ (5, s. 143)*

Výhodou sekundárních dat je nízká pořizovací cena, a proto nejprve marketingoví výzkumníci zkontrolují, zda by tato data zcela nebo alespoň částečně nevyřešila daný problém.

Sekundární data jsou dále členěna na interní (obchodních, technické a finanční povahy) a externí (statistické přehledy, agenturní výzkumy, finanční zprávy aj.).

Jestliže jsou sekundární data například zastaralá, neúplná či vůbec neexistují, přechází marketingový výzkumník na finančně náročná primární data. (5; 18)

#### **▪ Primární data**

*„Primární data jsou nově shromážděná data ke specifickému účelu nebo pro specifický výzkumný projekt.“ (5, s. 143)*



Před získáváním primárních dat je standardní svolat skupinu respondentů a dotázat se, co si o daném projektu myslí. Na základě toho se použije výzkumná metoda, pomocí které se získají potřebná primární data.

Nevýhodou primárních dat je ovšem časová náročnost, náklady na pořízení a nutnost dobré přípravy. (5; 18)

Správně naplánovaný marketingový výzkum pomáhá zamezit dopouštění se chyb, které jsou mnohdy velmi nákladné.

### **3. Shromáždění informací**

Mezi základní metody shromažďování informací primárních dat patří pozorování, experiment a dotazování.

- **Pozorování**

Metoda pozorování se využívá při získávání informací pomocí smyslového vnímání, aniž by o tom sledovaný subjekt věděl. Pomocí této metody získáváme nezkreslené informace a také možnost zkontrolovat pravdivost vyhodnocených dotazníků. (7)

- **Výzkum prostřednictvím focus groups**

Tato metoda je založena na skupinové diskuzi, která je obsazena 6 – 10 účastníky. Respondenti jsou pečlivě vybráni dle předem daných kritérií, jako jsou například věk, pohlaví, osobnost či bydliště. Následně profesionální moderátor pokládá otázky a snaží se vysledovat postřehy a motivace respondentů. Za sledovaný skupinový rozhovor získávají respondenti zpravidla finanční odměnu. (5)

- **Dotazování**

Dotazování je možné provádět mnoha technikami, které se následně různým způsobem analyzují a vyhodnocují. Mezi základní techniky dotazování patří dotazování písemné, ústní, telefonické, on-line a kombinované. Každá z těchto technik má svá pozitiva, ale i negativa. (20)

## **Zásady tvorby dotazníků**

Při sestavování dotazníků je důležité splnit určitá pravidla. Hlavní, a také často opomíjenou, zásadou je formulovat srozumitelné otázky. Dalšími zásadami, neméně důležitými, jsou formulování jasných a jednoznačných otázek a zaznamenávání odpovědí jedním způsobem. (19)

### **Další zásady respektované při tvorbě dotazníku:**

*„1. Otázky se systematicky rozloží jako kontaktní, k věci, kontrolní a k dotazované osobě.*

*2. Snadné otázky předcházejí složitějším.*

*3. Obecné předcházejí konkrétnějším.*

*4. Využívá se pokynů pro přechod na další otázky v případě, že se předchozí respondentovi netýká.*

*5. Dostatečný prostor pro vyplňování odpovědí u volných otázek.“ (19, s. 81)*

#### **▪ Typy otázek**

Chceme-li získat použitelné informace, musíme se také na ně umět správně zeptat. K získávání potřebných informací slouží níže uvedené typy otázek.

#### **Uzavřené otázky**

*„Uzavřené otázky specifikují veškeré možné odpovědi a odpovědi na ně je snadnější interpretovat a zařazovat do tabulky.“ (5, s. 145)*

**Tabulka 1 - Uzavřené otázky**

Dichotomické	Otázky se dvěma možnými odpověďmi.
Multiple choice	Otázky se třemi a více odpověďmi.
Likertova škála	Výrok, v němž respondent projevuje míru souhlasu / nesouhlasu.
Sémantický diferenciál	Stupnice spojující dva bipolární světy. Respondent vybírá výrok, který reprezentuje jeho názor.
Stupnice důležitosti	Stupnice, pomocí níž se známkuje důležitost atributu.
Stupnice známek	Stupnice, pomocí níž se známkuje atribut od špatného k vynikajícímu.
Stupnice úmyslu něco koupit	Stupnice, pomocí níž se známkuje respondentův úmysl něco koupit.

*Zdroj: (5), vlastní zpracování*

## Otevřené

*„Otevřené otázky umožňují respondentům odpovídat vlastními slovy a často odhalí víc o tom, co si lidé myslí.“ (5, s. 145)*

**Tabulka 2 - Otevřené otázky**

Zcela nestrukturované	Otázky, na něž mohou respondenti odpovědět mnoha různými způsoby.
Slovní asociace	Jsou představována slova a respondenti reagují prvním výrazem, který jim přijde na mysl.
Dokončení věty	Je představena neúplná věta a respondenti ji dokončí.
Dokončení příběhu	Je představen neúplný příběh a respondenti jej dokončí.
Obrázek	Je představen obrázek se dvěma postavami, z nichž jedna něco říká, a respondenti se ztotožní s druhou postavou a vyplní prázdný rámeček.
Test matematické apercepce	Je představen obrázek a respondenti vypráví příběh o tom, co si myslí, že se děje, nebo může na obrázku být.

*Zdroj: (5), vlastní zpracování*

#### **4. Analýza informací**

V této fázi probíhá třídění a zjištění četnosti dat. Pro důležité proměnné se následně spočítají průměry a hodnoty rozptylu. Dále je také možné použít modely rozhodování, pokročilé statistické techniky nebo nástroje BI.

#### **5. Prezentace závěrů**

Marketingový výzkumník předloží manažerům relevantní závěry.

#### **6. Rozhodnutí**

Marketingoví manažeři musí na základě předložených závěrů vyhodnotit, zda marketingová strategie bude postavena na základě výsledků výzkumu či nikoliv. Další možností je zadat další marketingový výzkum.

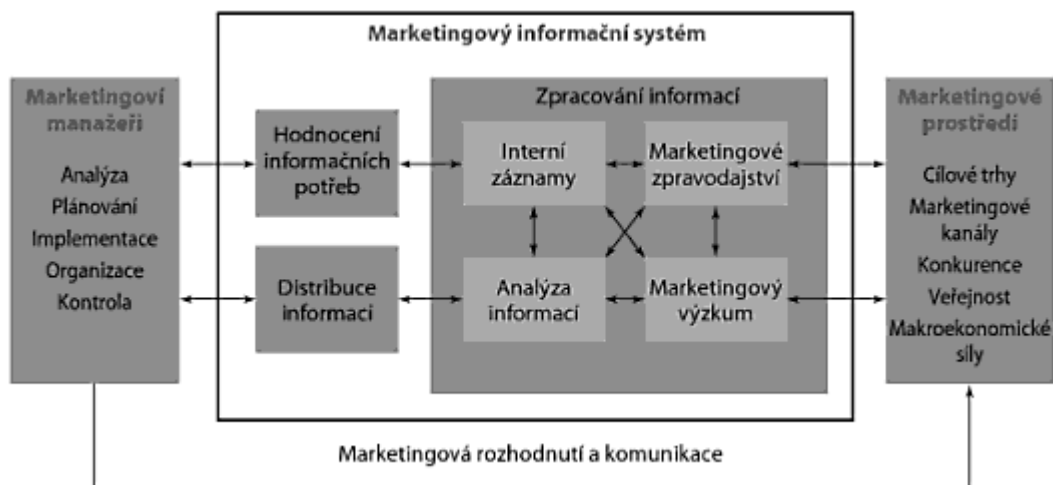
##### **2.1.4. Marketingový informační systém**

Přes snahu získávat mnoho informací si firmy stěžují, že nemají dostatek správných informací. Dle provedených výzkumů marketingoví manažeři nepotřebují pro rozhodování velké množství informací, ale informace správné. Proto společnosti začínají sledovat informační potřeby manažerů a na základě získaných poznatků vytvořit informační systémy, které splňují dané potřeby.

*„Marketingový informační systém (MIS) - to jsou lidé, vybavení a postupy pro sběr, třídění, analýzu, hodnocení a distribuci potřebných, včasných a přesných informací pro marketingové a ostatní manažery.“ (6, s. 399)*

Na níže uvedeném obrázku (viz Obrázek 2) je znázorněna koncepce marketingového informačního systému. Tento systém nejprve zpracuje informační potřeby manažerů, následně získá informace z interních záznamů, marketingového zpravodajství a výzkumů. Pomocí analytických nástrojů se získané informace vyhodnotí a zpracují do vhodné formy. V poslední fázi marketingový informační systém zpracované informace distribuuje zpět marketingovým manažerům. Zpracované informace slouží manažerům k plánování, implementaci či kontrole marketingových plánů.

Obrázek 2 - Marketingový informační systém



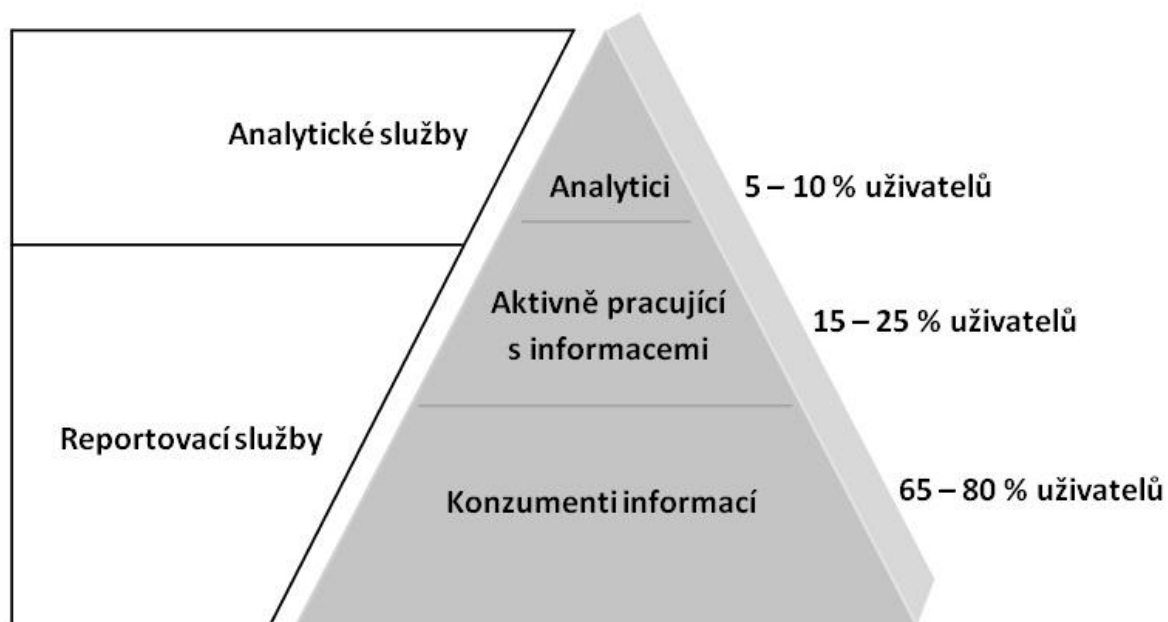
Zdroj: (6, s. 399)

## 2.2. Business Intelligence

*“Business Intelligence (BI) představuje komplex přístupů a aplikací IS/ICT, které téměř výlučně podporují analytické a plánovací činnosti podniků a organizací a jsou postaveny na principu multidimenzionality, kterým zde rozumíme možnost pohlížet na realitu z několika možných úhlů.” (13, s. 7)*

*“Business Intelligence je množina konceptů a metodik, které zlepšují rozhodovací proces za použití metrik, nebo systémů založených na metrikách. Účelem procesu je konvertovat velké objemy dat na poznatky, které jsou potřebné pro koncové uživatele. Tyto poznatky potom můžeme efektivně použít například v procesu rozhodování a mohou tvořit velmi významnou konkurenční výhodu.” (9, s. 14)*

**Obrázek 3 - Procentuální využití jednotlivých Business Intelligence technologií**

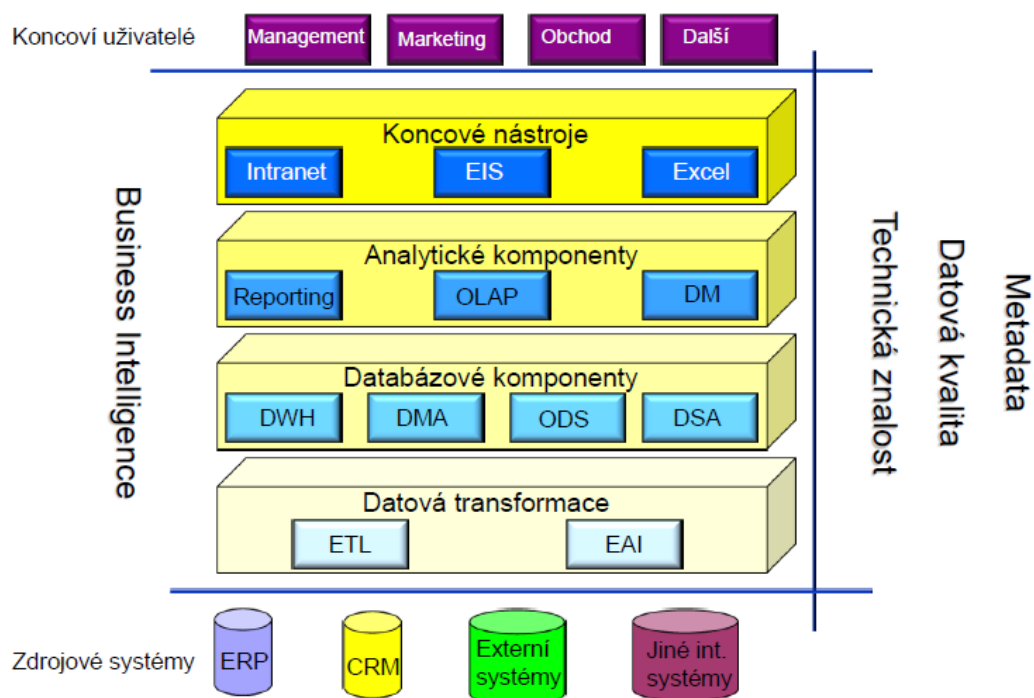


*Zdroj: (9, s. 15), vlastní zpracování*

## 2.2.1. Architektura Business Intelligence

Obecná koncepce architektury BI se skládá z několika vrstev a zahrnuje i zdrojové systémy a koncové uživatele.

Obrázek 4 - Architektura BI



Zdroj: (8, s. 13)

### 2.2.1.1. Datová transformace

- **ETL (Extrakce, Transformace, Loading)**

Cílem procesu ETL je výběr dat z různých zdrojů (**extrakce**), úprava, vyčištění a integrace dat (**transformace**) a jejich nahrání do cílové databáze nebo datového skladu (**loading**). Tato fáze je jednou z nejdůležitějších v zavádění BI řešení.

- **EAI (Enterprise Application Integration)**

Nástroje EAI zajišťují integraci mezi nesourodými podnikovými aplikacemi a systémy.

### **2.2.1.2. Databázové komponenty**

- **DWH (Data Warehouse)**

Datový sklad je základní prvek v řešení Business Intelligence, ve kterém jsou uloženy velké objemy dat podle určitých pravidel.

- **DMA (Data Mart)**

Datová tržiště se používají v případě, že firma nemá zaveden centralizovaný datový sklad nebo slouží pouze k potřebě určité skupiny lidí, a jsou zaměřeny na určitou problematiku ve firmě.

- **ODS (Operational Data Store)**

Do operativních datových úložišť se mohou ukládat agregovaná data z transakčních databází nebo různé číselníky. ODS také může sloužit pro podporu komunikace se zákazníkem.

- **DSA (Data Staging Areas)**

Dočasná úložiště slouží pro převod dat mezi datovým skladem a produkčními systémy nebo systémy, které pracují na jiném než databázovém formátu dat. (13)

### **2.2.1.3. Analytické komponenty**

- **Reporting**

Reporting slouží pro opakované nebo ad-hoc dotazování do databází nebo datových skladů.

- **OLAP (On-Line Analytical Processing)**

OLAP kostky obsahují agregovaná data uspořádaná do struktur podle předem definovaných dimenzí.



- **Data Mining**

Nástroje pro dolování dat slouží k hledání skrytých znalostí z dat uložených v datových skladech. (13)

- 2.2.1.4. **Koncové nástroje**

*„Prezentační vrstva (nástroje pro koncové uživatele), zajišťující komunikaci koncových uživatelů s ostatními komponentami řešení BI, tedy zejména sběr požadavků na analytické operace a následnou prezentaci výsledků.“ (13, s. 27)*

- 2.2.2. **Datový sklad**

*„Datový sklad je podnikově strukturovaný depozitář subjektivě orientovaných, integrovaných, časově proměnných, historických dat použitých pro získávání informací a podporu rozhodování. V datovém skladu jsou uložena atomická a sumární data.“ (9, s. 38)*

- **Orientace na předmět**

Do datových skladů by se data měla zapisovat podle předmětu zájmu (zákazník, zaměstnanec atd.), ne podle aplikace, v níž byla vytvořena.

- **Integrovanost**

Data v datovém skladu musí být konzistentní a integrovaná (jednotná terminologie, jednotky veličin atd.).

- **Neměnnost**

Do datových skladů se data pouze přidávají (nemění se, nemažou se) a následně se z nich načítají.

- **Časová variabilita**

V datových skladech musí klíčové atributy obsahovat čas. Data jsou z transakčních databází načítána jako série snímků za určitý časový úsek. (9)

### 2.2.2.1. Metody budování datového skladu

Pro budování datového skladu se v praxi používají dvě základní metody.

- **Metoda „velkého třesku“**

Při budování metodou „velkého třesku“ se celý datový sklad vytváří během jediného projektu. Výhodou je, že je možné vypracovat celý projekt ještě před jeho realizací, a tím minimalizovat náklady s případným přerušením projektu.

- **Přírůstková metoda**

Budování datového skladu přírůstkovou metodou spočívá v postupném přidávání jednotlivých řešení pro určitou předmětnou oblast. Tato řešení musí zapadat do celkového konceptu plánovaného datového skladu. V přírůstkové metodě směrem „shora dolů“ je podle požadavků uživatelů nejprve sestaven konceptuální model datového skladu a hierarchie předmětných oblastí a až následně se sestavují konceptuální modely jednotlivých předmětných oblastí. U přírůstkové metody směrem „zdola nahoru“ se konceptuální model odvíjí od zdrojových systémů (nejprve předmětné oblasti) a je zde kladen důraz spíše na data než na obchod. (9)

### 2.2.3. Integroční služby

*„Procesy integračních služeb mohou být navrženy pro jednorázovou akci, nebo pro akce periodicky se opakující. Jednorázovou akcí může být například migrace dat z jedné databázové platformy do jiné, nebo přenos dat ze souborů dokumentů kancelářských balíků do cílových databází. U periodicky se opakujících úloh, například při každodenním zavádění dat z produkčních databází do datového skladu, je důležité, aby tyto operace proběhly v požadovaném čase.“ (9, s. 75)*

Hlavní součástí integračních služeb je již výše zmíněný proces ETL.

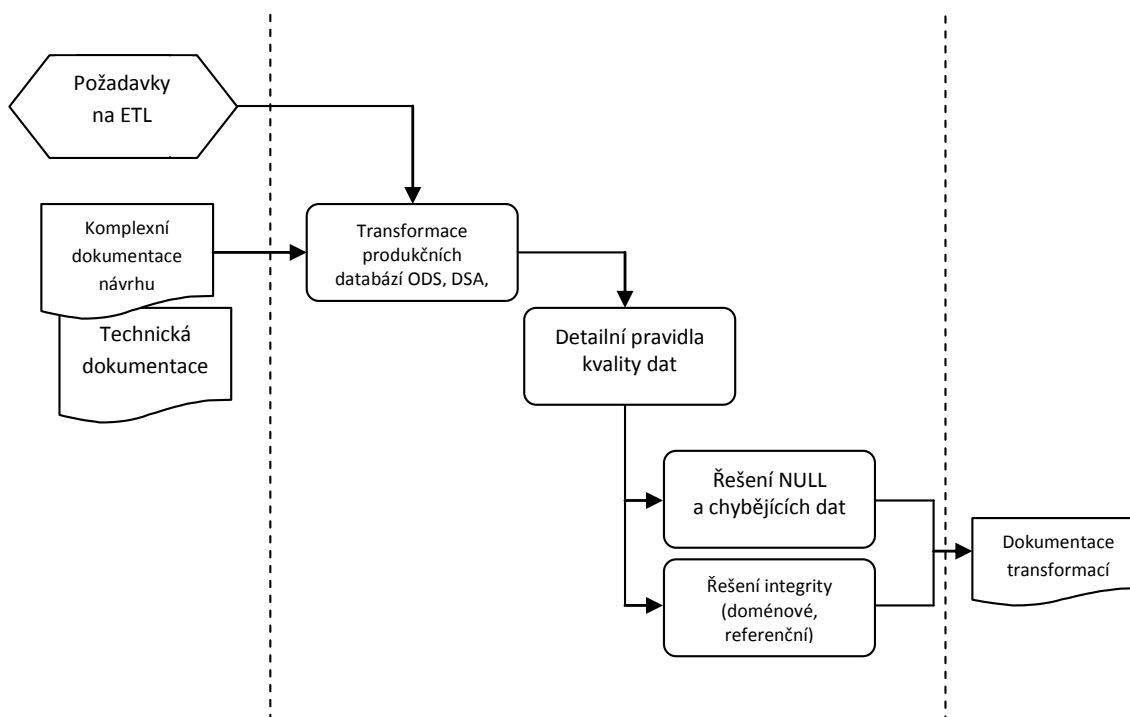
### 2.2.3.1. Extrakce

Cílem této etapy procesu ETL je sběr dat z rozmanitých zdrojů. Data se často získávají z nehomogenních operačních prostředí vzniklých prací v různých operačních nebo databázových systémech, kombinací dat z podnikových a archivních systémů nebo různých dokumentů. (9)

### 2.2.3.2. Transformace

Po výběru dat z různých zdrojů je nutná jejich transformace, to znamená, že se data upraví do stejné podoby, formátu a kvality.

Obrázek 5 - Návrh transformací, ETL



Zdroj: (13, s. 74), vlastní zpracování

„Při transformaci se nejčastěji vyskytují tyto okruhy problémů:

- *Nejednoznačnost dat,*
- *chybějící hodnoty,*

- *duplicitní záznamy,*
- *nejednotnost názvů pojmů a objektů,*
- *nejednotnost peněžních měn,*
- *nejednotnost formátů čísel a textových řetězců,*
- *problémy s referenční integritou,*
- *chybějící datum.*“ (9, s. 79)

### **2.2.3.3. Loading**

V poslední etapě procesu ETL se data zavedou v podobě databázových tabulek do datového skladu.

## **2.3. Analytické služby – OLAP kostky**

Pojem OLAP definoval Dr. E. F. Codd, britský informatik, takto:

*„OLAP je volně definovaná řada principů, které poskytují dimenzionální rámec pro podporu rozhodování.“* (9, s. 175)

### **2.3.1. „Dvanáctero“ původních pravidel OLAP**

- 1) Multidimenzionální konceptuální pohled,
- 2) transparentnost,
- 3) dostupnost,
- 4) konzistentní vykazování,
- 5) architektura klient-server,
- 6) generická dimenzionalita,
- 7) dynamické ošetření řídkých matic,
- 8) podpora pro více uživatelů,
- 9) neomezené křížové dimenzionální operace,
- 10) intuitivní manipulace s daty,
- 11) flexibilní vykazování,
- 12) neomezené dimenze a úrovně agregace. (9)

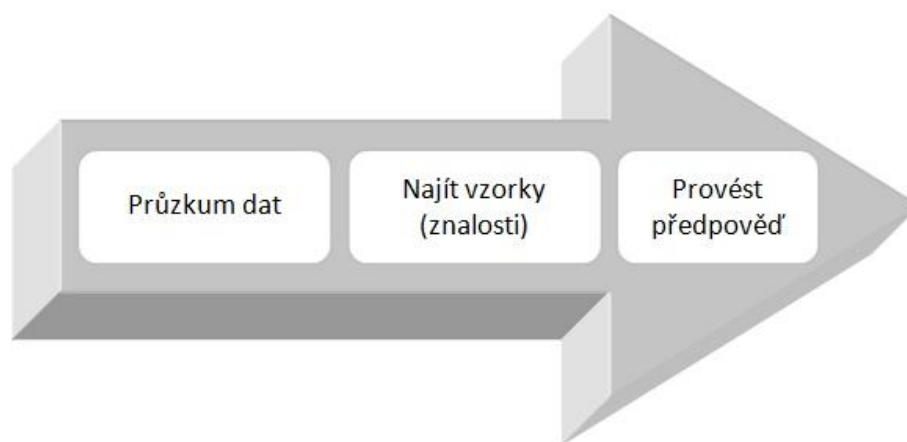
### 2.3.2. Fakta a dimenze

OLAP kostky se skládají z tabulek faktů a tabulek dimenzí. Tabulka faktů obsahuje velké množství reálných měřitelných údajů. Tabulky dimenzí tato data popisují a umožňují jejich hierarchické uspořádání. Tyto tabulky mohou být uspořádány do schématu hvězdy (jedna tabulka faktů, která je přímo spojena s tabulkami dimenzí, mezi nimiž není žádné spojení), do schématu sněhové vločky (jedna tabulka faktů a existuje spojení mezi dvěma nebo více tabulkami dimenzí) nebo do schématu souhvězdí (obsahuje více než jednu tabulku faktů). (9)

### 2.4. Analytické služby – Data mining

*„Data mining je proces analýzy dat z různých perspektiv a jejich přeměna na užitečné informace. Z matematického a statistického hlediska jde o hledání korelací, tedy vzájemných vztahů nebo vzorů v datech. Data mining je proces, jehož cílem je těžba informací v databázích. Využívá statistické metody a další metody hraničící s oblastí umělé inteligence.“ (9, s. 265)*

Obrázek 6 - Procesní schéma data miningu



*Zdroj: (9, s 265), vlastní zpracování*

### **2.4.1. Algoritmy pro data mining (9)**

Algoritmy pro data mining vycházejí z různých statistických metod, jako jsou korelace, lineární a logická regrese, předpovědi trendů, genetické algoritmy nebo například neuronové sítě a další.

- **Rozhodovací stromy**

Rozhodovací nebo rozpadové stromy odhalují závislosti, specifické vlastnosti a vzorce ve vstupních datech a hierarchicky je třídí na různé úrovně od kořenového atributu k nejnižším vrstvám. Tento algoritmus je velice rychlý, přehledný a snadno interpretovatelný.

- **Shlukování**

Tento algoritmus vyhledává klastry (shluky) dat na základě jejich určité podobnosti. Tyto shluky vytvářejí tzv. ostrovy podobnosti, které mohou identifikovat například cílovou skupinu zákazníků.

- **Asociační pravidla**

Asociační pravidla se nejčastěji používají pro analýzu nákupního košíku, sloužící k rozmístění zboží v supermarketech nebo pro vytvoření akčních nabídek. Tento algoritmus je založen na pozitivní a negativní korelaci.

- **Časové řady**

Časové řady zkoumají a predikují vývoj různých ukazatelů v čase. Nejčastěji se hledají trendy finančních, ekonomických nebo sociálních ukazatelů (zisk, objem prodeje, HDP, nezaměstnanost atd.).

- **Neuronové sítě**

Neuronové sítě pracují na principu lidského mozku, tedy na rozpoznávání vzorů a minimalizaci chyb. Neuronová síť se skládá z jedné nebo více vrstev a data se rozdělí

do dvou množin – trénovací a testovací. V každém kroku výpočtu se pak vypočítaná data porovnají se skutečnými, dokud není dosaženo minimální požadované chyby.

- **Naive Bayes**

Pomocí toho algoritmu se řeší složité analýzy, jako je například paralelní korelace apod. Je založen na Bayesově větě a postupně zpřesňuje odhad pravděpodobnosti jevu na základě kombinace domněnek s předchozími znalostmi.

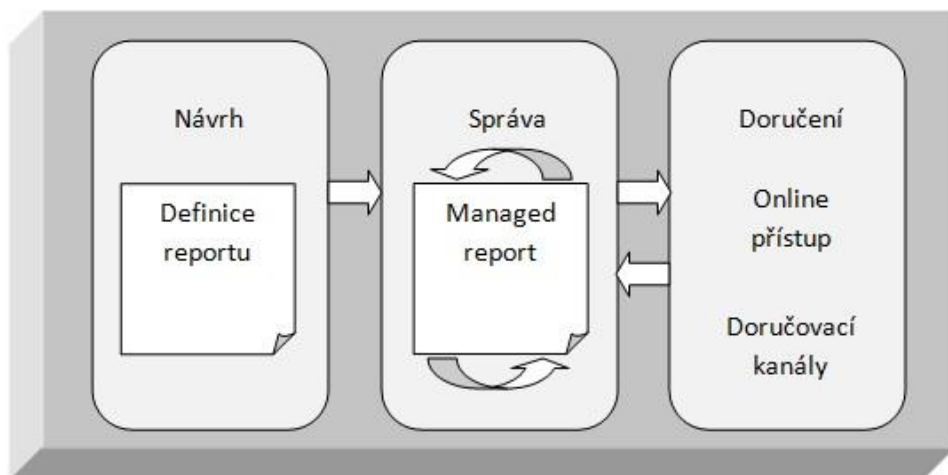
## 2.5. Reportovací služby

*„Úlohou reportovacích služeb je poskytovat ve vhodné formě a včas podklady pro podporu rozhodování na všech stupních organizační struktury. Hlavní důvod pro nasazování reportovacích služeb je generování výstupů v elektronické nebo papírové podobě, které umožní zaměstnancům na všech pozicích efektivní přístup k datům, a tím je podpoří v jejich činnosti, nebo v případě manažerů jim poskytnou důležité informace ve vhodné formě, které budou potřebovat v procesu rozhodování.“ (9, s. 324)*

### 2.5.1. Životní cyklus reportu

Životní cyklus reportu se skládá ze tří základních částí – návrh, správa a doručení reportu.

Obrázek 7 - Životní cyklus reportování



Zdroj: (9, s. 329), vlastní zpracování

- **Návrh reportu**

V této části se vybírá, jak budou data zobrazena, ale zejména jaká data bude report obsahovat. Výběr dat se provádí buď SQL příkazem select, nebo použitím již existujícího pohledu. Data je možné zobrazovat v podobě tabulek, matic či grafů a je u nich možné nastavit různé řazení, seskupování, filtrování apod.

- **Správa reportu**

Ve správě reportů se určuje, kdo a kdy bude reporty dostávat. Je možné odesílat reporty na vyžádání (On-Demand) nebo automaticky podle časového plánu.

- **Doručení reportu**

Zde se určuje forma a způsob doručení. Uživatelé mohou reporty obdržet emailem nebo v podobě sdílených souborů v různých formátech (webové, tisknutelné, datové).



### **3. ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE**

Hlavním cílem této části je postavit základ pro vlastní návrh řešení. Kapitola obsahuje analýzu včetně přípravy vstupních dat a analýzu vybraných aplikací pro analýzu dat a řešení Business Intelligence.

#### **3.1. Analýza vstupních dat**

Jak už bylo zmíněno v teorii, v marketingovém výzkumu existují dva druhy zdrojových dat – sekundární a primární. V případě, že marketingový výzkum sestavujeme pro konkrétní účel zkoumání, nazýváme vzniklé záznamy daty primárními. Pokud existují data již z nějakého předešlého marketingového výzkumu nebo například data relačních databází a chceme z těchto dat získat nové informace, které by nám mohly poskytnout konkurenční výhodu, nazýváme tato data daty sekundárními.

Zdrojovými daty v této práci jsou odpovědi z dotazníku, jenž je velice rozsáhlý a nebyl sestaven pro tento účel, proto je nejdříve nutné určit, která data budou pro nás užitečná a jaký výstup (informace, poznatky) z nich chceme získat.

Dotazník obsahoval tyto otázky:

- 1-7)** Obecné informace o firmě – sídlo firmy, rok založení, právní forma, počet zaměstnanců, obor podnikání atd.
- 8)** Ohodnoťte Váš vztah k jednotlivým skupinám stakeholderů dle jednotlivých indikátorů ve sloupcích (důležitost vztahu, důležitost budování a udržování vztahu pro budoucnost, důležitost intenzity tohoto vztahu, význam kvality tohoto vztahu).
- 9)** Ohodnoťte vztah jednotlivých skupin stakeholderů k Vašemu podniku dle jednotlivých indikátorů ve sloupcích (důležitost vztahu, důležitost budování a udržování vztahu pro budoucnost, důležitost vztahu z pohledu zvýšení hodnoty nabízených produktů a služeb, důležitost vztahu pro vyhledávání nových příležitostí na trzích).

- 10)** Ohodnoťte dle významu jednotlivé nabízené důvody, kvůli kterým má podnik zájem vztahy s jednotlivými stakeholdery navazovat, budovat a udržovat (informace, kontakty, finanční zdroje, flexibilita, inovace, loajalita, dobré jméno, výkonnost, kvalita, příležitosti, nutnost, přežití, kvalita služeb, kvalita výrobků, spokojenost, lidské zdroje, zakázky, dobrá atmosféra).
- 11)** Z pohledu jednotlivých fází klasického hodnototvorného řetězce ohodnoťte, jak vnímáte vztah s jednotlivými skupinami stakeholderů v souvislosti s tvorbou přidané hodnoty Vašich produktů? Zasahují do těchto procesů pozitivně, negativně nebo vůbec nemají vliv? (Vstupní logistika, výrobní operace, marketing, prodej, výstupní logistika, služby zákazníkům, řízení pracovních sil, technologický rozvoj, nákup).
- 12)** Jaké služby jsou poskytovány v souvislosti s fázemi hodnototvorného řetězce vašim podnikem jednotlivým stakeholderům nebo naopak jednotlivými stakeholdery Vám? U jednotlivých typů služeb označte kdo je poskytuje - podnik, stakeholder, ani jeden nebo oba dva? (Informační, finanční, poradenské, servisní, vzdělávací, obchodní, jiné).
- 13)** Vztahy k jednotlivým stakeholderům ovlivňují následující oblasti řízení? (Plánování, organizování, stimulace, kontrola, výrobní postupy, personální řízení, obchodní aktivity, finanční řízení, informační management).
- 14)** Jakým způsobem realizujete prodej vlastních výrobků a služeb? (Vlastní formy prodeje – přímo na vyžádané objednávky, přímý prodej, vlastní prodejna maloobchod, vlastní prodejna velkoobchod, vlastní distribuční síť, franšíza, elektronický obchod, zprostředkovaně – česká obchodní firma, zahraniční obchodní firma, maloobchod, velkoobchod, dealeři, cizí distribuční síť, jiná forma).

- 15)** Využíváte outsourcing v jednotlivých fázích hodnotového řetězce? (Vstupní logistika, výrobní operace, marketing, prodej, výstupní logistika, služby zákazníkům, řízení pracovních sil, technologický rozvoj, nákup).
- 16-18)** U následujících kategorií určete, zda jsou pro Vás předností, slabinou nebo ani jedno. (Například znalost zahraničního trhu, přístup k finančním zdrojům, vztahy s vlastníky, jazykové znalosti, změna v chování zákazníků, stanovování marketingových cílů atd.).
- 19-23)** Vybrané ukazatele podniku – celkový příjem, export, import, ROA, ROE, produktivita práce, výroba/služby.

Dotazník je zaměřen na tyto stakeholdery:

- Zákazníci,
- dodavatelé,
- zaměstnanci,
- vedení,
- vlastníci,
- finanční instituce,
- úřady místní správy,
- hospodářská nebo obchodní komora,
- inovační centrum,
- poradenská společnost,
- úřady státní správy,
- občané, místní komunita,
- vysoké školy, univerzity,
- akademie věd, výzkumná centra,
- konkurence,
- média,
- jiné.

Práce bude dále zaměřena na otázku č. 10 - Ohodnoťte dle významu jednotlivé nabízené důvody, kvůli kterým má podnik zájem vztahy s jednotlivými stakeholdery navazovat, budovat a udržovat.

### **Možné obchodní otázky**

- Navazují a udržují dotazované firmy vztahy se zákazníky a dodavateli kvůli kontaktům?
- Mají dotazované firmy zájem při inovacích spolupracovat s inovačními a výzkumnými centry, univerzitami a Akademií věd?
- Hledají oslovené firmy lidské zdroje u konkurence a na univerzitách?
- Využívají firmy pro získávání informací poradenskou činnost?
- Ovlivňují vztahy s vlastníky, finančními institucemi a úřady místní správy finanční zdroje podniku?

Jako vstupní data tedy budou použity odpovědi na otázku č. 10 a některé obecné informace o firmě – konkrétně právní forma, počet zaměstnanců a obor podnikání.

### **3.2. Příprava vstupních dat**

Aby bylo možné s daty opakovaně pracovat, je nutné, aby data měla předem určenou strukturu a splňovala určitá pravidla, která budou v této kapitole definována. Cílem je také identifikovat nutnost případných úprav vstupních dat při procesu ETL.

Všechny odpovědi na otázky jsou uloženy v souboru Excel (náhled viz Obr. 8). Jak je na obrázku vidět, první řádky obsahují znění otázky a následují sumační údaje všech typů odpovědí. Aby bylo možné data zpracovat, je nutné tyto řádky odstranit.

Dalšími problémy jsou sloučené buňky s názvy jednotlivých stakeholderů, které by SQL server nebyl schopen zpracovat, a opakující se stejné názvy jednotlivých sloupců u různých stakeholderů (10.1, 10.2, ...). Aby bylo možné využít funkce `ColumnNamesInFirstDataRow` (názvy sloupců v prvním řádku dat), je třeba, aby pouze první řádek obsahoval hlavičku (v SQL názvy atributů) a na druhém řádku už byla data,

podle kterých se bude určovat datový typ atributu. Nevhodné je také používání diakritiky pro názvy atributů a některých speciálních znaků. Řešením je tedy přejmenovat názvy sloupců například na Zakaznici\_1, Zakaznici\_2, ..., Dodavatele\_1 atd.

Tabulka by také měla obsahovat jeden klíčový atribut (obvykle první sloupec tabulky obsahující primární klíče záznamů). I zde je vhodná úprava, a to změna hodnoty na číslo (bez textu před číslem). Tuto změnu je možné provést v době přípravy dat v Excelu nebo během etap ETL.

Posledním zjištěním je neúplnost a nekonzistentnost dat, která je obvyklým problémem u většího množství dat. Z obr. 8 je patrné, že respondent měl na výběr z pěti možností (1 – velmi důležitý, 2 – důležitý, 3 – neutrální, 4 – nedůležitý, 5 – vůbec není důležitý). V datech jsou ale buňky, které obsahují hodnotu 0 nebo jsou prázdné (NULL), protože respondent otázku nevyplnil. Tyto buňky by mohly ovlivňovat dataminingový model. Z charakteru otázky vyplývá, že pokud respondent otázku nevyplnil, není pro něj pravděpodobně navazování a udržování vztahů s daným stakeholderem nebo z daného důvodu vůbec důležité, a proto je číslici 0 nebo hodnotu NULL potřeba nahradit číslem 5 (vůbec není důležitý).



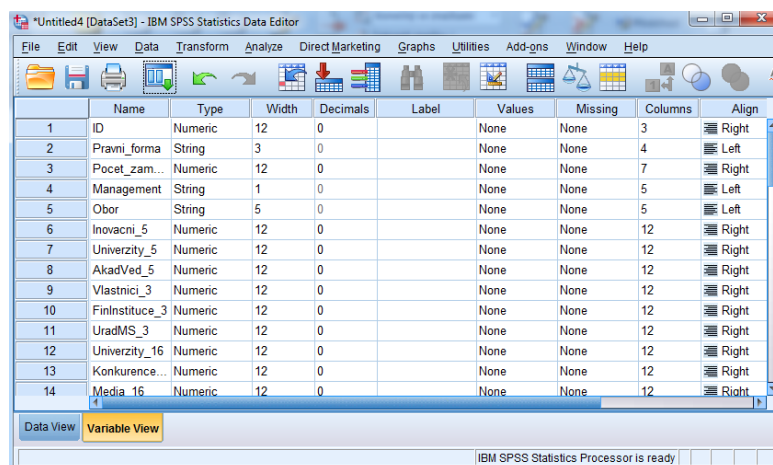
### 3.3. Software pro analýzu dat

Tyto programy umožňují analýzu dat pomocí různých statistických metod.

#### 3.3.1. IBM SPSS Statistics

„IBM SPSS Statistics je komplexní, uživatelsky příjemný modulární soubor nástrojů určený obchodním uživatelům, analytikům a statistickým programátorům. Primárním modulem systému je modul Base, k dispozici jsou i další moduly zaměřené na konkrétní oblasti statistiky.“ (1)

Obrázek 9 - Prostředí IBM SPSS Statistics



Zdroj: vlastní zpracování (screenshot)

Základní modul Base obsahuje nástroje pro přípravu dat, jejich následnou analýzu, prezentaci a sdílení.

Tento software umožňuje načtení dat ze souborů Excel, databázových souborů dBase, Syk, Lotus atd. Datové typy jsou automaticky získány z dostupných metadat, případně je možné si nadefinovat proměnné a jejich typy ručně. Po načtení je uživateli umožněno odstranění duplicitních dat použitím nástroje Identify Duplicate Cases.

Pro analýzu dat program nabízí využití jednoduchých statistických metod jako absolutní či relativní četnosti pro každou z diskretních hodnot, výpočet základních statistických

charakteristik jako je průměr, směrodatná odchylka a studentovo rozdělení nebo výpočty regrese, korelace apod. Další možností pro analýzu dat je využití některých pokročilých metod – shlukovací algoritmy, škálování, neuronové sítě i predikční modely.

Další zajímavou součástí jsou techniky Direct Marketing, které umožňují v několika krocích jednoduše identifikovat nejlepší zákazníky, segmentaci zákazníků a podporu a vyhodnocování marketingových kampaní.

**Obrázek 10 - Techniky Direct Marketing v SPSS Statistics**



*Zdroj: vlastní zpracování (screenshot)*

Nevýhodou těchto technik je požadavek na konkrétní data (např. datum nákupu, počet transakcí), bez kterých nelze získat žádné výsledky.



Na serveru IBM je možné získat 14ti denní Trial verzi tohoto programu. Studentská licence se všemi součástmi potom stojí od 199 USD<sup>1</sup>.

### 3.3.2. StatSoft STATISTICA 10

Aplikace STATISTICA 10 od společnosti StatSoft je další kvalitní nástroj pro statistickou analýzu dat, který je dostupný i v českém jazyce. Software se skládá z několika balíčků od obecných analytických nástrojů po pokročilé dataminingové nástroje.

Tabulka 3 - Balíčky STATISTICA 10

Speciální balíčky	Obecné analytické nástroje	Průmyslové nástroje	Neuronové sítě	Dataminingové nástroje	Podnikové systémy	Webové aplikace
<ul style="list-style-type: none"> <li>• Standard</li> <li>• Standard Plus</li> <li>• SPC Standard</li> <li>• SPC Standard Plus</li> </ul>	<ul style="list-style-type: none"> <li>• Base</li> <li>• Pokročilé lineární a nelineární modely</li> <li>• Vícerozměrné průzkumné techniky</li> <li>• Analýza síly testu</li> <li>• VEPAC</li> <li>• SVB Mapy</li> <li>• SVB Hypotézy</li> </ul>	<ul style="list-style-type: none"> <li>• Diagramy pro řízení kvality</li> <li>• Analýza procesů</li> <li>• Navrhování experimentů</li> <li>• Vícerozměrné statistické řízení procesů (MSPC)</li> </ul>	<ul style="list-style-type: none"> <li>• Automatizované neuronové sítě</li> <li>• Automatizované neuronové sítě - Generátor kódu</li> </ul>	<ul style="list-style-type: none"> <li>• Data Miner</li> <li>• QC Miner</li> <li>• Text Miner</li> </ul>	<ul style="list-style-type: none"> <li>• Enterprise</li> <li>• Enterprise/QC</li> <li>• Data Warehouse</li> <li>• Analýza sekvencí, asociací a propojení</li> <li>• Document Management System</li> <li>• Monitoring and alerting system</li> <li>• Six Sigma</li> <li>• ETL</li> </ul>	<ul style="list-style-type: none"> <li>• Web Server</li> <li>• Web Knowledge Portal</li> </ul>

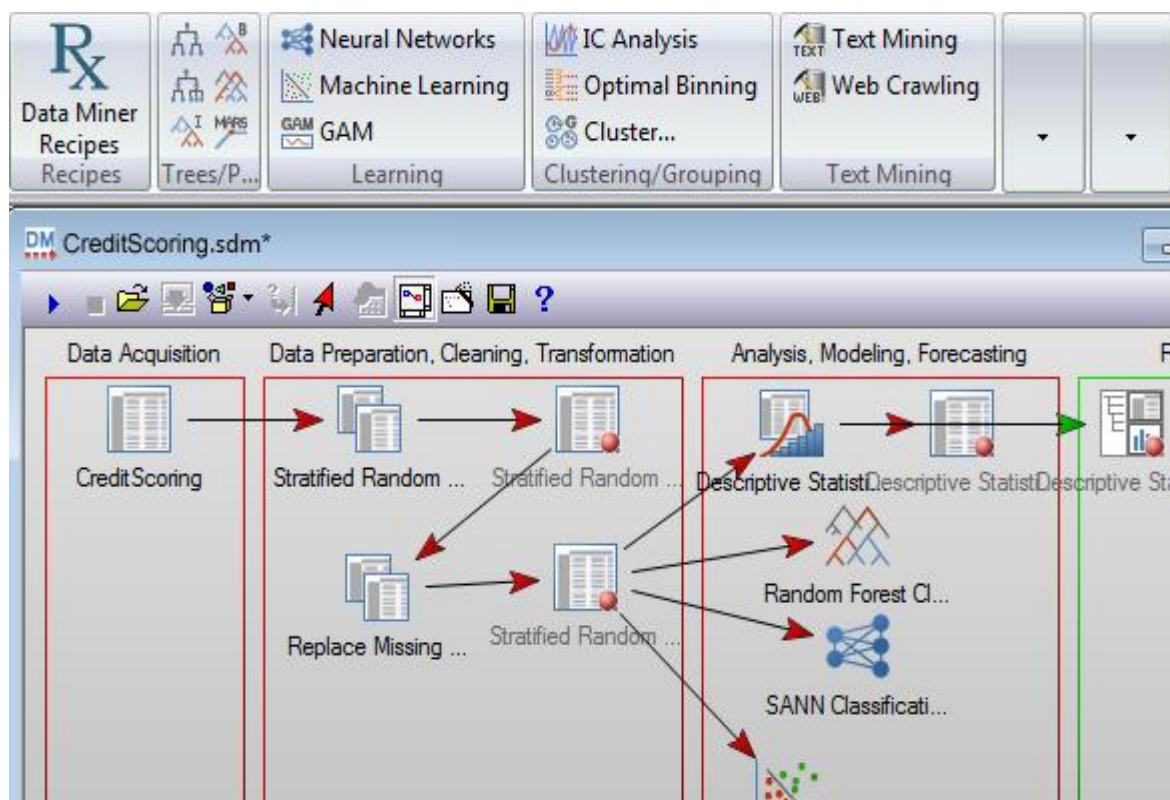
Zdroj: (17), vlastní zpracování

STATISTICA umožňuje pracovat s daty ze souborů Microsoft Office, databází OLE DB, datových kostek OLAP nebo platformy Microsoft SharePoint. Další výhodou je možnost přístupu ke všem funkcím pomocí vlastního integrovaného jazyka SVB (STATISTICA Visual Basic) umožňující například jednotlivé funkce programu pomocí maker automatizovat. (16)

<sup>1</sup>SOFTWARE TIME. *SPSS Statistics Premium GradPack 20*. [online]. 2012 [cit. 2012-05-01]. Dostupné z: [http://www.softwaretime.com.au/spss-statistics-premium-gradpack-20?utm\\_source=myshopping&utm\\_medium=cpc&utm\\_campaign=Software&utm\\_term=SPSS+Statistics+Premium+GradPack+20](http://www.softwaretime.com.au/spss-statistics-premium-gradpack-20?utm_source=myshopping&utm_medium=cpc&utm_campaign=Software&utm_term=SPSS+Statistics+Premium+GradPack+20).

Dataminingové nástroje obsahují interaktivní a klasifikační stromy, asociační pravidla, shlukovou analýzu, regresní modely, neuronové sítě a mnohé další pokročilé rozhodovací techniky. Další součástí je také v poslední době často vyhledávaný Text Miner, který značně usnadňuje dolování dat z různých textových souborů.

Obrázek 11 - Prostředí STATISTICA Data Miner



Zdroj: STATSOFT. Data Mining. [online]. 2012 [cit. 2012-05-01]. Dostupné z:  
<http://www.statsoft.com/Portals/0/Products/Data-Mining/CreditScoring1.png>.

Program je velice přehledný, intuitivní a uživatelsky příjemný. Na internetových stránkách je k dispozici po vyplnění krátkého formuláře 30ti denní Trial verze obsahující ovšem pouze základní funkce. Cena akademické licence základního produktu STATISTICA Standard, který se skládá z modulů STATISTICA Base, Pokročilé lineární a nelineární

modely a Vícerozměrné průzkumné techniky, je 2707 Kč bez DPH.<sup>2</sup> Stejný balíček pro profesionální využití (nesíťová verze) stojí 41640 Kč bez DPH.<sup>3</sup>

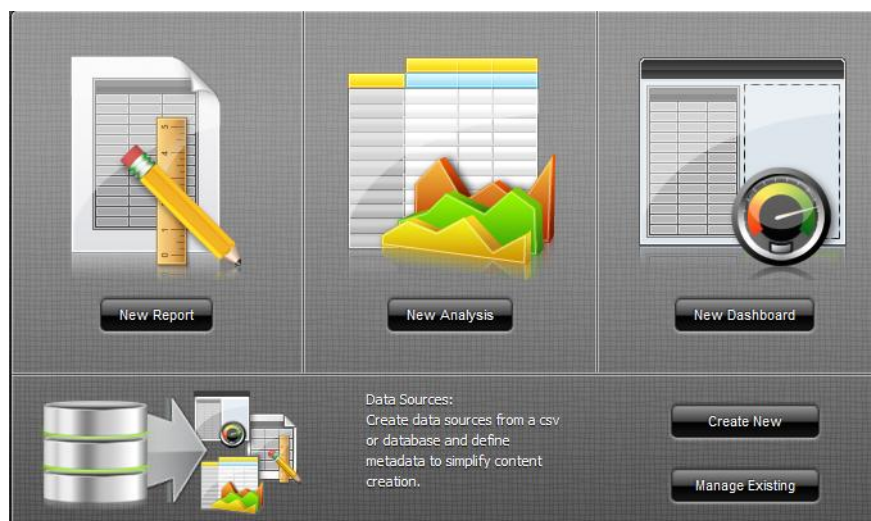
### 3.4. Software pro Business Intelligence řešení

Software pro BI řešení je obvykle nástavba různých služeb k databázovému serveru. Existuje řada komerčních i Open-source řešení.

#### 3.4.1. Pentaho Business Analytics

*„Pentaho je na špici žebříčku business intelligence nástrojů pro vytváření manažerských informačních systémů založených na Open Source. Dlouholetým vývojem došlo k integraci více nástrojů, řešících jednotlivé části problematiky MIS, do jednotné a konzistentní formy. Sada nástrojů Pentaho pomůže při řešení prakticky všech úkolů na poli MIS:*

Obrázek 12 - Pentaho Business Analytics



Zdroj: vlastní zpracování (screenshot)

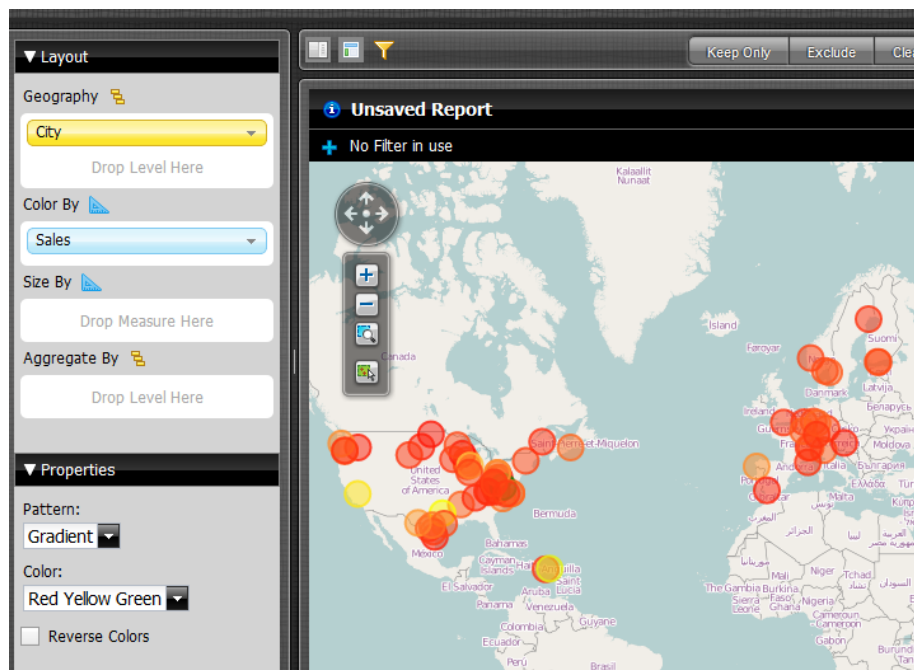
<sup>2</sup>STATSOFT. Ceník Statistica 10. Akademické ceny. [online]. 2011 [cit. 2012-05-01]. Dostupné z: [http://www.interway.sk/files/statistica/08-2011/cennik\\_statistica10\\_jednouzivatelske\\_sk\\_29-04-2011\\_akademicke.pdf](http://www.interway.sk/files/statistica/08-2011/cennik_statistica10_jednouzivatelske_sk_29-04-2011_akademicke.pdf).

<sup>3</sup>STATSOFT. Ceník Statistica 10. Jednouzivatelská instalace. [online]. 2011 [cit. 2012-05-01]. Dostupné z: [http://www.interway.sk/files/statistica/08-2011/cenik\\_statistica\\_10\\_jednouzivatelske\\_14-04-2011.pdf](http://www.interway.sk/files/statistica/08-2011/cenik_statistica_10_jednouzivatelske_14-04-2011.pdf).

- *Datová integrace (projekt Kettle jakožto ETL/ETTL nástroj řeší problémy přípravy dat pro MIS)*
- *Analýzy (projekt Mondrina umožňuje snadné vytváření analýz na bázi OLAP)*
- *Reporting (realizován z původního projektu JFreeReport)*
- *Publikování dat*
- *Dolování dat*
- *... a řadu dalších“ (2)*

Pro využití integračních služeb společnost nabízí ještě druhý produkt - Pentaho Data Integration, který umožňuje získat data z komerčních i Open-source souborů a databází. Kromě klasických analytických a reportovacích služeb software nabízí zajímavé zobrazení analyzovaných dat v neobvyklých grafech – například zobrazení výše objemů v jednotlivých městech s přímým napojením na Google Maps (viz obr. 13).

**Obrázek 13 - Zobrazení prodejů v Pentaho na mapě Google**



*Zdroj: vlastní zpracování (screenshot)*

### **3.4.2. Microsoft Business Intelligence Development Studio**

Základem Business Intelligence Development Studia je Microsoft Visual Studio 2008, do něhož jsou přidány nové typy projektů, které jsou specifické pro SQL Server Business Intelligence. Je to základní prostředí, v němž je možné vypracovat různá obchodní řešení obsahující projekty integračních, analytických a reportovacích služeb. Každý typ projektu poskytuje šablony pro vytváření objektů potřebných pro řešení Business Intelligence a nabízí velké množství nástrojů a wizardů pro návrh a práci s těmito objekty.

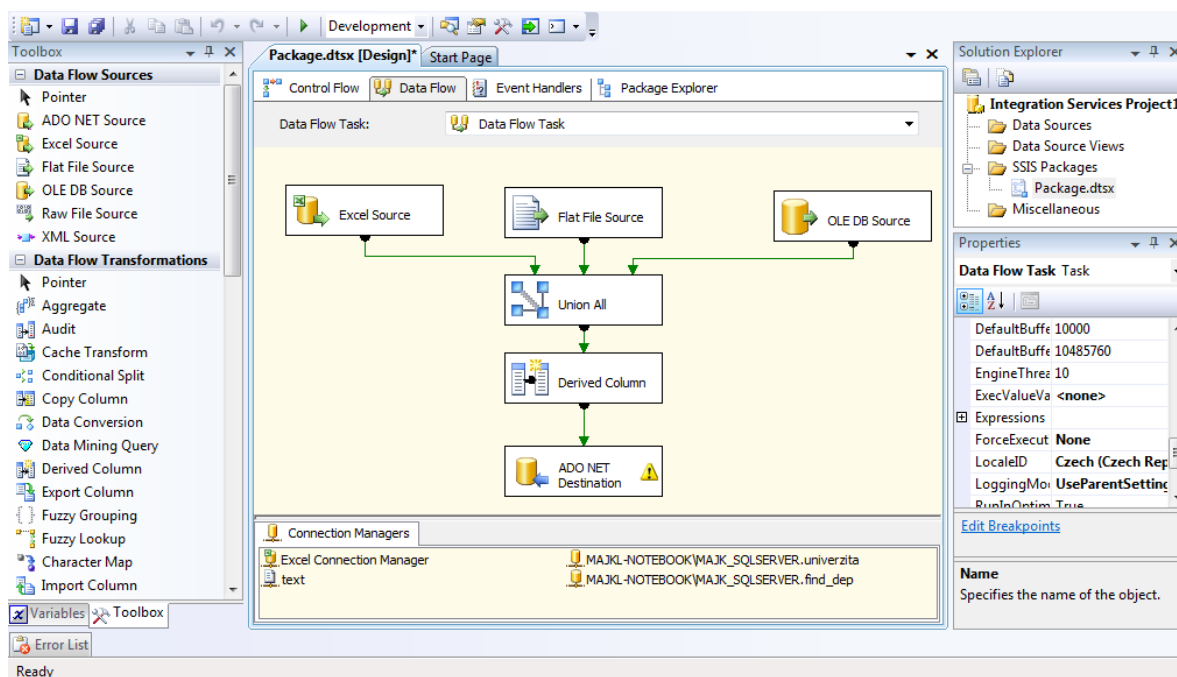
Prostředí BI Development Studia umožňuje spravovat více projektů najednou a nahlížet a upravovat vlastnosti jednotlivých objektů v projektech. (10)

#### **Projekt integračních služeb**

Prostředí integračního projektu Development Studia je rozděleno do čtyř hlavních částí. Solution Explorer umožňuje vytvářet a spravovat různé datové zdroje, pohledy, SQL Server Integration Services (SSIS) balíčky a další. V okně Properties se dají nastavovat vlastnosti jednotlivých objektů obsažených v SSIS balíčku. Toolbox nabízí všechny bloky použitelné v jednotlivých diagramech a jeho prvky se mění na základě aktivního diagramu. V neposlední řadě se zde nachází hlavní návrhové okno se 4 hlavními záložkami:

- Control Flow – návrh a sestavení procesů a úkolů, které je třeba při řešení problému vykonat.
- Data Flow – modelace datových zdrojů, transformací a datových úložišť.
- Event Handlers – nastavení událostí, které se mají vykonat v případě, že nastane během vykonávání nějaká událost (například odeslání e-mailu při selhání).
- Package Explorer – zobrazuje shrnutí obslužných rutin, proměnných, manažerů spojení atd., které se ve vytvořeném balíčku nacházejí.

Obrázek 14 - Prostředí BI Development Studio



*Zdroj: vlastní zpracování (screenshot)*

V záložce Data Flow je možné namodelovat celý proces ETL. Jako datové zdroje lze využít Excel, Raw, Flat File či XML soubory nebo databáze s připojením pomocí OLE DB nebo ADO .NET. Datová transformace nabízí mnoho prvků pro pokročilou transformaci dat, jako je sjednocování dat z více zdrojů, hledání a odstraňování duplicit (včetně hledání „neostrých“ duplicit pomocí Fuzzy logiky), konverze datových typů, řazení dat, multicasting, podmíněné třídění a mnoho dalších. Pro výstup je možné použít nejen stejné bloky jako ve zdrojích, ale i přímo přistoupit do analytických služeb.

### **Projekt reportovacích služeb**

Vytvořit report lze pomocí wizardu nebo ručně v návrhovém prostředí. Reportovací služby Business Intelligence Development Studia umožňují z dat z jakýchkoliv databází vytvářet přehledné reporty v podobě různých tabulek, grafů, ukazatelů nebo zapsat data v podobě agregovaných údajů přímo do map (v základu jsou pouze mapy USA). Zdrojovými daty mohou být celé tabulky nebo pohledy na ně, ale i data vybraná SQL dotazem. Typy grafů,

keré program nabízí, jsou stejné jako v Excelu. Hotové sestavy je možné exportovat do souborů Microsoft Office, CSV souborů, PDF, XML, HTML a dalších.

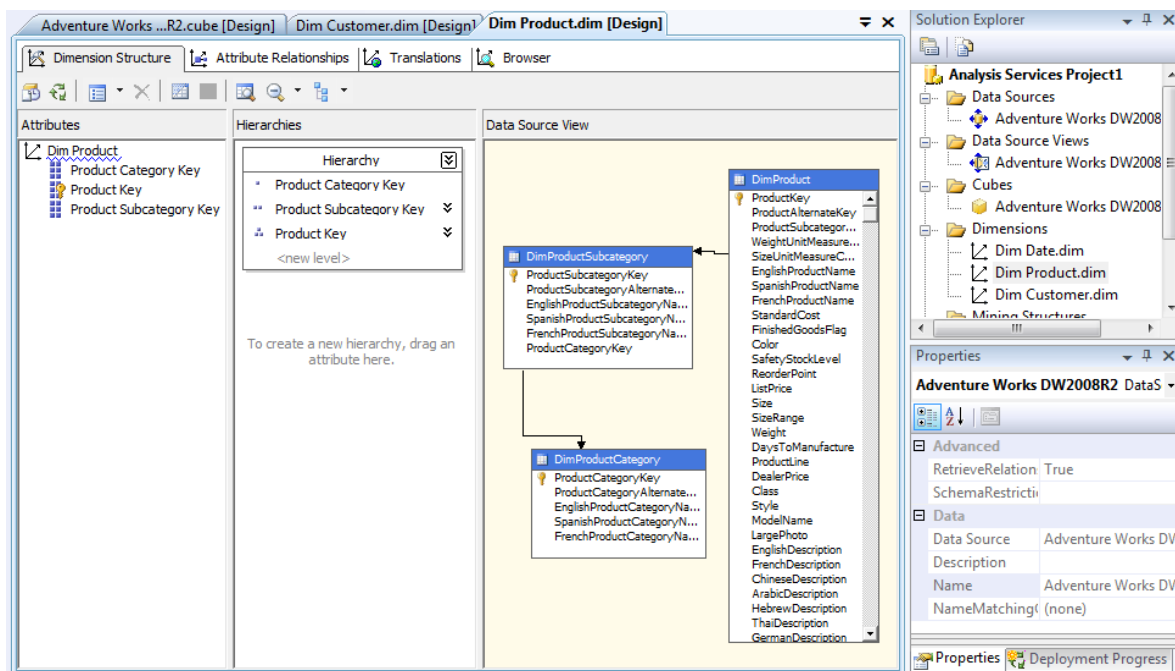
## Projekt analytických služeb

Prostředí analytických služeb je uzpůsobeno vytváření multidimenzionálních datových kostek a dataminingových modelů.

## OLAP kostky

Při vytváření multidimenzionální datové kostky je třeba určit zdroje dat a pohledy na ně. Dalším krokem je definice a sestavení hierarchií jednotlivých dimenzí. Poslední krok je sestavení kostky, určení tabulek dimenzí a tabulek faktů a samotný návrh zobrazení datové kostky.

Obrázek 15 - Hierarchie dimenzí v BI Development Studio



Zdroj: vlastní zpracování (screenshot)

K sestavení tabulek dimenzí je možné použít existující tabulky nebo vygenerovat nové časové i nečasové dimenze. Dalším důležitým krokem je určení klíčových atributů a vztahů

mezi nimi. Klíčový atribut nemusí být nutně v kostce zobrazován, a tak je ještě možné definovat, jaké atributy budou pro snazší orientaci v kostce zobrazeny. Například místo ID města, které by nám nic neřeklo, bude v kostce zobrazen jeho název.

Výsledná multidimenzionální datová kostka se zobrazuje v podobě kontingenční tabulky a stejně tak se s ní i pracuje. Z kostky je možné s použitím reportovacích služeb získávat data i do reportů.

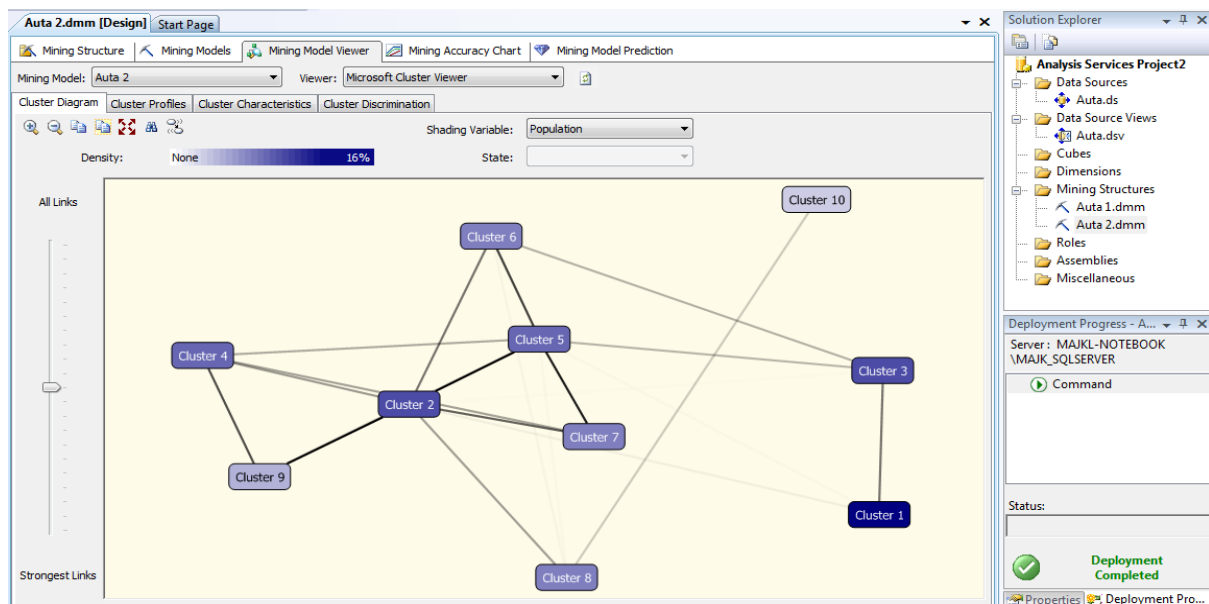
## **Data Mining**

Stejně jako u vytváření multidimenzionálních OLAP kostek i zde je třeba definovat datové zdroje a pohledy na data. Dalším úkolem je výběr algoritmu pro Data Mining, podle kterého se mění i hlavní okno programu. Microsoft Business Intelligence Development Studio nabízí tyto typy algoritmů:

- asociační pravidla
- shlukování
- rozhodovací stromy
- lineární regrese
- logická regrese
- algoritmus Naive Bayes
- neuronové sítě
- sekvenční shlukování
- časové řady



Obrázek 16 - Shlukovací algoritmus



Zdroj: vlastní zpracování (screenshot)

## 4. VLASTNÍ NÁVRH ŘEŠENÍ, PŘÍNOS NÁVRHŮ ŘEŠENÍ

Po analýze problému, popisu současné situace a základních znalostí problematiky jsem měl dostatek informací pro vlastní návrh řešení.

### 4.1. Integrovaní služby

Pomocí integračních služeb je zajištěno nahrání vstupních dat z textových a Excel souborů na SQL server v podobě tabulek, které budou vstupem pro natrénování dataminingového modelu a predikci dat.

#### 4.1.1. Vytvoření cílových tabulek na SQL serveru

Pokud se k integračním službám přistupuje prostřednictvím BI Development Studia, je aplikace schopna sama rozpoznat strukturu dat a navrhnout datové typy atributů a jejich délky. V případě, že k integračním službám přistupujeme například pomocí programovacího jazyka C#, neexistuje žádná funkce, která by automatické vytvoření tabulky umožňovala, a tak je tento problém potřeba naprogramovat.

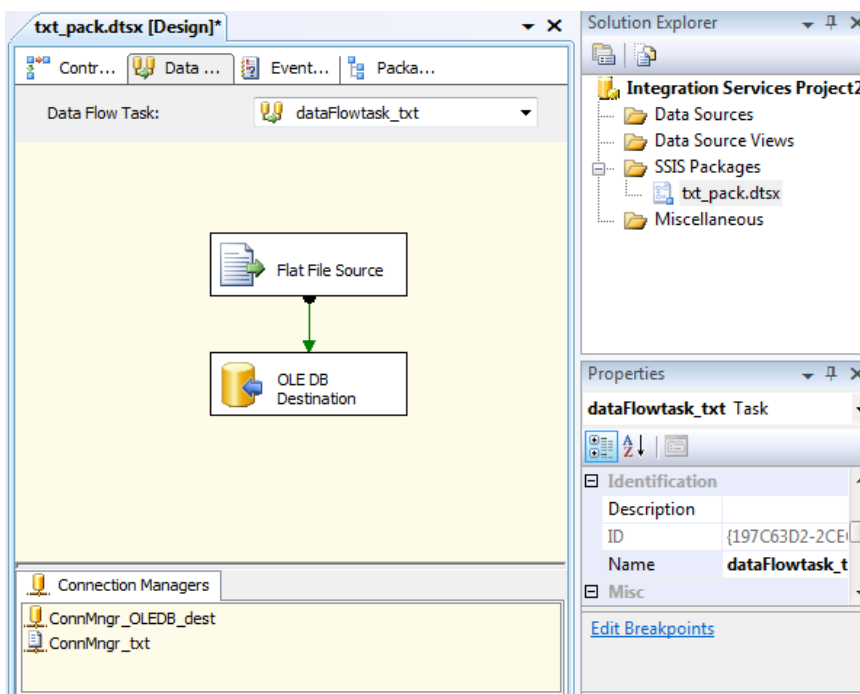
Pokud tedy budou splněny všechny požadavky na přípravu dat z kapitoly 4.2., budou na druhém řádku vstupních dat již reálné údaje, jež chceme do databáze uložit. Protože v analýze dat byly identifikovány sloupce obsahující text nebo číslo, stačí SQL dotaz generovat právě pro tyto dva datové typy.

```
string qSQL_newTable = "Create table " + textNameTable.Text + "(";  
double Num;  
int _i = 0;  
foreach (string name in srcColumns)  
{ qSQL_newTable += name;  
  bool isNum = double.TryParse(srcDTypes[_i], out Num);  
  _i += 1;  
  if (isNum)  
  {qSQL_newTable += " float NULL, "};  
  else  
  {qSQL_newTable += " varchar (" + FileCount.ToString() + "), "};  
}  
qSQL_newTable = qSQL_newTable.Substring(0, qSQL_newTable.Length - 2);  
qSQL_newTable += ")";
```

## 4.1.2. Načítání dat ze souborů Flat File

Pro načtení dat z textových souborů (CSV, Fixed Width) je třeba vytvořit SSIS balíček, který bude obsahovat dva bloky a dva manažery spojení.

Obrázek 17 - SSIS pro textový soubor



*Zdroj: vlastní zpracování*

### Manažer spojení ConnMgr\_txt

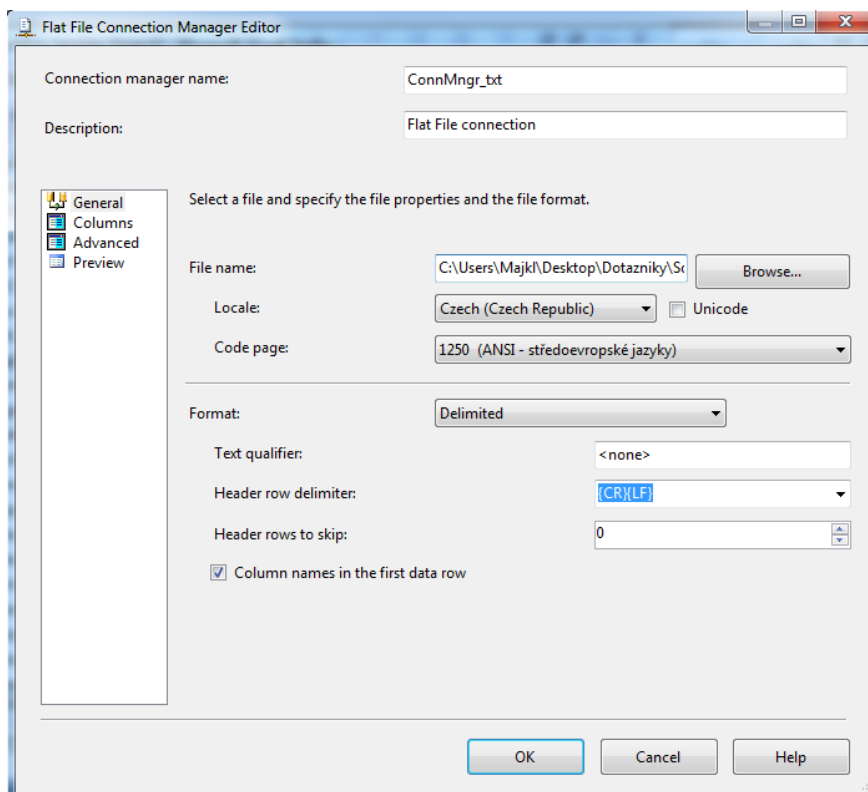
Kromě cesty ke zdrojovému souboru, jeho jazyku a kódovací stránky je zde nutné vybrat jeho formát. Výsledný program v této práci umožňuje načítání dat ze dvou souborových formátů – Delimited (CSV) a Fixed Width (s pevnou délkou). Program také musí obdržet informace, zda se v prvním řádku nacházejí názvy sloupců, v případě Fixed Width šířku dat a v případě Delimited oddělovač dat. Dále je třeba určit datové typy a délky jednotlivých sloupců a data namapovat.

```

Start.GlobalVars.CM_txt =
    Start.GlobalVars.Package_txt.Connections.Add("Flatfile");
Start.GlobalVars.CM_txt.Name = "ConnMgr_txt";
Start.GlobalVars.CM_txt.ConnectionString = Start.GlobalVars.PathSourceFile;
Start.GlobalVars.CM_txt.Description = "Flat File Connection";
Start.GlobalVars.CM_txt.Properties["Format"].
    SetValue(Start.GlobalVars.CM_txt, FileFormat);
Start.GlobalVars.CM_txt.Properties["HeaderRowDelimiter"].
    SetValue(Start.GlobalVars.CM_txt, "{CR}{LF}");
Start.GlobalVars.CM_txt.Properties["ColumnNamesInFirstDataRow"].
    SetValue(Start.GlobalVars.CM_txt, true);

```

Obrázek 18 - Flat File Connection Manager



*Zdroj: vlastní zpracování*

## Blok Flat File Source

Tento blok načte data podle parametrů uložených v manažeru spojení ConnMgr\_txt.

```

IDTSComponentMetaData100 FlatFile_txt =
    dataFlowTask_txt.ComponentMetaDataCollection.New();
FlatFile_txt.ComponentClassID = "DTSAdapter.FlatFileSource";
FlatFile_txt.Name = "Source_txt";

FlatFile_txt.RuntimeConnectionCollection[0].ConnectionManagerID =
    Start.GlobalVars.CM_txt.ID;
FlatFile_txt.RuntimeConnectionCollection[0].ConnectionManager =
    Microsoft.SqlServer.Dts.Runtime.DtsConvert.GetExtendedInterface(
        Start.GlobalVars.CM_txt);

```

## Manažer spojení ConnMngr\_OLEDB\_Dest

Tento manažer vytvoří spojení s SQL serverem a vybranou databází.

```

ConnectionManager CM_OLEDB_dest =
    Start.GlobalVars.Package_txt.Connections.Add("OLEDB");
CM_OLEDB_dest.Name = "ConnMngr_OLEDB_dest";
CM_OLEDB_dest.ConnectionString = Start.GlobalVars.SrvConnString +
    "Provider=SQLNCLI10;";
MessageBox.Show(Start.GlobalVars.SrvConnString + "Provider=SQLNCLI10;");
CM_OLEDB_dest.Description = "OLEDB Connection";

```

## Blok OLE DB Destination

V tomto bloku je potřeba vybrat existující tabulku v databázi nebo vytvořit novou tabulku, do níž se data uloží.

```

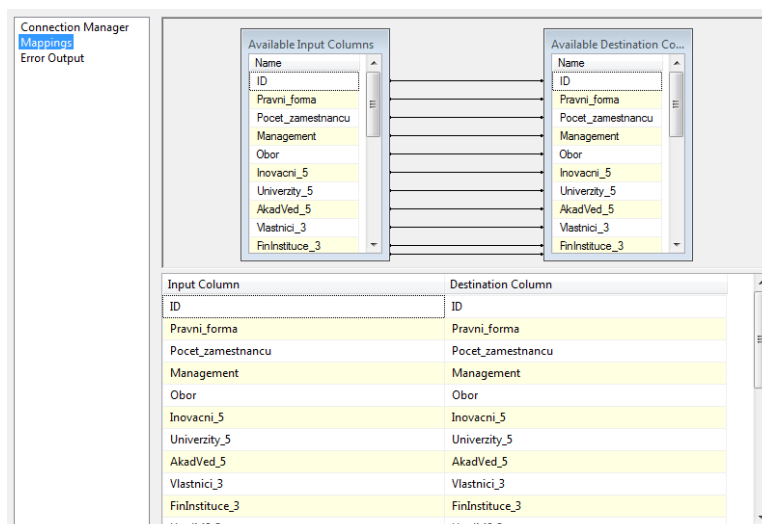
IDTSComponentMetaData100 OLEDB_Dest =
    dataFlowTask_txt.ComponentMetaDataCollection.New();
OLEDB_Dest.ComponentClassID = "DTSAdapter.OLEDBDestination";
OLEDB_Dest.Name = "OLEDB dest";

OLEDB_Dest.RuntimeConnectionCollection[0].ConnectionManagerID = CM_OLEDB_dest.ID;
OLEDB_Dest.RuntimeConnectionCollection[0].ConnectionManager =
    Microsoft.SqlServer.Dts.Runtime.DtsConvert.GetExtendedInterface(CM_OLEDB_dest);
inst_OLEDB.SetComponentProperty("OpenRowset", "[dbo].[ " +
    textNameTable.Text + " ]");

```

Aby bylo možné data namapovat, musí cílová tabulka odpovídat datovému zdroji. V opačném případě lze data mapovat ručně.

Obrázek 19 - Mapování výstupních sloupců



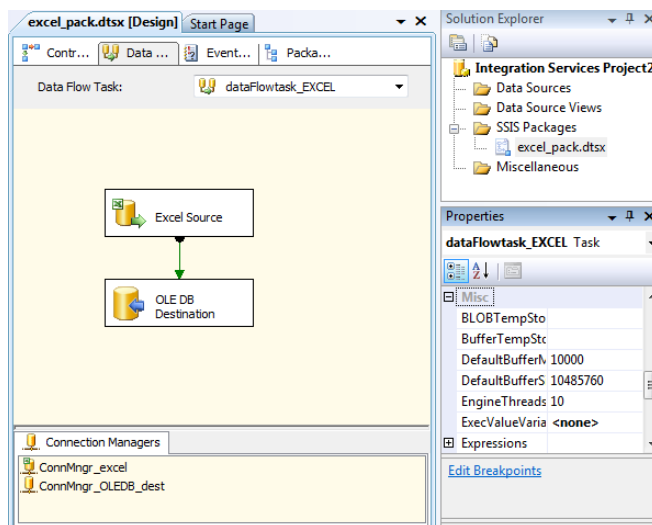
*Zdroj: vlastní zpracování*

Po nastavení obou bloků se balíček spustí a data se nahrají na SQL server.

#### 4.1.3. Načítání dat ze souborů Excel

V SSIS balíčku pro načítání dat z Excelu je výstupní blok (OLE DB) a manažer spojení stejný jako u Flat File souborů.

Obrázek 20 - SSIS pro Excel soubor



*Zdroj: vlastní zpracování*

## Manažer spojení ConnMngr\_excel

Oproti velkému množství informací, které je potřeba zadat při vytváření manažera spojení pro načítání dat z textových souborů, se zde zadává pouze cesta ke zdrojovému souboru a verze Excelu. Pro Microsoft Excel 2007 a vyšší se pro spojení používá provider Microsoft ACE OLEDB 12.0. Ke starším verzím Excelu se připojuje pomocí provideru Microsoft Jet OLEDB 4.0.

```
// EXCEL CONNECTION STRING
ExcelConnStr = "Provider=" + excel_provider + "Data Source=" + textFilePath.Text
+ @";Extended Properties="" + excel_ver + @";HDR=YES;IMEX=1""";

// SOURCE CONNECTION MANAGER EXCEL
ConnectionManager CM_excel = Package_Excel.Connections.Add("EXCEL");
CM_excel.Name = "ConnMngr_excel";
CM_excel.ConnectionString = ExcelConnStr;
```

## Blok Excel Source

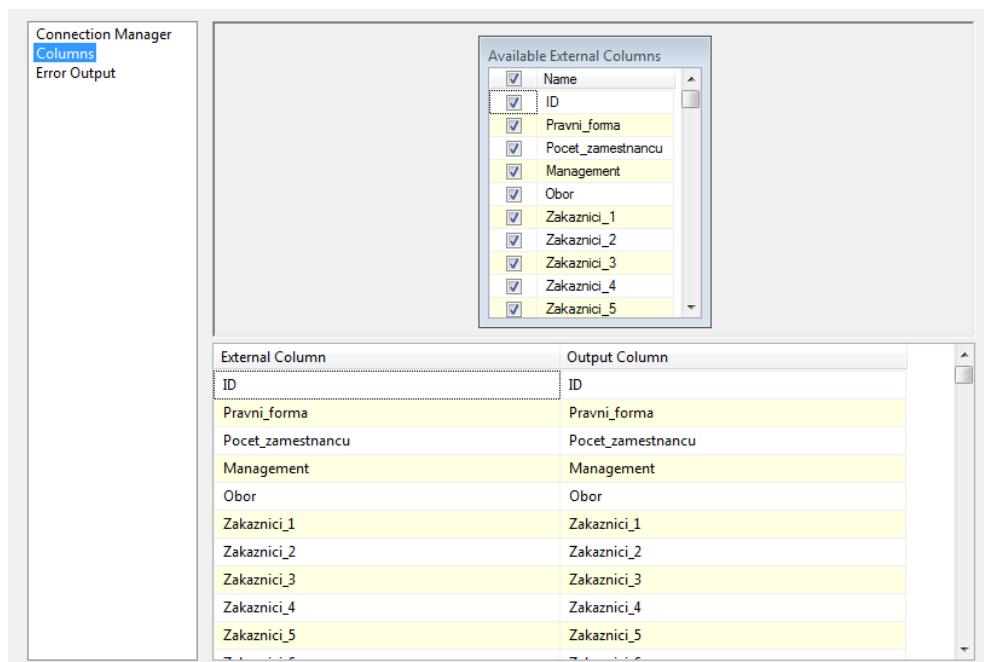
V tomto bloku je po uživateli požadován výběr listu ze zdrojového Excel souboru.

```
IDTSComponentMetaData100 Excel_source =
    dataFlowTask_excel.ComponentMetaDataCollection.New();
Excel_source.ComponentClassID = "DTSAdapter.ExcelSource";
Excel_source.Name = "Source_Excel";

Excel_source.RuntimeConnectionCollection[0].ConnectionManagerID = CM_excel.ID;
Excel_source.RuntimeConnectionCollection[0].ConnectionManager =
    Microsoft.SqlServer.Dts.Runtime.DtsConvert.GetExtendedInterface(CM_excel);
inst_excel.SetComponentProperty("AccessMode", 0);
inst_excel.SetComponentProperty("OpenRowset", comboExcelList.Text.ToString()
+ "$");
```

Pak si uživatel může vybrat, které sloupce z tabulky chce nahrát do databáze SQL.

Obrázek 21 - Výběr zdrojových sloupců



*Zdroj: vlastní zpracování*



## 4.2. Analytické služby – Data Mining

V této části je popsán celý postup od vytvoření a natrénování dataminingového modelu až po samotnou predikci dat.

### 4.2.1. Analytický projekt

Prvním krokem je založení nového projektu, který v podstatě představuje databázi na analytickém serveru (Microsoft Analysis Server). Na analytickém serveru již není možné používat klasické DML dotazy z databázového serveru, ale je nutné použít speciální příkazy pro OLAP (MDX) a data mining (DMX). Příkazy DMX neobsahují žádný příkaz, který by byl ekvivalentní příkazu create database v jazyce DML. Proto se nový projekt musí vytvořit pomocí jazyka XMLA (Extensible Markup Language for Analysis) nebo skriptem v jazyce C#.

```
// XMLA
<Create
xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">
  <ObjectDefinition>
    <Database xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      <ID>PREDIKCE_DOTAZNIKY</ID>
      <Name>PREDIKCE_DOTAZNIKY</Name>
      <Language>1033</Language>
      <Collation>Czech_CI_AS</Collation>
      <DataSourceImpersonationInfo>
        <ImpersonationMode>Default</ImpersonationMode>
      </DataSourceImpersonationInfo>
    </Database>
  </ObjectDefinition>
</Create>

// C#
// NOVÁ ANALYTICKÁ DATABÁZE
Database Anal_Database = new Database();
Anal_Database = Anal_Server.Databases.Add(Anal_Server.Databases.
  GetNewName("C#_PREDIKCE_DOTAZNIKY"));
Anal_Database.Update();
```

Jak je vidět z předcházejícího skriptu, vytváření objektů pomocí XMLA je poněkud komplikované, a proto se dále v práci budu zaměřovat pouze na jazyk C# a DMX.

### 4.2.2. Data Source

Druhým krokem je vytvoření nového zdroje dat. Jako zdroj dat je potřeba vybrat databázi z databázového serveru, která obsahuje trénovací a predikční (vstupní) tabulky.

```
// NOVÝ DATASOURCE
RelationalDataSource Anal_DataSource = new RelationalDataSource();
Anal_DataSource =
    Anal_Database.DataSources.Add(Anal_Server.Databases.GetNewName("DATASOURCE"));
Anal_DataSource.ConnectionString = Start.GlobalVars.SrvConnString;
Anal_DataSource.Update();
```

### 4.2.3. Data Source View

Při vytváření datového pohledu se již vybírají konkrétní tabulky a atributy. Je třeba vytvořit připojení k databázovému serveru a pomocí adaptéru provést SQL příkaz pro výběr dat. Program je navržen tak, aby vytvořil pohled na data z jedné konkrétní tabulky na databázovém serveru (SELECT \* FROM [název\_tabulky]). Dále je nutné nastavit vlastnosti tabulky – typ, databázové schéma, název a alias.

```
// NOVÝ DATASOURCE VIEW
DataSourceView Anal_View = new DataSourceView();
Anal_View = Anal_Database.DataSourceViews.
    Add(Anal_Database.DataSourceViews.GetNewName("DATASOURCE_VIEW"));
Anal_View.DataSourceID = Anal_DataSource.ID;
Anal_View.Schema = new DataSet();
string commSQL_SelectALL = "SELECT * FROM [dbo].[" +
    listTables.SelectedItem.ToString() + "];";
OleDbDataAdapter SQL_adapter = new OleDbDataAdapter(commSQL_SelectALL,
    Start.GlobalVars.SrvConnString + ";Provider=SQLOLEDB.1;");

DataTable[] dataTables = SQL_adapter.FillSchema(Anal_View.Schema,
    SchemaType.Mapped, listTables.SelectedItem.ToString());
DataTable dataTable = dataTables[0];

dataTable.ExtendedProperties.Add("TableType", "Table");
dataTable.ExtendedProperties.Add("DbSchemaName", "dbo");
dataTable.ExtendedProperties.Add("DbTableName",
    listTables.SelectedItem.ToString());
dataTable.ExtendedProperties.Add("FriendlyName",
    listTables.SelectedItem.ToString());
Anal_View.Update();
```

#### 4.2.4. Mining Structure

V Mining Structure se definují atributy z datového pohledu, které budou dostupné pro dataminingové modely. U všech atributů je nutné definovat datový typ a typ obsahu dat (spojitý, diskrétní, diskretizovaný, cyklický, pořadový). Nakonec se do parametru Holdout zadá maximální množství testovacích případů nebo obsah testovací množiny v procentech. Všechny případy se podle této podmínky náhodně rozdělí na testovací a trénovací množinu.

```
CREATE MINING STRUCTURE DOTAZNIKY_STRUCTURE
(
  [ID] DOUBLE KEY,
  Inovacni_5 DOUBLE DISCRETE,
  Univerzity_5 DOUBLE DISCRETE,
  AkadVed_5 DOUBLE DISCRETE,
  Vlastnici_3 DOUBLE DISCRETE,
  FinInstitute_3 DOUBLE DISCRETE,
  UradMS_3 DOUBLE DISCRETE,
  Univerzity_16 DOUBLE DISCRETE,
  Konkurence_16 DOUBLE DISCRETE,
  Zakaznici_2 DOUBLE DISCRETE,
  Dodavatele_2 DOUBLE DISCRETE,
  Poradna_2 DOUBLE DISCRETE,
  T_INOVACE TEXT DISCRETE,
  T_FINANCE TEXT DISCRETE,
  T_HR TEXT DISCRETE,
  T_KONTAKTY TEXT DISCRETE,
  T_PORADNA TEXT DISCRETE
)
WITH HOLDOUT(30 PERCENT OR 200 CASES) REPEATABLE(0)
```

#### 4.2.5. Mining Model

V návrhu dataminingového modelu se v první řadě určuje algoritmus pro data mining. Na řešení tohoto problému jsem - pro jejich jednoduchost, přehlednost a možnost zobrazení nalezených závislostí - vybral rozhodovací stromy (Microsoft Decision Trees). V modelu je také potřeba určit, které atributy budou vstupní, a které predikované.

```

ALTER MINING STRUCTURE DOTAZNIKY_STRUCTURE
ADD MINING MODEL DOTAZNIKY_MODEL
(
  [ID],
  Inovacni_5,
  Univerzity_5,
  AkadVed_5,
  Vlastnici_3,
  FinInstitute_3,
  UradMS_3,
  Univerzity_16,
  Konkurence_16,
  Zakaznici_2,
  Dodavatele_2,
  Poradna_2,
  T_INOVACE PREDICT,
  T_FINANCE PREDICT,
  T_HR PREDICT,
  T_KONTAKTY PREDICT,
  T_PORADNA PREDICT
)
USING Microsoft_Decision_Trees

```

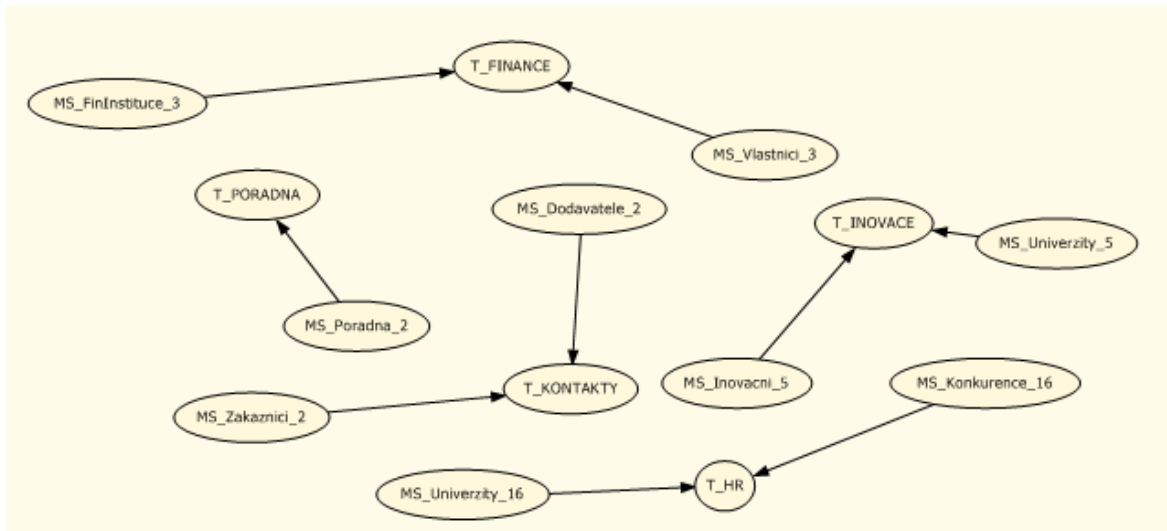
#### 4.2.6. Výsledky Mining Modelu

Po vytvoření Mining Modelu a jeho natrénování je možné sledovat jeho různé charakteristiky, včetně charakteristik jednotlivých predikovaných veličin.

##### 4.2.6.1. Síť závislostí (Dependency Network)

Síť závislostí zobrazuje, na kterých vstupních atributech jsou predikované veličiny závislé. Diagram umožňuje identifikovat závislosti od nejsilnější po nejslabší. Například T\_FINANCE má silnější vazbu s Vlastnici\_3 než FinInstitute\_3. To znamená, že výsledná hodnota predikované veličiny se z větší části řídí daty Vlastnici\_3.

Obrázek 22 - Dependency Network

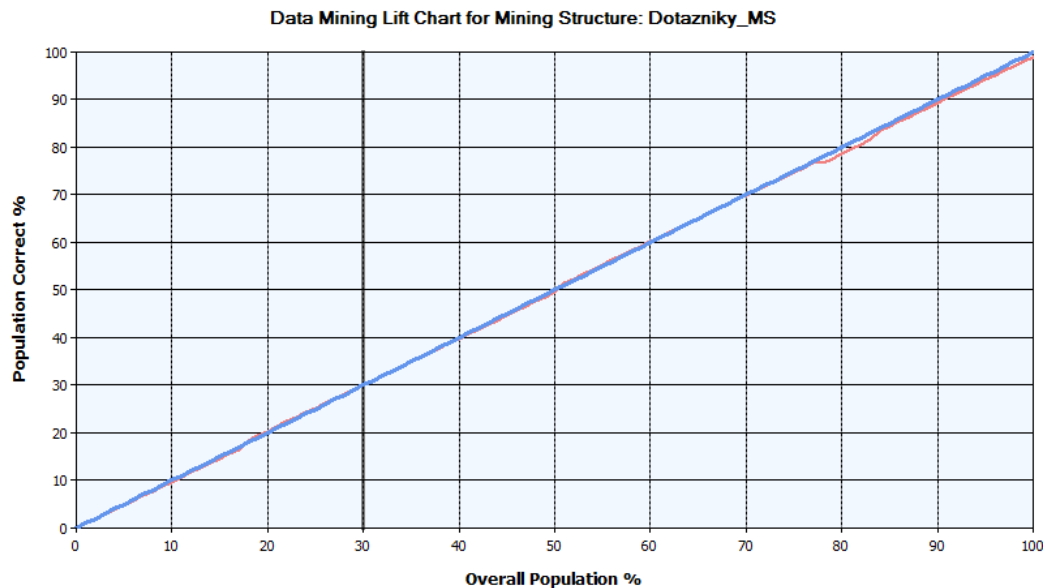


Zdroj: vlastní zpracování

#### 4.2.6.2. Lift Chart

Lift Chart zobrazuje, zda je model správně natrénován. Na následujícím obrázku je vidět, že model téměř kopíruje křivku ideálního modelu.

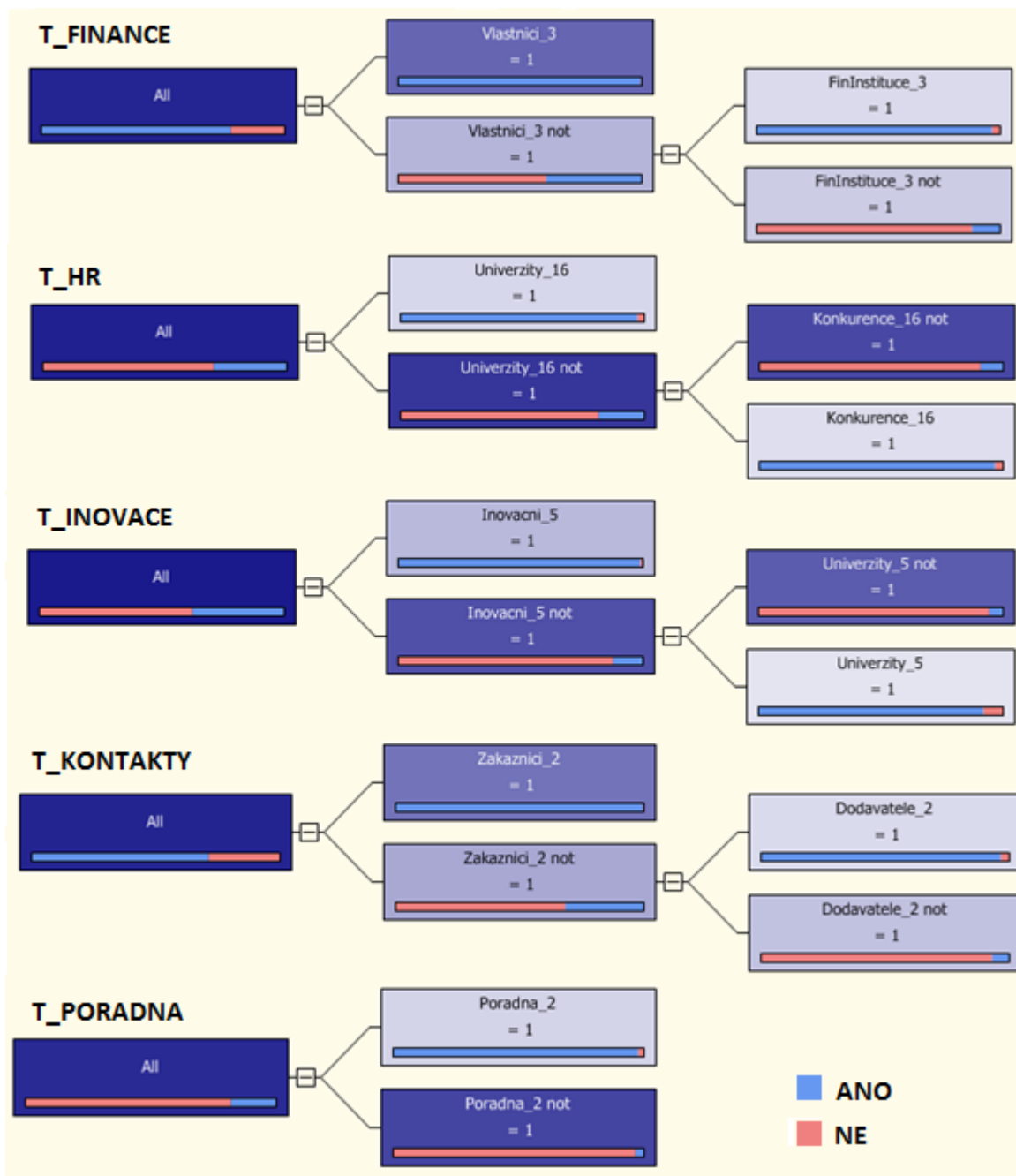
Obrázek 23 - Lift Chart



Zdroj: vlastní zpracování

### 4.2.6.3. Rozhodovací stromy

Obrázek 24 - Nalezené rozhodovací stromy



Zdroj: vlastní zpracování

V rozhodovacích stromech je znázorněno rozložení všech odpovědí z trénovací tabulky a také závislost výsledku na jednotlivých atributech. Modrá barva popisuje stav „ANO“, červená barva stav „NE“.

Například u stromu T\_INOVACE je po natrénování modelu přibližně 38% pravděpodobnost, že výsledek bude ANO. Z toho, pokud odpověď na otázku Inovacni\_5 bude 1, je 98 % pravděpodobnost ANO. Pokud odpověď na otázku Inovacni\_5 nebude rovna 1, řídí se výsledek podle odpovědí na otázku Univerzity\_5 (12 % ANO, 88 % NE).

#### 4.2.7. Predikce

```

SELECT
  t.[ID],t.[Pocet_zamestnancu],t.[Obor],t.[Management],
  t.[Pravni_forma],
  (Predict([Dotazniky_MM].[MM_T_FINANCE])) as [FINANCE],
  (Predict([Dotazniky_MM].[MM_T_HR])) as [HR],
  (Predict([Dotazniky_MM].[MM_T_INOVACE])) as [INOVACE],
  (Predict([Dotazniky_MM].[MM_T_KONTAKTY])) as [KONTAKTY],
  (Predict([Dotazniky_MM].[MM_T_PORADNA])) as [PORADNA]
From [Dotazniky_MM]
PREDICTION JOIN
  OPENQUERY ([DATASOURCE],
    'SELECT
      [ID],[Pocet_zamestnancu],[Obor],[Management],[Pravni_forma],
      [Inovacni_5],[Univerzity_5],[AkadVed_5],[Vlastnici_3],
      [FinInstitute_3],[UradMS_3],[Univerzity_16],[Konkurence_16],
      [Zakaznici_2],[Dodavatele_2],[Poradna_2],''
      [T_INOVACE],[T_FINANCE],[T_HR],[T_KONTAKTY],[T_PORADNA]
    FROM
      [dbo].[PREDIKCNI]') AS t
ON
  [Dotazniky_MM].[Inovacni_5] = t.[Inovacni_5] AND
  [Dotazniky_MM].[Univerzity_5] = t.[Univerzity_5] AND
  [Dotazniky_MM].[AkadVed_5] = t.[AkadVed_5] AND
  [Dotazniky_MM].[Vlastnici_3] = t.[Vlastnici_3] AND
  [Dotazniky_MM].[FinInstitute_3] = t.[FinInstitute_3] AND
  [Dotazniky_MM].[UradMS_3] = t.[UradMS_3] AND
  [Dotazniky_MM].[Univerzity_16] = t.[Univerzity_16] AND
  [Dotazniky_MM].[Konkurence_16] = t.[Konkurence_16] AND
  [Dotazniky_MM].[Zakaznici_2] = t.[Zakaznici_2] AND
  [Dotazniky_MM].[Dodavatele_2] = t.[Dodavatele_2] AND
  [Dotazniky_MM].[Poradna_2] = t.[Poradna_2] AND
  [Dotazniky_MM].[MM_T_INOVACE] = t.[T_INOVACE] AND
  [Dotazniky_MM].[MM_T_FINANCE] = t.[T_FINANCE] AND
  [Dotazniky_MM].[MM_T_HR] = t.[T_HR] AND
  [Dotazniky_MM].[MM_T_KONTAKTY] = t.[T_KONTAKTY] AND
  [Dotazniky_MM].[MM_T_PORADNA] = t.[T_PORADNA]

```

Při predikci se proti sobě staví natrénovaný dataminingový model a predikční (vstupní) tabulka obsahující nové odpovědi na dotazníky. Podle těchto nových odpovědí a podle nalezených závislostí se predikuje, zda bude odpověď na obchodní otázku ano nebo ne.

Výsledky predikce jsou nakonec uloženy do uživatelem zvoleného Excel souboru (viz Příloha 1).

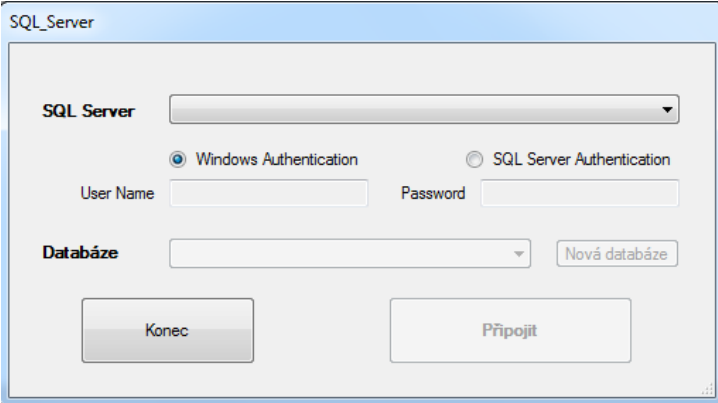
### 4.3. Vytvořená aplikace

Aplikace byla vytvořena nad platformou .NET v prostředí Microsoft Visual Studio 2010 a pro její funkčnost je nutné mít dostupný Microsoft SQL Server s nainstalovanými databázovými, integračními a analytickými službami. Aplikace byla testována na Microsoft SQL Server 2008 R2.

#### 4.3.1. Okno SQL\_Server

Formulář SQL\_server slouží pro připojení k severu. Uživatelé jsou nabídnuty všechny dostupné SQL servery v místní síti. Dále musí uživatel zvolit typ autentizace a vybrat si konkrétní databázi na serveru nebo vytvořit databázi novou.

Obrázek 25 - Okno SQL\_Server



The screenshot shows a dialog box titled "SQL\_Server". It contains the following elements:

- A dropdown menu labeled "SQL Server".
- Two radio buttons: "Windows Authentication" (which is selected) and "SQL Server Authentication".
- Two input fields: "User Name" and "Password".
- A dropdown menu labeled "Databáze" and a button labeled "Nová databáze".
- Two buttons at the bottom: "Konec" and "Připojit".

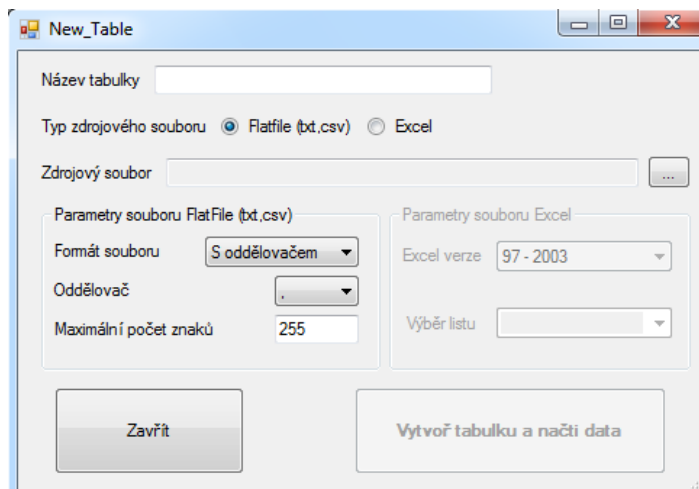
*Zdroj: vlastní zpracování*



### 4.3.2. Okno New\_Table

Toto okno uživateli umožňuje vytvoření nové tabulky v databázi a její naplnění daty. Uživatel může po zadání parametrů nahrát data z textových nebo Excel souborů.

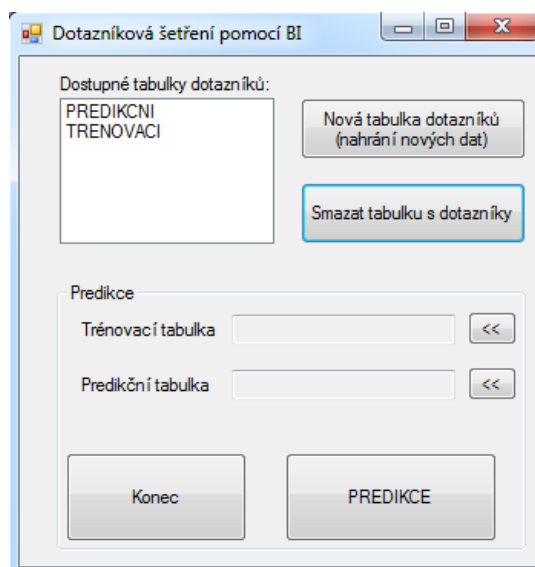
Obrázek 26 - Okno New\_Table



*Zdroj: vlastní zpracování*

### 4.3.3. Hlavní okno programu

Obrázek 27 - Hlavní okno programu



*Zdroj: vlastní zpracování*

V hlavním okně programu jsou zobrazeny všechny dostupné tabulky ve vybrané databázi na SQL serveru. Aplikace umožňuje vytvořit novou tabulku (okno New\_table) nebo tabulku smazat. Po výběru trénovací a predikční (vstupní) tabulky může uživatel přistoupit k samotné predikci. Výsledky predikce budou uloženy do vybraného Excel souboru.

#### 4.4. Interpretace výsledků – přínosy

Výsledky predikcí z jednotlivých dostupných dotazníků jsou zobrazeny v příloze 1.

Tabulka 4 - Celkové výsledky predikce

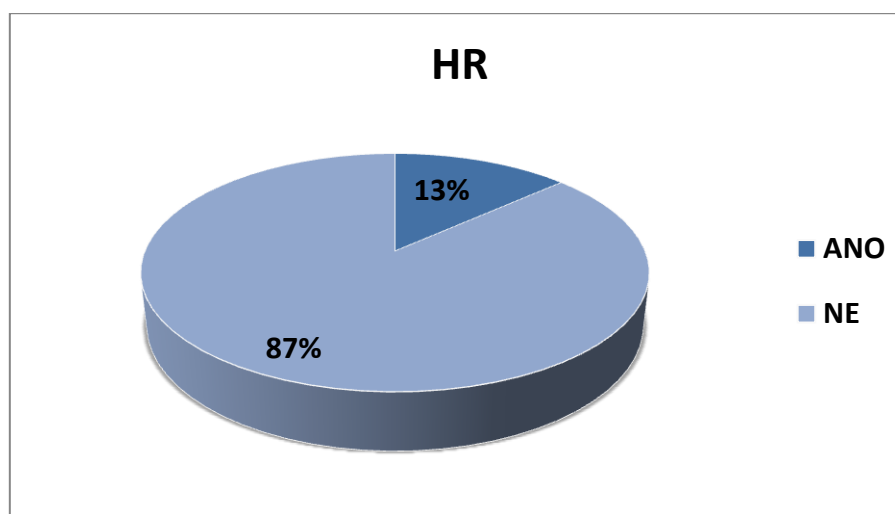
	FINANCE	HR	INOVACE	KONTAKTY	PORADENSKÁ ČINNOST
<b>ANO</b>	277	46	127	254	64
<b>NE</b>	67	298	217	90	280
<b>Celkem</b>	344	344	344	344	344

*Zdroj: vlastní zpracování*

##### 4.4.1. HR - Hledají oslovené firmy lidské zdroje u konkurence a na univerzitách?

Z následujícího grafu je vidět, že pouze 13 % oslovených firem hledá nebo chce hledat nové zaměstnance z řad studentů nebo u konkurence.

Graf 1 - Výsledky HR



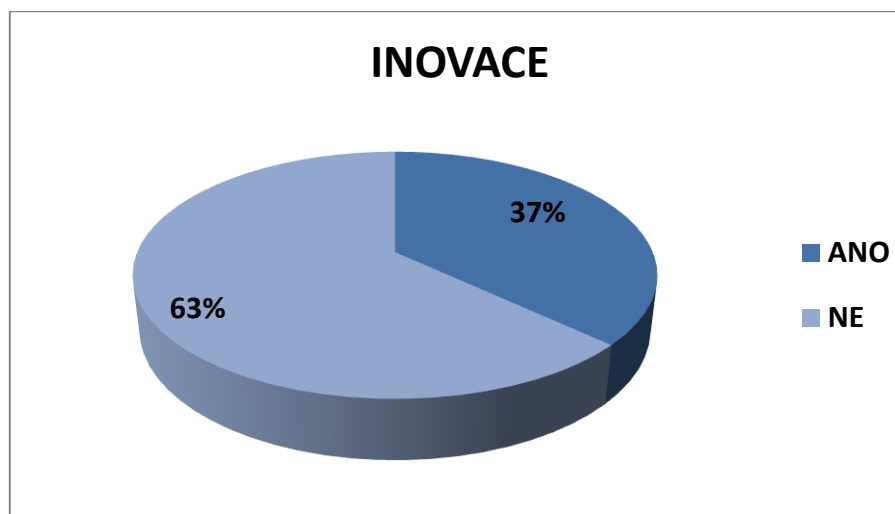
*Zdroj: vlastní zpracování*

Tato nízká čísla mohou být způsobena tím, že firmy často hledají nové zaměstnance na základě doporučení vlastních zaměstnanců. Tento způsob je pro firmu výhodný v tom, že je levný a časově méně náročný. Společnosti také často nevyhledávají studenty, protože i u většiny nižších pozic vyžadují minimální praxi. Hledání zaměstnanců u konkurence zatím není v České republice příliš rozšířené a firmy, které takto chtějí hledat zaměstnance, musí mít velice silné finanční zázemí, aby byly schopny získat zaměstnance konkurence na svou stranu.

#### **4.4.2. INOVACE - Mají dotazované firmy zájem při inovacích spolupracovat s inovačními a výzkumnými centry, univerzitami a Akademií věd?**

Níže uvedený graf prezentuje, že 37% firem má zájem o spolupráci s externími univerzitami a inovačními a výzkumnými centry. Tato skutečnost by mohla být zapříčiněna nejspíš tím, že budování interních inovačních center si mohou dovolit pouze velké, finančně zajištěné společnosti. Tím vzniká potenciál (například pro univerzity) tyto firmy oslovit k vzájemné oboustranné spokojenosti. Firmy by mohly zajistit své inovační potřeby při nižších nákladech a univerzity by získaly finanční prostředky a praxi pro své studenty.

Graf 2 - Výsledky INOVACE



*Zdroj: vlastní zpracování*

Další informací, kterou lze z dat získat, je, na jaké firmy by se univerzity a inovační centra měly obrátit. Z následující tabulky vyplývá, že největší zájem o tuto spolupráci mají firmy působící v oborech výroba potravinářských výrobků, výroba a opravy strojů a výroba z kovu. Naopak nejmenší zájem je ze strany firem působících ve výrobě koksu, jaderných paliv a ropy a výroby z usní.

**Tabulka 5 - Inovace v závislosti na jednotlivých oborech**

<b>Obor podnikání</b>	<b>ANO</b>
Výroba potravinářských výrobků	20
Výroba a opravy strojů	18
Výroba z kovu	14
Jiné	12
Výroba elek. a optic. přístrojů	11
Zpracovatelský průmysl	11
Výroba dopravních prostředků	10
Zpracování dřeva	6
Výroba plastových výrobků	6
Výroba textilií	5
Výroba chemických výrobků	5
Výroba ostatních nekovových výrobků	4
Výroba papíru a vlákniny	2
Výroba z usní	1
Koks, jaderná paliva, ropa	1

*Zdroj: vlastní zpracování*

## ZÁVĚR

Cílem mé diplomové práce bylo vytvoření aplikace pro analýzu dotazníků zaměřených na stakeholdery. Aplikace měla umožňovat nahrání dat z různých zdrojů na SQL server a následně identifikovat závislosti a predikovat výstupní veličiny. Pro tento účel mělo být využito integračních a analytických služeb architektury Business Intelligence.

Na základě teoretických poznatků a analýzy problému byla vytvořena aplikace odpovídající všem výše uvedeným požadavkům.

Tato diplomová práce plní svůj cíl a vytvořená aplikace umožňuje hledání závislostí a predikci dat ze zdrojových dotazníků.

V případě potřeby by bylo možné z dat získat další prospěšné informace. Nejdůležitější činností by však bylo zjištění a sestavení požadavků na informace, které by manažeri nebo vedoucí pracovníci chtěli z dat získat. Přizpůsobení aplikace, aby splňovala tyto nové požadavky, by už bylo jednoduché a časové nenáročné.

## SEZNAM POUŽITÉ LITERATURY

- (1) ACREA. *IBM SPSS Statistics*. [online]. 2011 [cit. 2012-05-01]. Dostupné z: <http://www.acrea.cz/ibmspss-statistics.htm>.
- (2) BAADER COMPUTER. *Open Source manažerské informační systémy*. [online]. 2012 [cit. 2012-05-01]. Dostupné z: <http://www.bcpraha.cz/open-source-manazerske-informacni-systemy-pentaho-business-intelligence>.
- (3) BIGRESOURCE. *BigResource*. [online]. 2012 [cit. 2012-05-01]. Dostupné z: <http://www.bigresource.com/>.
- (4) BRUST, J. A., FORTE, S. *Mistrovství v programování SQL Serveru 2005. Tvorba databázových aplikací a řešení pro BI*. Brno : Computer Press, 2007. 833 s. ISBN 978-80-251-1607-4.
- (5) KOTLER, P., KELLER, K. L. *Marketing management*. Praha : Grada Publishing, 2007. 788 s. ISBN 978-80-247-1359-5.
- (6) KOTLER, P., WONG, V., SAUNDERS, J., AMSTRONG, G. *Moderní marketing*. Praha : Grada Publishing, 2007. 1041 s. ISBN 978-80-247-1545-2.
- (7) KOZEL, R. *Moderní marketingový výzkum*. Praha : Grada Publishing, 2006. 277 s. ISBN 978-80-247-0966-6.
- (8) KRŮŽ, J. *Business Intelligence*. [online]. 2012 [cit. 2012-05-01]. Dostupné z: [https://www.vutbr.cz/www\\_base/priloha.php?dpid=53997](https://www.vutbr.cz/www_base/priloha.php?dpid=53997).
- (9) LACKO, Ľ. *Business Intelligence v SQL Serveru 2008: reportovací, analytické a další datové služby*. Brno : Computer Press, 2009. 456 s. ISBN 978-80-251-2887-9.
- (10) MICROSOFT. *Microsoft TechNet*. [online]. 2012 [cit. 2012-05-01]. Dostupné z: <http://technet.microsoft.com/cs-cz/>.
- (11) MSDN. *Fóra*. [online]. 2012 [cit. 2012-05-01]. Dostupné z: <http://social.msdn.microsoft.com/Forums/cs-CZ/categories>.
- (12) MSDN. *Knihovna MSDN*. [online]. 2012 [cit. 2012-05-01]. Dostupné z: <http://msdn.microsoft.com/cs-cz/library/default.aspx>.
- (13) NOVOTNÝ, O., POUR, J., SLÁNSKÝ, D. *Business Intelligence: jak využít bohatství ve vašich datech*. Praha : Grada, 2005. 256 s. ISBN 80-247-1094-3.

- (14) PRICE, J. *C#: programování databází*. Praha : Grada Publishing, 2005. 623 s. ISBN 80-247-0982-1.
- (15) PŘIKRYLOVÁ, J., JAHODOVÁ, H. *Moderní marketingová komunikace*. Praha : Grada Publishing, 2010. 303 s. ISBN 978-80-247-3622-8.
- (16) STATSOFT. *Podpora. Statistica Visual Basic*. [online]. 2012 [cit. 2012-05-01]. Dostupné z: <http://www.statsoft.cz/podpora/statistica-visual-basic/>.
- (17) STATSOFT. *Produkty*. [online]. 2012 [cit. 2012-05-01]. Dostupné z: <http://www.statsoft.cz/produkty/seznam/>.
- (18) STUDENTSKÉ. *Primární – sekundární zdroje dat*. [online]. 2010 [cit. 2012-05-01]. Dostupné z: <http://management-marketing.studentske.eu/2010/09/6-primarni-sekundarni-zdroje-dat.html>.
- (19) TOMEK, G., VÁVROVÁ, V. *Marketing management*. Praha : ČVUT, 1999. 406 s. ISBN 80-01-01904-7.
- (20) ZAMAZALOVÁ, M. *Marketing*. Praha : C. H. Beck, 2010. 499 s. ISBN 978-80-740-0115-4.

## SEZNAM ZKRATEK

<b>ADO .NET</b>	ActiveX Data Objects for .NET
<b>BI</b>	Business Intelligence
<b>CRM</b>	Customer Relationship Management
<b>CSV</b>	Comma-Separated Values
<b>DM</b>	Data Mining
<b>DMA</b>	Data Mart
<b>DML</b>	Data Manipulation Language
<b>DMX</b>	Data Mining Extensions
<b>DSA</b>	Data Staging Area
<b>DWH</b>	Data Warehouse
<b>EAI</b>	Enterprise Application Integration
<b>EIS</b>	Executive Information System
<b>ERP</b>	Enterprise Resource Planning
<b>ETL</b>	Extraction, Transformation, Loading
<b>ETTL</b>	Extracting, Transferring, Transforming, and Loading
<b>HTML</b>	HyperText Markup Language
<b>ICT</b>	Information and Communication Technologies
<b>IS</b>	Information System
<b>IT</b>	Information Technology



<b>MDX</b>	MultiDimensional Expressions
<b>MIS</b>	Management Information System
<b>ODS</b>	Operational Data Source
<b>OLAP</b>	On-Line Analytical Processing
<b>OLE DB</b>	Object Linking and Embedding Database
<b>PDF</b>	Portable Document Format
<b>ROA</b>	Return on Assets
<b>ROE</b>	Return on Equity
<b>SQL</b>	Structured Query Language
<b>SSIS</b>	SQL Server Integration Services
<b>XML</b>	eXtensible Markup Language
<b>XMLA</b>	Extensible Markup Language for Analysis

## SEZNAM OBRÁZKŮ

Obrázek 1 - Proces marketingového výzkumu .....	15
Obrázek 2 - Marketingový informační systém .....	21
Obrázek 3 - Procentuální využití jednotlivých Business Intelligence technologií .....	22
Obrázek 4 - Architektura BI .....	23
Obrázek 5 - Návrh transformací, ETL .....	27
Obrázek 6 - Procesní schéma data miningu .....	29
Obrázek 7 - Životní cyklus reportování .....	31
Obrázek 8 - Náhled zdrojových dat .....	38
Obrázek 9 - Prostředí IBM SPSS Statistics .....	39
Obrázek 10 - Techniky Direct Marketing v SPSS Statistics .....	40
Obrázek 11 - Prostředí STATISTICA Data Miner .....	42
Obrázek 12 - Pentaho Business Analytics .....	43
Obrázek 13 - Zobrazení prodejů v Pentaho na mapě Google .....	44
Obrázek 14 - Prostředí BI Development Studio .....	46
Obrázek 15 - Hierarchie dimenzí v BI Development Studio .....	47
Obrázek 16 - Shlukovací algoritmus .....	49
Obrázek 17 - SSIS pro textový soubor .....	51
Obrázek 18 - Flat File Connection Manager .....	52
Obrázek 19 - Mapování výstupních sloupců .....	54
Obrázek 20 - SSIS pro Excel soubor .....	54
Obrázek 21 - Výběr zdrojových sloupců .....	56
Obrázek 22 - Dependency Network .....	61
Obrázek 23 - Lift Chart .....	61
Obrázek 24 - Nalezené rozhodovací stromy .....	62
Obrázek 25 - Okno SQL_Server .....	64
Obrázek 26 - Okno New_Table .....	65
Obrázek 27 - Hlavní okno programu .....	65

## **SEZNAM TABULEK**

Tabulka 1 - Uzavřené otázky .....	19
Tabulka 2 - Otevřené otázky.....	19
Tabulka 3 - balíčky STATISTICA 10 .....	41
Tabulka 4 - Celkové výsledky predikce .....	66
Tabulka 5 - Inovace v závislosti na jednotlivých oborech.....	68

## **SEZNAM GRAFŮ**

Graf 1 - Výsledky HR.....	66
Graf 2 - Výsledky INOVACE .....	67

## SEZNAM PŘÍLOH

Příloha 1 - Výsledky predikce .....	77
Příloha 2 - Zdrojový kód – Form SQL_Server .....	84
Příloha 3 - Zdrojový kód – Form New_DB .....	88
Příloha 4 - Zdrojový kód – Form New_Table .....	89
Příloha 5 - Zdrojový kód – Hlavní okno – Form Start.....	102

Příloha 1 - Výsledky predikce

ID	Počet zaměstnanců	Obor	Management	Právní forma	FINANCE	HR	INOVACE	KONTAKTY	PORADENSKÁ ČINNOST
1	20	dk	s	s	ANO	NE	NE	ANO	ANO
2	125	db	t	a	ANO	NE	NE	ANO	NE
3	25	dn	t	s	ANO	ANO	ANO	ANO	NE
4	125	dh	s	a	ANO	NE	ANO	ANO	NE
5		da	t	a	ANO	NE	NE	ANO	NE
6	105	dn	n	a	ANO	NE	ANO	ANO	NE
7		da		j	NE	NE	NE	NE	NE
8	12	dk	s	s	ANO	NE	NE	ANO	ANO
9	172	da	n	a	ANO	NE	ANO	NE	ANO
10	2	dk	t	s	NE	NE	NE	ANO	NE
11	50	dn	t	s	NE	NE	NE	NE	NE
12		dj	n	s	ANO	NE	NE	ANO	NE
13	150	da	s	s	ANO	NE	NE	NE	NE
14	85	dd	t	s	ANO	NE	NE	ANO	NE
15	150	da	s	a	ANO	NE	NE	ANO	NE
16	86	db	t	s	ANO	NE	NE	ANO	NE
17	20	dh	t	s	NE	NE	NE	ANO	NE
18	70	dm	t	a	ANO	ANO	ANO	ANO	NE
19	3	dl	n	s	ANO	NE	NE	NE	NE
20	32	dk	s	s	ANO	NE	NE	ANO	NE
21	215	dk	s	s	ANO	NE	NE	NE	ANO
22	13	di	t	s	ANO	NE	ANO	ANO	NE
23	30	dm	n	s	ANO	NE	NE	NE	ANO
24	35	dg	t	a	NE	NE	ANO	NE	ANO
25	1500	dk	n	s	ANO	NE	NE	ANO	NE
26	87	da	s	a	ANO	NE	NE	ANO	NE
27	430	dj	t	a	ANO	NE	ANO	NE	NE
28	121	dk	n	s	ANO	NE	NE	NE	NE
29	150	dj		s	ANO	NE	NE	ANO	ANO
30	90	dn	n	s	ANO	NE	ANO	ANO	ANO
31	215	dk	s	s	ANO	NE	NE	NE	ANO
32	40	dh	t	s	ANO	NE	ANO	ANO	ANO
33	3	j	s	j	ANO	NE	ANO	ANO	NE
34	390	dm	n	s	ANO	ANO	ANO	ANO	ANO
35	2	dk	t	s	ANO	NE	NE	ANO	NE
36	270	dl	n	a	ANO	NE	ANO	ANO	NE
37	160	df	s	a	ANO	NE	NE	ANO	NE
38	21	da	s	j	ANO	NE	NE	NE	NE
39	22	dk	n	s	ANO	NE	ANO	ANO	NE
40	387	dl	n	j	NE	NE	ANO	NE	NE
41	13	da	s	j	ANO	NE	ANO	ANO	NE
42	5500	dj		a	ANO	NE	NE	ANO	NE
43	89	da	s	s	ANO	NE	ANO	ANO	NE
44	30	dc	t	s	NE	NE	NE	NE	NE
45	78	da	s	a	ANO	NE	NE	ANO	NE
46	140	da	t	a	ANO	NE	NE	NE	NE
47	767	dj	s	a	ANO	ANO	ANO	ANO	NE
48	50	da	s	a	ANO	ANO	ANO	NE	NE
49	84	da	t	s	ANO	ANO	ANO	ANO	ANO

50	400	dd	t	a	ANO	NE	NE	NE	NE
51	16	da	t	s	ANO	NE	NE	ANO	ANO
52	186	da	t	a	ANO	NE	NE	ANO	NE
53	180	dm	n	s	ANO	ANO	ANO	ANO	NE
54	50	da	n	s	ANO	NE	ANO	NE	NE
55	150	dk	n	a	ANO	NE	NE	ANO	ANO
56	32	da		s	ANO	NE	NE	NE	NE
57	273	da	t	s	ANO	NE	NE	NE	ANO
58	50	dm	t	a	ANO	ANO	ANO	ANO	NE
59	23	dm	s	s	ANO	NE	NE	NE	NE
60	4	dc	t	j	ANO	NE	ANO	ANO	NE
61	13	dn	s	j	ANO	NE	ANO	ANO	NE
62	385	di	s	a	ANO	ANO	ANO	NE	NE
63	1300	dj	s	a	ANO	NE	ANO	ANO	ANO
64	70	dm	t	a	ANO	ANO	ANO	ANO	NE
65	1689	dm	n	a	ANO	NE	ANO	ANO	NE
66	130	dn	t	s	ANO	NE	ANO	ANO	NE
67	55	da	s	a	NE	NE	NE	ANO	NE
68	2	dn	t	j	ANO	NE	NE	ANO	NE
69	66	dj	t	s	NE	NE	NE	ANO	NE
70	330	df	t	a	ANO	NE	ANO	ANO	NE
71	100	dk	t	a	ANO	NE	ANO	ANO	ANO
72	130	dm	t	s	ANO	NE	NE	ANO	NE
73	55	dj	t	s	NE	NE	ANO	ANO	ANO
74		dd	s	s	ANO	ANO	ANO	NE	NE
75	2	dj	t	s	ANO	NE	ANO	ANO	NE
76	4	de	t	s	ANO	NE	NE	ANO	NE
77	171	da	t	a	ANO	NE	NE	ANO	NE
78	500	dj	s	a	ANO	NE	NE	ANO	NE
79	68	df	s	a	ANO	NE	NE	ANO	NE
80	211	di	t	j	ANO	ANO	ANO	ANO	ANO
81	4	j	t	s	ANO	NE	NE	ANO	NE
82	36	dm	n	s	ANO	NE	NE	ANO	NE
83	83	dk	s	s	ANO	NE	NE	ANO	NE
84	298	dl	s	a	ANO	NE	ANO	ANO	NE
85	5	da	s	s	ANO	NE	NE	ANO	NE
86	85	dd	t	s	ANO	NE	NE	NE	NE
87	50	dj	t	s	ANO	NE	NE	NE	NE
88		da	s	s	NE	NE	NE	ANO	NE
89	32	dj	t	s	ANO	NE	NE	NE	NE
90	20	dd	t	s	NE	NE	NE	ANO	NE
91	97	de	n	a	ANO	NE	NE	ANO	ANO
92	130	dk	s	s	ANO	NE	ANO	ANO	NE
93	3	dn	t	s	NE	NE	NE	ANO	NE
94	35	db	t	s	ANO	NE	NE	ANO	NE
95	3	dn	t	j	ANO	NE	ANO	NE	NE
96	1200	dn	t	s	ANO	NE	ANO	NE	NE
97	20	dd	t	s	NE	NE	NE	ANO	NE
98	267	dm	t	a	ANO	NE	NE	ANO	NE
99	680	dk	n	s	ANO	ANO	ANO	ANO	NE
100	15	da	t	j	NE	ANO	ANO	ANO	NE
101	50	db	t	s	NE	NE	NE	NE	NE

102	45	da	t	a	NE	NE	NE	ANO	NE
103	195	dk	t	a	NE	NE	NE	NE	NE
104	1400	dd	t	a	ANO	NE	ANO	ANO	NE
105	1245	dn	t	a	ANO	NE	NE	ANO	NE
106	35	da	t	s	NE	NE	NE	ANO	NE
107	1400	dj	t	a	ANO	NE	ANO	ANO	NE
108	150	dh	t	s	ANO	NE	NE	NE	NE
109	2800	dk	n	a	ANO	NE	ANO	NE	ANO
110	345	dk	n	a	ANO	NE	NE	ANO	NE
111	28	da	s	s	NE	NE	NE	NE	NE
112	5	dl	t	s	ANO	NE	NE	NE	NE
113	486	dk		a	ANO	NE	NE	ANO	NE
114	84	dg	t	a	ANO	NE	NE	NE	NE
115	3	dd	t	j	ANO	NE	NE	ANO	NE
116	32	db	s	s	ANO	NE	ANO	ANO	ANO
117	20	j		s	ANO	ANO	ANO	ANO	NE
118	27	dg	t	s	ANO	NE	NE	NE	NE
119	824	dj	t	a	ANO	NE	NE	NE	NE
120	280	dj	s	a	NE	ANO	ANO	ANO	NE
121		db	s	a	NE	NE	ANO	ANO	NE
122	23	j	t	s	NE	NE	NE	ANO	NE
123	30	dd	t	s	ANO	NE	NE	ANO	NE
124	1620	dm	s	a	ANO	NE	ANO	ANO	NE
125	229	dn	n	a	ANO	NE	NE	ANO	NE
126	87	dn	t	s	ANO	NE	NE	ANO	NE
127	16	dk	s	s	ANO	NE	NE	ANO	NE
128	223	dk	t	s	ANO	NE	ANO	ANO	ANO
129	3	j	n	s	ANO	NE	NE	ANO	NE
130	23	dh	t	s	ANO	NE	ANO	ANO	NE
131	663	dk	s	a	ANO	ANO	ANO	ANO	NE
132	10	dk	t	j	ANO	NE	NE	ANO	ANO
133	20	dm	t	s	ANO	NE	ANO	ANO	NE
134	29	j	t	a	ANO	NE	ANO	ANO	ANO
135	30	dk	t	s	ANO	NE	NE	ANO	NE
136	18	dn	t	s	ANO	NE	ANO	ANO	NE
137	14	dg	s	s	ANO	NE	NE	NE	NE
138	322	j	s	a	ANO	NE	NE	ANO	NE
139	21	j	s	s	ANO	NE	NE	ANO	NE
140	386	di	t	a	ANO	NE	NE	ANO	ANO
141	510	di	s	a	ANO	NE	NE	ANO	NE
142	90	dd	t	s	ANO	NE	NE	ANO	NE
143	80	dk	s	s	ANO	NE	ANO	NE	ANO
144	46	dd	t	s	ANO	NE	NE	NE	NE
145	35	da	t	s	ANO	NE	NE	ANO	NE
146	110	da	t	a	ANO	ANO	ANO	ANO	NE
147	29	dk	t	a	NE	NE	ANO	NE	NE
148	11	dh	t	s	ANO	NE	NE	ANO	NE
149	86	da	n	a	ANO	ANO	ANO	NE	ANO
150	325	de	t	s	ANO	NE	NE	NE	NE
151	50	db	s	s	ANO	NE	ANO	NE	ANO
152	55	dc	t	s	ANO	NE	NE	ANO	NE
153	105	dd	t	s	ANO	NE	NE	ANO	NE

154	113	dl	s	s	ANO	NE	NE	ANO	NE
155		da	t	s	NE	NE	NE	NE	NE
156	40	dk	s	s	ANO	NE	NE	ANO	NE
157	2200	dk dl	t	s	ANO	ANO	ANO	ANO	NE
158	400	de	s	a	ANO	NE	ANO	ANO	NE
159	25	dd	t	s	ANO	NE	NE	ANO	NE
160	7	dk	t	s	ANO	NE	NE	ANO	NE
161	79	dl	t	s	ANO	NE	NE	ANO	ANO
162	279	dl	s	s	ANO	ANO	ANO	ANO	NE
163	135	da	t	a	ANO	NE	NE	ANO	NE
164	108	de	n	j	NE	NE	NE	ANO	NE
165	1800	dl	s	s	ANO	ANO	ANO	ANO	NE
166	150	dg	s	a	NE	NE	NE	ANO	NE
167	8	db	t	s	ANO	NE	NE	ANO	NE
168		dj	t	s	ANO	NE	NE	NE	NE
169	10	dd	t	s	ANO	NE	NE	ANO	ANO
170	350	de	s	k	ANO	NE	ANO	ANO	NE
171	170	db	t	s	NE	NE	NE	NE	NE
172	120	dn	t	s	ANO	NE	NE	ANO	NE
173	350	dj	n	s	ANO	NE	ANO	ANO	NE
174		dk	n	j	ANO	NE	NE	ANO	NE
175	200	dl	t	a	ANO	ANO	ANO	ANO	NE
176	1	dk	t	j	ANO	NE	NE	ANO	NE
177	600	dj	s	a	NE	NE	NE	NE	NE
178	30	da	t	j	ANO	NE	NE	NE	NE
179	185	dl	t	s	ANO	NE	NE	ANO	ANO
180	111	dk	t	a	ANO	NE	NE	ANO	NE
181	135	dk	s	s	ANO	NE	NE	ANO	NE
182	150	dk	t	s	ANO	NE	NE	NE	NE
183	31	j	s	s	ANO	NE	NE	ANO	ANO
184	198	j	n	a	ANO	NE	ANO	ANO	ANO
185	589	j	n	a	ANO	NE	ANO	ANO	NE
186	109	dn	t	a	ANO	NE	NE	ANO	NE
187	20	dd	t	s	ANO	ANO	ANO	ANO	NE
188	2538	dl	n	s	NE	NE	NE	ANO	NE
189	80	dg	t	a	ANO	ANO	ANO	ANO	ANO
190	250	da	t	a	ANO	NE	ANO	ANO	NE
191	13	db	t	s	ANO	ANO	ANO	ANO	NE
192	550	dk	s	s	ANO	NE	NE	ANO	NE
193	77	dh	s	s	NE	NE	NE	NE	NE
194	197	dk	s	s	ANO	NE	NE	ANO	NE
195	46	j	t	s	ANO	NE	NE	NE	NE
196	164	dk	s	a	ANO	NE	NE	NE	NE
197	662	dn	s	a	ANO	ANO	ANO	ANO	ANO
198	17	dd	s	s	ANO	NE	ANO	ANO	ANO
199	49	dl	t	a	ANO	ANO	ANO	ANO	NE
200	239	da	t	a	ANO	NE	NE	NE	NE
201	30	de	s	s	ANO	NE	NE	NE	NE
202	111	dn	t	a	ANO	NE	NE	ANO	NE
203	305	dg	t	a	ANO	ANO	ANO	ANO	ANO
204	45	da	t	j	NE	NE	ANO	NE	NE
205	2429	dl	n	a	ANO	NE	NE	ANO	ANO



206	28	da	t	s	ANO	NE	NE	ANO	NE
207	172	dh	s	s	ANO	NE	NE	NE	NE
208	18	dl	t	s	ANO	ANO	ANO	ANO	NE
209	185	dk	s	a	ANO	NE	NE	ANO	NE
210	130	df	n	a	NE	NE	NE	NE	NE
211	124	da	t	a	ANO	ANO	ANO	ANO	NE
212	100	db	t	j	ANO	NE	NE	ANO	NE
213	1006	dk	s	a	ANO	NE	ANO	ANO	NE
214	106	di	t	s	NE	NE	ANO	ANO	NE
215	56	de	s	s	ANO	NE	NE	ANO	NE
216	43	j	t	s	ANO	NE	NE	NE	NE
217	1807	da	n	s	ANO	NE	NE	NE	NE
218	20	dh	s	s	NE	NE	ANO	ANO	NE
219	8	dd	n	s	NE	NE	NE	ANO	NE
220	50	j	t	a	ANO	ANO	ANO	ANO	NE
221	45	j	t	s	ANO	NE	NE	ANO	NE
222	635	dk	s	a	ANO	ANO	ANO	ANO	NE
223	96	di	t	a	ANO	NE	NE	ANO	NE
224	125	j	t	s	ANO	NE	NE	NE	ANO
225		dl	s	s	ANO	NE	NE	ANO	NE
226	117	j	n	a	ANO	NE	NE	NE	NE
227	1062	dl	s	a	ANO	ANO	ANO	ANO	NE
228	17	de	n	s	ANO	NE	NE	ANO	NE
229	260	dd	n	a	ANO	NE	ANO	ANO	NE
230	1	dk	t	s	NE	NE	ANO	NE	NE
231	254	da	t	a	ANO	NE	NE	NE	NE
232	20	di	t	a	NE	NE	NE	NE	NE
233		dj	t	j	ANO	NE	NE	NE	NE
234	100	dk	t	s	NE	NE	NE	ANO	NE
235		j	t	J	ANO	NE	NE	ANO	NE
236	58	j	t	s	ANO	NE	NE	ANO	NE
237	32	da	t	s	ANO	ANO	ANO	ANO	NE
238	38	da	t	j	NE	NE	NE	NE	NE
239	7	j	n	s	ANO	NE	NE	ANO	NE
240	1	dj	t	j	NE	NE	NE	ANO	NE
241	37	dk	t	s	ANO	NE	ANO	ANO	NE
242	7	j	s	s	ANO	NE	ANO	ANO	NE
243		j		a	ANO	NE	NE	NE	NE
244	251	dk	t	a	ANO	ANO	ANO	ANO	NE
245	10	dl	t	j	ANO	NE	NE	ANO	NE
246	1900	j	n	s	ANO	NE	NE	ANO	ANO
247	30	j	t	s	ANO	NE	NE	ANO	ANO
248	29	dh	n	s	NE	NE	NE	ANO	NE
249	16	dh	t	s	ANO	NE	NE	ANO	NE
250	29	dh	n	s	NE	NE	NE	ANO	NE
251	545	da	n	a	ANO	NE	ANO	ANO	NE
252	7	dj	t	s	ANO	NE	ANO	ANO	ANO
253	6	j		s	ANO	NE	NE	ANO	NE
254	229	dm	n	a	ANO	ANO	ANO	ANO	ANO
255	16	dk	s	s	ANO	NE	NE	NE	NE
256	178	dh	n	s	NE	NE	NE	ANO	NE
257	60	j	t	j	NE	NE	NE	ANO	NE

258	2	j	t	j	NE	NE	NE	NE	NE
259	35	dn	s	s	NE	NE	NE	NE	NE
260	25	dk	s	s	ANO	NE	NE	ANO	ANO
261	24	dj	n	s	ANO	NE	ANO	ANO	ANO
262	1	j	t	s	NE	NE	NE	ANO	NE
263	6	dj	n	s	ANO	NE	ANO	ANO	ANO
264	2	j	t	j	ANO	NE	NE	ANO	NE
265	302	j	n	a	ANO	NE	NE	NE	NE
266	38	da	t	j	NE	NE	NE	NE	NE
267	195	dl	t	s	NE	NE	NE	NE	ANO
268	6		t	s	NE	NE	NE	ANO	ANO
269	350	da	s	a	ANO	NE	NE	NE	NE
270	52	dk	s	s	ANO	NE	ANO	ANO	NE
271		j	t	s	ANO	NE	ANO	ANO	NE
272	561	dj	n	s	ANO	NE	ANO	ANO	NE
273	5	j	t	j	ANO	NE	NE	NE	ANO
274	145	dn	s	s	ANO	NE	ANO	ANO	ANO
275	2		t	s	ANO	NE	NE	ANO	NE
276	120	j	n	s	ANO	NE	NE	ANO	NE
277	208	dj	t	a	ANO	NE	ANO	ANO	ANO
278	50	dg	s	s	NE	NE	NE	ANO	NE
279	13	dk	t	s	ANO	NE	ANO	ANO	NE
280	25	db	t	s	ANO	NE	NE	ANO	NE
281	1473	di	n	a	ANO	NE	NE	ANO	NE
282	800	dj	s	a	NE	NE	NE	NE	NE
283	6	dn	t	s	ANO	NE	NE	ANO	NE
284	45	dl	s	s	ANO	ANO	ANO	ANO	ANO
285	1500	dg		a	ANO	NE	ANO	ANO	NE
286	24	j	t	s	ANO	NE	ANO	NE	ANO
287	64	dn	n	s	ANO	NE	NE	ANO	NE
288	530	dh	t	a	ANO	NE	ANO	ANO	ANO
289	1	de	t	s	NE	NE	NE	ANO	NE
290	302	dj	t	a	ANO	ANO	ANO	NE	ANO
291	16	da	t	s	ANO	ANO	ANO	ANO	NE
292	429	da	s	a	ANO	ANO	ANO	ANO	ANO
293	5	dj	t	s	ANO	NE	NE	ANO	NE
294	44	di	n	s	ANO	NE	NE	ANO	NE
295	10	j	t	s	NE	NE	NE	ANO	NE
296	268	j	t	s	ANO	NE	NE	ANO	NE
297	307	dm	t	s	ANO	NE	NE	ANO	NE
298	30	da	t	a	ANO	NE	ANO	NE	NE
299	5	db	t	j	ANO	NE	NE	ANO	NE
300	117	da		a	ANO	NE	ANO	ANO	ANO
301	100	dj	n	s	NE	NE	NE	ANO	NE
302	915	dh	n	a	ANO	ANO	ANO	ANO	NE
303	53	dk	s	s	ANO	NE	NE	ANO	NE
304	206	j	t	a	ANO	NE	NE	ANO	NE
305	3	dl	s	s	NE	NE	NE	ANO	NE
306	158	dk	s	s	NE	NE	NE	ANO	NE
307	389	dj	s	a	ANO	NE	NE	ANO	ANO
308	26	dg	t	j	ANO	ANO	ANO	NE	NE
309	52	dl	n	s	NE	NE	NE	NE	NE

310		db	t	a s	NE	NE	NE	NE	NE
311	35	dk		a	ANO	NE	NE	NE	NE
312	750	dk	t	a	ANO	NE	NE	ANO	NE
313	280	db	t	j	NE	NE	NE	NE	NE
314	30	j	t	s	ANO	NE	ANO	ANO	NE
315	0	j	t	j	ANO	ANO	ANO	NE	NE
316	23		n	s	NE	NE	NE	ANO	NE
317	9	j	t	s	NE	NE	NE	NE	NE
318	1790	db	n	a	NE	NE	NE	ANO	NE
319		da	s	s	ANO	NE	NE	ANO	NE
320		da	t	j	ANO	NE	ANO	ANO	ANO
321	17	dn	t	s	ANO	NE	ANO	NE	NE
322	25	dn	t	s	ANO	NE	NE	ANO	NE
323	288	db	s	s	ANO	ANO	ANO	ANO	NE
324	27	j	n	s	NE	NE	NE	ANO	NE
325	400	dl	s	s	ANO	NE	ANO	ANO	ANO
326	60	j	s	s	ANO	NE	NE	ANO	NE
327	34	dd		a	ANO	NE	ANO	ANO	NE
328	97	de	t	s	ANO	NE	NE	ANO	NE
329	64	dk	n	s	ANO	NE	NE	ANO	NE
330	106	j	t	j	ANO	NE	NE	ANO	NE
331	35	di	s	k	ANO	NE	NE	ANO	NE
332	25	dk	t	s	ANO	NE	ANO	ANO	NE
333	416	dk	s	a	NE	NE	NE	NE	NE
334	98	dj	t	s	ANO	NE	NE	ANO	NE
335	32	j	t	a	ANO	NE	NE	ANO	ANO
336	350	dj	s	a	NE	NE	NE	NE	NE
337	100	j	t	s	ANO	NE	ANO	ANO	NE
338	18	dj	t	s	ANO	NE	NE	ANO	NE
339	480	dm	t	s	ANO	NE	ANO	ANO	ANO
340	406	da	s	a	ANO	ANO	ANO	ANO	ANO
341	3	j	t	s	ANO	NE	NE	ANO	NE
342	12	dg	n	s	ANO	NE	NE	ANO	NE
343	2	dk	t	s	ANO	NE	ANO	ANO	NE
344	25	dk	t	s	ANO	NE	NE	ANO	NE

*Zdroj: vlastní zpracování*

## Příloha 2 - Zdrojový kód – Form SQL\_Server

```
using System;
using System.Data;
using System.Data.Sql;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace C_dotazniky
{
    public partial class SQL_Server : Form
    {
        public SQL_Server()
        {
            InitializeComponent();
        }

        private void SQL_Server_Load(object sender, EventArgs e)
        {
            // POČÁTEČNÍ VLASTNOSTI OBJEKTŮ
            SQL_UserName.ReadOnly = true;
            SQL_Password.ReadOnly = true;

            comboSQLname.DropDownStyle = ComboBoxStyle.DropDownList;
            comboDatabase.DropDownStyle = ComboBoxStyle.DropDownList;

            comboDatabase.Enabled = false;
            New_database.Enabled = false;
            buttCreateConn.Enabled = false;

            // NAČTENÍ DOSTUPNÝCH SQL SERVERŮ DO COMBOBOXU
            SqlDataSourceEnumerator inst_Srv_load =
                SqlDataSourceEnumerator.Instance;
            System.Data.DataTable seznam_srv = inst_Srv_load.GetDataSources();
            System.Data.DataRow[] rows = seznam_srv.Select();
            foreach (System.Data.DataRow row in rows)
            {
                comboSQLname.Items.Add(row["ServerName"]+"\\ "+row["InstanceName"]);
            }
        }

        private void comboSQLname_SelectedIndexChanged(object sender, EventArgs e)
        {
            // ODEMKNUTÍ COMBOBOXU A BUTTONU DATABÁZE
            if (comboSQLname.SelectedIndex == -1)
            {
                comboDatabase.Enabled = false;
                New_database.Enabled = false;
            }
            else
            {
                comboDatabase.Enabled = true;
                New_database.Enabled = true;
            }
        }
    }
}
```

```

    }

    comboDatabase.Items.Clear();
}

private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    // PŘEPÍNAČ WINDOWS X SQL AUTHENTICATION
    if (radioButton1.Checked == true)
    {
        SQL_UserName.ReadOnly = true;
        SQL_Password.ReadOnly = true;
    }
    else
    {
        SQL_UserName.ReadOnly = false;
        SQL_Password.ReadOnly = false;
    }
}

private void comboDatabase_Click(object sender, EventArgs e)
{
    comboDatabase.Items.Clear();

    // CONNECTION STRING SQL SERVER
    Start.GlobalVars.ServerName = comboSQLname.SelectedItem.ToString();

    if (radioButton1.Checked == true)
    {
        Start.GlobalVars.SrvConnString = "Data source=" +
            Start.GlobalVars.ServerName
            + ";Integrated security=SSPI;Connect Timeout=10";
    }
    else
        Start.GlobalVars.SrvConnString = "Data source=" +
            Start.GlobalVars.ServerName
            + ";Integrated security=TRUE;Connect Timeout=10;User ID=" +
            SQL_UserName
            + ";Password=" + SQL_Password+"";

    // CONNECTION SQL SERVER + NAČTENÍ DOSTUPNÝCH DATABÁZÍ NA SERVERU
    SqlConnection SrvConnection = new
        SqlConnection(Start.GlobalVars.SrvConnString);
    SrvConnection.Open();
    DataTable seznamDB = SrvConnection.GetSchema("Databases");
    SrvConnection.Close();

    foreach (DataRow row in seznamDB.Rows)
    {
        comboDatabase.Items.Add(row["database_name"]);
    }
}

private void New_database_Click(object sender, EventArgs e)
{

```

```

// CONNECTION STRING SQL SERVER
if (radioButton1.Checked == true)
{
    Start.GlobalVars.SrvConnString = "Data source=" +
        Start.GlobalVars.ServerName
        + ";Integrated security=SSPI;Connect Timeout=10";
}
else
    Start.GlobalVars.SrvConnString = "Data source=" +
        Start.GlobalVars.ServerName
        + ";Integrated security=TRUE;Connect Timeout=10;User ID=" +
            SQL_UserName
        + ";Password=" + SQL_Password + ";";

// OTEVŘENÍ FORMU NEW_DB
New_DB New_DB_form = new New_DB();
New_DB_form.ShowDialog();
}

private void button2_Click(object sender, EventArgs e)
{
    Environment.Exit(0);
}

private void comboDatabase_SelectedIndexChanged(object sender, EventArgs e)
{
    // AKTIVACE TLAČÍTKA NA VYTVOŘENÍ SPOJENÍ
    if (comboDatabase.SelectedIndex == -1)
        buttCreateConn.Enabled = false;
    else
        buttCreateConn.Enabled = true;
}

private void buttCreateConn_Click(object sender, EventArgs e)
{
    if (comboDatabase.SelectedIndex == -1)
    {
        MessageBox.Show("Musí být vybrána databáze", "Fail",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        buttCreateConn.Enabled = false;
    }
    else
    {
        // CONNECTION STRING SQL SERVER + DATABÁZE - FINAL
        if (radioButton1.Checked == true)
        {
            Start.GlobalVars.SrvConnString = "Data source=" +
                Start.GlobalVars.ServerName
                + ";Integrated security=SSPI;Connect Timeout=20;Initial
                Catalog="
                + comboDatabase.SelectedItem.ToString() + ";";
        }
        else
        {
            Start.GlobalVars.SrvConnString = "Data source=" +

```

```
        Start.GlobalVars.ServerName
        + ";Integrated security=TRUE;Connect Timeout=20;User ID="
        + SQL_UserName + ";Password=" + SQL_Password + ";Initial
          Catalog="
        + comboDatabase.SelectedItem.ToString() + ";";
    }
    this.Close();
}
}
}
}
```

### Příloha 3 - Zdrojový kód – Form New\_DB

```
using System;
using System.Data;
using System.Data.Sql;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace C_dotazniky
{
    public partial class New_DB : Form
    {
        public New_DB()
        {
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // SQL příkaz CREATE DATABASE
            string qSQL_newDB= "CREATE DATABASE "+ textBox1.Text;

            // PŘIHOJENÍ K SQL SERVERU
            SqlConnection SrvConnection = new SqlConnection(
                Start.GlobalVars.SrvConnString);
            SrvConnection.Open();

            SqlCommand commSQL_newDB = new SqlCommand(qSQL_newDB,SrvConnection);

            // OVĚŘENÍ EXISTENCE DATABÁZE
            try
            {
                commSQL_newDB.ExecuteNonQuery();
                MessageBox.Show("Databáze vytvořena", "Success",
                    MessageBoxButtons.OK, MessageBoxIcon.Information);
                SrvConnection.Close();
                this.Close();
            }
            catch (System.Exception ex)
            {
                MessageBox.Show(ex.ToString(), "Fail",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            finally
            {
                SrvConnection.Close();
            }
        }
    }
}
```



#### Příloha 4 - Zdrojový kód – Form New\_Table

```
using System;
using System.IO;
using System.Windows.Forms;
using System.Data;
using System.Data.Sql;
using System.Data.SqlClient;
using excel = Microsoft.Office.Interop.Excel;
using Microsoft.SqlServer.Dts.Runtime;
using Microsoft.SqlServer.Dts.Runtime Wrapper;
using Microsoft.SqlServer.Dts.Pipeline;
using Microsoft.SqlServer.Dts.Pipeline Wrapper;
using System.Data.OleDb;

namespace C_dotazniky
{
    public partial class New_Table : Form
    {
        public New_Table()
        {
            InitializeComponent();
        }

        // BUTTON KONEC
        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        public string FileFormat = "Delimited";
        public char FileDelimiter = ',';
        public int FileCount=255;
        String ExcelConnStr;
        string excel_ver = "Excel 8.0";
        string excel_provider = "Microsoft.Jet.OLEDB.4.0";

        private void New_Table_Load(object sender, EventArgs e)
        {
            // POČÁTEČNÍ VLATNOSTI OBJEKTŮ
            comboFileFormat.DropDownStyle = ComboBoxStyle.DropDownList;
            comboDelimiter.DropDownStyle = ComboBoxStyle.DropDownList;
            comboExcelVer.DropDownStyle = ComboBoxStyle.DropDownList;
            radioButton1.Checked = true;
            comboFileFormat.Items.Add("S oddělovačem");
            comboFileFormat.Items.Add("Pevná délka");
            comboFileFormat.SelectedIndex = 0;
            comboDelimiter.Items.Add(',');
            comboDelimiter.Items.Add(';');
            comboDelimiter.Items.Add(':');
            comboDelimiter.SelectedIndex = 0;
            textPocetZnaku.Text = "255";
            buttCreateTable.Enabled = false;
            comboExcelVer.Items.Add("97 - 2003");
            comboExcelVer.Items.Add("2007");
            comboExcelVer.SelectedIndex = 0;
        }
    }
}
```

```

}

private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    // ZAPNUTÍ NASTAVENÍ FLATFILE PARAMETRŮ
    if (radioButton1.Checked == true)
    {
        groupBox1.Enabled = true;
        groupBox2.Enabled = false;
    }
    else
    {
        groupBox1.Enabled = false;
        groupBox2.Enabled = true;
    }
}

private void comboFileFormat_SelectedIndexChanged(
    object sender, EventArgs e)
{
    // NASTAVENÍ FILEFORMAT
    if (comboFileFormat.SelectedItem.ToString() == "S oddělovačem")
    {
        comboDelimiter.Enabled = true;
        FileFormat = "Delimited";
        label6.Text="Maximální počet znaků";
        textPocetZnaku.Text = "255";
    }
    else
    {
        comboDelimiter.Enabled = false;
        FileFormat = "FixedWidth";
        label6.Text = "Pevná délka";
        textPocetZnaku.Text = "8";
    }
}

private void textPocetZnaku_TextChanged(object sender, EventArgs e)
{
    // NASTAVENÍ PEVNÉ DÉLKY
    bool result = Int32.TryParse(textPocetZnaku.Text, out FileCount);
    if (result == false)
    {
        MessageBox.Show("Zadejte platné číslo!");
    }
}

private void comboDelimiter_SelectedIndexChanged(
    object sender, EventArgs e)
{
    // NASTAVENÍ FLATFILE DELIMITERU
    if (comboDelimiter.SelectedItem.ToString() == ",")
    {
        FileDelimiter = ',';
    }
}

```

```

        if (comboDelimiter.SelectedItem.ToString() == ";")
        {
            FileDelimiter = ';';
        }
        if (comboDelimiter.SelectedItem.ToString() == ":")
        {
            FileDelimiter = ':';
        }
    }

    private void buttAddFilePath_Click(object sender, EventArgs e)
    {
        // OPENFILE DIALOG PRO ZDROJOVÝ SOUBOR
        if (openFileDialog_SourceFile.ShowDialog() == DialogResult.OK)
        {
            textFilePath.Text = openFileDialog_SourceFile.FileName;
            Start.GlobalVars.PathSourceFile = textFilePath.Text;
        }
    }

    private void textNameTable_TextChanged(object sender, EventArgs e)
    {
        // OŠETŘENÍ PRÁZDNÉHO NÁZVU NOVÉ TABULKY
        if (textNameTable.Text == "" | textFilePath.Text == "")
        {
            buttCreateTable.Enabled = false;
        }
        else
        {
            buttCreateTable.Enabled = true;
        }
    }

    private void textFilePath_TextChanged(object sender, EventArgs e)
    {
        // OŠETŘENÍ NEZADÁNÍ VSTUPNÍHO SOUBORU
        if (textNameTable.Text == "" | textFilePath.Text == "")
        {
            buttCreateTable.Enabled = false;
        }
        else
        {
            buttCreateTable.Enabled = true;
        }
    }

    private void buttCreateTable_Click(object sender, EventArgs e)
    {
        // FLAT FILE
        if (radioButton1.Checked == true)
        {
            string[] srcColumns;
            string[] srcDTypes;

            // STREAM READER PRO FORMAT DELIMITED

```

```

if (FileFormat == "Delimited")
{
    using (StreamReader sr = new StreamReader(
        Start.GlobalVars.PathSourceFile,
        System.Text.Encoding.Default))
    {
        string strcolumns = sr.ReadLine();
        srcColumns = strcolumns.Split(new char[] { FileDelimiter },
            StringSplitOptions.RemoveEmptyEntries);
        string strDtype = sr.ReadLine();
        srcDTypes = strDtype.Split(new char[] { FileDelimiter },
            StringSplitOptions.RemoveEmptyEntries);
    }
}
// STREAM READER PRO FORMAT FIXED WIDTH
else
{
    using (StreamReader sr = new StreamReader(
        Start.GlobalVars.PathSourceFile,
        System.Text.Encoding.Default))
    {
        string strcolumns = sr.ReadLine();
        srcColumns = new string[strcolumns.Length / FileCount];
        string strDtype = sr.ReadLine();
        srcDTypes = new string[strcolumns.Length / FileCount];

        for (int i = 0; i <= srcColumns.Length - 1; i++)
        {
            srcColumns[i] = strcolumns.Substring(i * FileCount,
                FileCount);
            srcColumns[i] = srcColumns[i].Replace(" ", "");
            srcDTypes[i] = strDtype.Substring(i * FileCount,
                FileCount);
            srcDTypes[i] = srcDTypes[i].Replace(" ", "");
        }
    }
}

// VYTVOŘENÍ SQL DOTAZU CREATE TABLE
string qSQL_newTable = "Create table " + textNameTable.Text + "(";

double Num;
int _i = 0;
foreach (string name in srcColumns)
{
    qSQL_newTable += name;
    bool isNum = double.TryParse(srcDTypes[_i], out Num);
    _i += 1;

    if (isNum)
    {
        qSQL_newTable += " float NULL, ";
    }
    else
    {

```

```

        qSQL_newTable += " varchar (" + FileCount.ToString() + "), ";
    }
}
qSQL_newTable = qSQL_newTable.Substring(0, qSQL_newTable.Length -
2);
qSQL_newTable += ")";

MessageBox.Show(qSQL_newTable);

// PŘIPOJENÍ K SQL SERVERU
SqlConnection SrvConnection = new SqlConnection(
    Start.GlobalVars.SrvConnString);
SrvConnection.Open();

SqlCommand commSQL_newTable = new SqlCommand(
    qSQL_newTable, SrvConnection);

// VYTVOŘENÍ A OVĚŘENÍ EXISTENCE TABULKY
try
{
    commSQL_newTable.ExecuteNonQuery();
    MessageBox.Show("Tabulka vytvořena", "Success",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
}
catch (System.Exception ex)
{
    MessageBox.Show(ex.ToString(), "Fail",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}
finally
{
    SrvConnection.Close();

    // NEW PACKAGE
    Start.GlobalVars.Package_txt = new
        Microsoft.SqlServer.Dts.Runtime.Package();

    // SOURCE CONNECTION MANAGER FLATFILE
    Start.GlobalVars.CM_txt = Start.GlobalVars.Package_txt.
        Connections.Add("Flatfile");
    Start.GlobalVars.CM_txt.Name = "ConnMngr_txt";
    Start.GlobalVars.CM_txt.ConnectionString = Start.GlobalVars.
        PathSourceFile;
    Start.GlobalVars.CM_txt.Description = "Flat File Connection";
    Start.GlobalVars.CM_txt.Properties["Format"].
        SetValue(Start.GlobalVars.CM_txt, FileFormat);
    Start.GlobalVars.CM_txt.Properties["HeaderRowDelimiter"].
        SetValue(Start.GlobalVars.CM_txt, "{CR}{LF}");

    Start.GlobalVars.CM_txt.Properties["ColumnNamesInFirstDataRow"].
        SetValue(Start.GlobalVars.CM_txt, true);

    Microsoft.SqlServer.Dts.Runtime.Wrapper.IDTSConnectionMan
agerFlatFile100

```

```

        ffSrcCM = Start.GlobalVars.CM_txt.InnerObject as
Microsoft.SqlServer.Dts.Runtime.Wrapper.IDTSConnectionManagerFlatFile100;

foreach (string name in srcColumns)
{
    Microsoft.SqlServer.Dts.Runtime.Wrapper.
        IDTSConnectionManagerFlatFileColumn100 column =
        ffSrcCM.Columns.Add();
    Microsoft.SqlServer.Dts.Runtime.Wrapper.IDTSName100 colName
= (Microsoft.SqlServer.Dts.Runtime.Wrapper.IDTSName100)column;
    colName.Name = name.Trim();
    column.TextQualified = true;
    column.ColumnType = FileFormat;
    column.DataType =
        Microsoft.SqlServer.Dts.Runtime.Wrapper.DataType.DT_STR;
    column.ColumnWidth = FileCount;
    column.MaximumWidth = FileCount;
    column.DataPrecision = 0;
    column.DataScale = 0;
    if (name == srcColumns[srcColumns.Length - 1])
    {
        column.ColumnDelimiter = "\r\n";
    }
    else
    {
        column.ColumnDelimiter = FileDelimiter.ToString();
    }
}

// DESTINATION CONNECTION MANAGER OLEDB
ConnectionManager CM_OLEDB_dest = Start.GlobalVars.
    Package_txt.Connections.Add("OLEDB");
CM_OLEDB_dest.Name = "ConnMgr_OLEDB_dest";
CM_OLEDB_dest.ConnectionString = Start.GlobalVars.
    SrvConnString + "Provider=SQLNCLI10;";
CM_OLEDB_dest.Description = "OLEDB Connection";

// NEW DATA FLOW TASK
Executable new_dataflow = Start.GlobalVars.Package_txt.
    Executables.Add("STOCK:PipelineTask");
Microsoft.SqlServer.Dts.Runtime.TaskHost thMainPipe
    = new_dataflow as Microsoft.SqlServer.Dts.Runtime.TaskHost;
MainPipe dataFlowTask_txt = thMainPipe.InnerObject as MainPipe;
thMainPipe.Name = "dataFlowtask_txt";

// NEW FLAT FILE SOURCE
IDTSComponentMetaData100 FlatFile_txt = dataFlowTask_txt.
    ComponentMetaDataCollection.New();
FlatFile_txt.ComponentClassID = "DTSAdapter.FlatFileSource";
FlatFile_txt.Name = "Source_txt";

CManagedComponentWrapper inst_txt = FlatFile_txt.Instantiate();

```

```

inst_txt.ProvideComponentProperties();

FlatFile_txt.RuntimeConnectionCollection[0].ConnectionManagerID
    = Start.GlobalVars.CM_txt.ID;
FlatFile_txt.RuntimeConnectionCollection[0].ConnectionManager =
Microsoft.SqlServer.Dts.Runtime.DtsConvert.GetExtendedInterface(
    Start.GlobalVars.CM_txt);

inst_txt.AcquireConnections(null);
inst_txt.ReinitializeMetaData();
inst_txt.ReleaseConnections();

// NEW OLEDB
IDTSComponentMetaData100 OLEDB_Dest = dataFlowTask_txt.
    ComponentMetaDataCollection.New();
OLEDB_Dest.ComponentClassID = "DTSAdapter.OLEDBDestination";
OLEDB_Dest.Name = "OLEDB dest";

CManagedComponentWrapper inst_OLEDB = OLEDB_Dest.Instantiate();
inst_OLEDB.ProvideComponentProperties();

OLEDB_Dest.RuntimeConnectionCollection[0].ConnectionManagerID =
    CM_OLEDB_dest.ID;
OLEDB_Dest.RuntimeConnectionCollection[0].ConnectionManager =
    Microsoft.SqlServer.Dts.Runtime.DtsConvert.
    GetExtendedInterface(CM_OLEDB_dest);

inst_OLEDB.SetComponentProperty("OpenRowset", "[dbo].[ " +
    textNameTable.Text + " ]");

// PATH FLAT FILE - OLEDB
IDTSPath100 path = dataFlowTask_txt.PathCollection.New();

Path.AttachPathAndPropagateNotifications(FlatFile_txt.OutputColl
ection[0],
    OLEDB_Dest.InputCollection[0]);

IDTSInput100 destinationInput = OLEDB_Dest.InputCollection[0];
IDTSVirtualInput100 destinationVirtualInput =
    destinationInput.GetVirtualInput();
IDTSVirtualInputColumnCollection100
    destinationVirtualInputColumns
    = destinationVirtualInput.VirtualInputColumnCollection;

inst_OLEDB.AcquireConnections(null);
inst_OLEDB.ReinitializeMetaData();
inst_OLEDB.ReleaseConnections();

// MAPOVÁNÍ
foreach (IDTSVirtualInputColumn100 virtualInputColumn in
    destinationVirtualInputColumns)
{
    // VÝBĚR ZDROJOVÉHO SLOUPCE
    IDTSInputColumn100 inputColumn = inst_OLEDB.

```

```

        SetUsageType(destinationInput.ID,
            destinationVirtualInput, virtualInputColumn.LineageID,
            DTSUsageType.UT_READONLY);
    // NALEZENÍ VÝSTUPNÍHO SLOUPCE
    IDTSExternalMetadataColumn100 externalColumn =
        destinationInput.ExternalMetadataColumnCollection[
            inputColumn.Name];
    // MAPOVÁNÍ VSTUPU A VÝSTUPU
    inst_OLEDB.MapInputColumn(destinationInput.ID,
        inputColumn.ID, externalColumn.ID);
    }

    // SAVE PACKAGE
    Microsoft.SqlServer.Dts.Runtime.Application ulozeni
        = new Microsoft.SqlServer.Dts.Runtime.Application();

    //ulozeni.SaveToXml(@"c:\Users\Majkl\Desktop\Dotazniky\
    BI_dotazniky\BI_dotazniky"
    // + @"\bin\txt_pack.dtsx", Start.GlobalVars.Package_txt,
    null);

    // EXECUTE PACKAGE
    try
    {
        Start.GlobalVars.Package_txt.Execute();
        MessageBox.Show("Data nahrána", "Success",
            MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        this.Close();
    }
    catch (System.Exception ex)
    {
        MessageBox.Show(ex.ToString(), "Fail",
            MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
}
}
// EXCEL FILE
else
{
    // EXCEL CONNECTION STRING OLEDB
    ExcelConnStr = "Provider=" + excel_provider +
        "Data Source=" + textFilePath.Text
        + @";Extended Properties="" + excel_ver +
        @";HDR=YES;IMEX=1""";

    OleDbConnection Conn_excel = new OleDbConnection(ExcelConnStr);
    Conn_excel.Open();

    // EXCEL CREATE TABLE
    string sheetName = comboExcelList.SelectedItem.ToString() + "$";
    DataTable dt_excel =
        Conn_excel.GetOleDbSchemaTable(OleDbSchemaGuid.Columns,

```



```

        new object[] { null, null, sheetName, null });

DataView dv_col = new DataView(dt_excel);
dv_col.Sort = "ORDINAL_POSITION";

excel.Application excel_APP_D = new excel.Application();
excel.Workbook excel_WB_D =
    excel_APP_D.Workbooks.Open(textFilePath.Text);
excel.Worksheet excel_WS_D = excel_WB_D.Worksheets.get_Item(
    comboExcelList.SelectedIndex + 1);

string qSQL_newTable = "Create table " + textNameTable.Text + "(";

double Num;

DataRowView drv_col;
for (int j = 0; j < dv_col.Count; j++)
{
    drv_col = dv_col[j];
    qSQL_newTable += drv_col.Row.ItemArray[3].ToString();

    if (excel_WS_D.Cells[2, j + 1].Value == null)
    {
        excel_WS_D.Cells[2, j + 1].Value = "aaa";
    }

    bool IsNum =
double.TryParse(excel_WS_D.Cells[2, j+1].Value.ToString(), out
    Num);

    if (IsNum)
    {
        qSQL_newTable += " float NULL, ";
    }
    else
    {
        qSQL_newTable += " nvarchar (" + FileCount.ToString() + "),
";
    }
}

qSQL_newTable = qSQL_newTable.Substring(0, qSQL_newTable.Length -
    2);
qSQL_newTable += ")";

excel_WB_D.Saved = true;
excel_APP_D.Quit() ;

// PŘIPOJENÍ K SQL SERVERU
SqlConnection SrvConnection = new
    SqlConnection(Start.GlobalVars.SrvConnString);
SrvConnection.Open();

```

```

SqlCommand commSQL_newTable = new SqlCommand(qSQL_newTable,
    SrvConnection);

// VYTVOŘENÍ A OVĚŘENÍ EXISTENCE TABULKY
try
{
    commSQL_newTable.ExecuteNonQuery();
    MessageBox.Show("Tabulka vytvořena", "Success",
        MessageBoxButtons.OK,
        MessageBoxIcon.Information);
}
catch (System.Exception ex)
{
    MessageBox.Show(ex.ToString(), "Fail", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
finally
{
    SrvConnection.Close();
}

// NEW PACKAGE
Microsoft.SqlServer.Dts.Runtime.Package Package_Excel =
    new Microsoft.SqlServer.Dts.Runtime.Package();

// SOURCE CONNECTION MANAGER EXCEL
ConnectionManager CM_excel =
    Package_Excel.Connections.Add("EXCEL");
CM_excel.Name = "ConnMgr_excel";
CM_excel.ConnectionString = ExcelConnStr;

// DESTINATION CONNECTION MANAGER OLEDB
ConnectionManager CM_OLEDB_dest =
    Package_Excel.Connections.Add("OLEDB");
CM_OLEDB_dest.Name = "ConnMgr_OLEDB_dest";
CM_OLEDB_dest.ConnectionString = Start.GlobalVars.SrvConnString +
    "Provider=SQLNCLI10;";
CM_OLEDB_dest.Description = "OLEDB connection";

// NEW DATA FLOW TASK
Executable new_dataflow = Package_Excel.Executables.Add(
    "STOCK:PipelineTask");
Microsoft.SqlServer.Dts.Runtime.TaskHost thMainPipe
    = new_dataflow as Microsoft.SqlServer.Dts.Runtime.TaskHost;
MainPipe dataFlowTask_excel = thMainPipe.InnerObject as MainPipe;
thMainPipe.Name = "dataFlowtask_EXCEL";

// NEW EXCEL SOURCE
IDTSComponentMetaData100 Excel_source = dataFlowTask_excel.
    ComponentMetaDataCollection.New();
Excel_source.ComponentClassID = "DTSAdapter.ExcelSource";
Excel_source.Name = "Source_Excel";

CManagedComponentWrapper inst_excel = Excel_source.Instantiate();
inst_excel.ProvideComponentProperties();

```

```

Excel_source.RuntimeConnectionCollection[0].ConnectionManagerID =
    CM_excel.ID;
Excel_source.RuntimeConnectionCollection[0].ConnectionManager =

Microsoft.SqlServer.Dts.Runtime.DtsConvert.GetExtendedInterface(CM_excel);
inst_excel.SetComponentProperty("AccessMode", 0);
inst_excel.SetComponentProperty("OpenRowset", comboExcelList.
    Text.ToString() + "$");

inst_excel.AcquireConnections(null);
inst_excel.ReinitializeMetaData();
inst_excel.ReleaseConnections();

// NEW OLEDB
IDTSComponentMetaData100 OLEDB_Dest = dataFlowTask_excel.
    ComponentMetaDataCollection.New();
OLEDB_Dest.ComponentClassID = "DTSAdapter.OLEDBDestination";
OLEDB_Dest.Name = "OLEDB dest";

CManagedComponentWrapper inst_OLEDB = OLEDB_Dest.Instantiate();
inst_OLEDB.ProvideComponentProperties();

OLEDB_Dest.RuntimeConnectionCollection[0].ConnectionManagerID =
    CM_OLEDB_dest.ID;
OLEDB_Dest.RuntimeConnectionCollection[0].ConnectionManager =

Microsoft.SqlServer.Dts.Runtime.DtsConvert.GetExtendedInterface(
    CM_OLEDB_dest);

inst_OLEDB.SetComponentProperty("OpenRowset", "[dbo].[" +
    textNameTable.Text + "]);

// PATH FLAT FILE - OLEDB
IDTSPath100 path = dataFlowTask_excel.PathCollection.New();
path.AttachPathAndPropagateNotifications(Excel_source.OutputCollection[0],
    OLEDB_Dest.InputCollection[0]);

IDTSInput100 destinationInput = OLEDB_Dest.InputCollection[0];
IDTSVirtualInput100 destinationVirtualInput = destinationInput.
    GetVirtualInput();
IDTSVirtualInputColumnCollection100 destinationVirtualInputColumns
    = destinationVirtualInput.VirtualInputColumnCollection;

inst_OLEDB.AcquireConnections(null);
inst_OLEDB.ReinitializeMetaData();
inst_OLEDB.ReleaseConnections();

// MAPOVÁNÍ
foreach (IDTSVirtualInputColumn100 virtualInputColumn in
    destinationVirtualInputColumns)
{
    // VÝBĚR ZDROJOVÉHO SLOUPCE
    IDTSInputColumn100 inputColumn = inst_OLEDB.SetUsageType(
        destinationInput.ID, destinationVirtualInput,

```

```

        virtualInputColumn.LineageID, DTSUsageType.UT_READONLY);
    // NALEZENÍ VÝSTUPNÍHO SLOUPCE
    IDTSExternalMetadataColumn100 externalColumn =
        destinationInput.ExternalMetadataColumnCollection
            [inputColumn.Name];
    // MAPOVÁNÍ VSTUPU A VÝSTUPU
    inst_OLEDB.MapInputColumn(destinationInput.ID, inputColumn.ID,
        externalColumn.ID);
}

// SAVE PACKAGE
Microsoft.SqlServer.Dts.Runtime.Application ulozeni_excel
    = new Microsoft.SqlServer.Dts.Runtime.Application();

//ulozeni_excel.SaveToXml(@"c:\\Users\Majkl\Desktop\Dotazniky\BI_dotazniky"
//    + @"\BI_dotazniky\bin\excel_pack.dtsx", Package_Excel, null);

// EXECUTE PACKAGE
try
{
    Package_Excel.Execute();
    MessageBox.Show("Data nahrána", "Success",
        MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    this.Close();
}
catch (System.Exception ex)
{
    MessageBox.Show(ex.ToString(), "Fail", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
}

private void comboExcelVer_SelectedIndexChanged(object sender, EventArgs e)
{
    if (comboExcelVer.SelectedItem.ToString() == "97 - 2003")
    {
        excel_ver = "Excel 8.0";
        excel_provider = "Microsoft.Jet.OLEDB.4.0;";
    }
    else
    {
        excel_ver = "Excel 12.0 Xml";
        excel_provider = "Microsoft.ACE.OLEDB.12.0;";
    }
}

private void comboExcelList_Click(object sender, EventArgs e)
{
    if (textFilePath.Text == "")
    {
        MessageBox.Show("Musíte zadat cestu ke zdrojovému souboru Excel.");
    }
    else

```



## Příloha 5 - Zdrojový kód – Hlavní okno – Form Start

```
using System;
using System.IO;
using System.Windows.Forms;
using System.Collections.Generic;
using System.Data;
using System.Data.Sql;
using System.Data.SqlClient;
using System.Data.OleDb;
using excel = Microsoft.Office.Interop.Excel;
using Microsoft.AnalysisServices;
using Microsoft.AnalysisServices.AdomdClient;
using Microsoft.SqlServer.Dts.Runtime;
using Microsoft.SqlServer.Dts.Runtime Wrapper;
using Microsoft.SqlServer.Dts.Pipeline;
using Microsoft.SqlServer.Dts.Pipeline Wrapper;

namespace C_dotazniky
{
    public partial class Start : Form
    {
        public Start()
        {
            InitializeComponent();
        }

        // GLOBÁLNÍ PROMĚNNÉ
        public static class GlobalVars
        {
            // GLOBAL CONNECTION STRING
            private static string _SrvConnString;
            public static string SrvConnString
            {
                get { return _SrvConnString; }
                set { _SrvConnString = value; }
            }

            // GLOBAL Server
            private static string _ServerName;
            public static string ServerName
            {
                get { return _ServerName; }
                set { _ServerName = value; }
            }

            // GLOBAL CESTA KE ZDROJOVÉMU SOUBORU
            private static string _PathSourceFile;
            public static string PathSourceFile
            {
                get { return _PathSourceFile; }
                set { _PathSourceFile = value; }
            }

            // CONNECTION MANAGER TXT
            private static ConnectionManager _CM_txt;
        }
    }
}
```

```

public static ConnectionManager CM_txt
{
    get { return _CM_txt; }
    set { _CM_txt = value; }
}

// PACKAGE TXT
private static Microsoft.SqlServer.Dts.Runtime.Package _Package_txt;
public static Microsoft.SqlServer.Dts.Runtime.Package Package_txt
{
    get { return _Package_txt; }
    set { _Package_txt = value; }
}
}

private void Start_Load(object sender, EventArgs e)
{
    // OTEVŘENÍ FORMU SQL_SERVER
    SQL_Server SQL_Server_form = new SQL_Server();
    SQL_Server_form.ShowDialog();

    // NAČTENÍ TABULEK DO LISTBOXU
    listTables.Items.Clear();
    SqlConnection SrvConnection = new
        SqlConnection(Start.GlobalVars.SrvConnString);
    SrvConnection.Open();
    System.Data.DataTable seznamDB = SrvConnection.GetSchema("Tables");
    SrvConnection.Close();

    foreach (DataRow row in seznamDB.Rows)
    {
        listTables.Items.Add(row["table_name"]);
    }
}

private void buttNewTable_Click(object sender, EventArgs e)
{
    New_Table New_Table_form = new New_Table();
    New_Table_form.ShowDialog();

    // NAČTENÍ TABULEK DO LISTBOXU
    listTables.Items.Clear();
    SqlConnection SrvConnection = new
        SqlConnection(Start.GlobalVars.SrvConnString);
    SrvConnection.Open();
    System.Data.DataTable seznamDB = SrvConnection.GetSchema("Tables");
    SrvConnection.Close();

    foreach (DataRow row in seznamDB.Rows)
    {
        listTables.Items.Add(row["table_name"]);
    }
}

private void buttDelTable_Click(object sender, EventArgs e)

```

```

{
    // SMAZÁNÍ TABULKY
    if (listTables.SelectedIndex == -1)
    {
        MessageBox.Show("Musí být vybrána tabulka");
    }
    else
    {
        string qSQL_DelTable = "Drop Table " +
            listTables.SelectedItem.ToString();

        // PŘIPOJENÍ K SQL SERVERU
        if (MessageBox.Show("Opravdu chcete smazat tabulku " +
            listTables.SelectedItem.ToString()
            + "?", "Smazání tabulky", MessageBoxButtons.YesNo,
            MessageBoxIcon.Warning) == DialogResult.Yes)
        {
            SqlConnection SrvConnection = new SqlConnection(
                Start.GlobalVars.SrvConnString);
            SrvConnection.Open();

            SqlCommand commSQL_DelTable = new SqlCommand(
                qSQL_DelTable, SrvConnection);

            commSQL_DelTable.ExecuteNonQuery();

            // NAČTENÍ TABULEK DO LISTBOXU
            listTables.Items.Clear();
            System.Data.DataTable seznamDB =
                SrvConnection.GetSchema("Tables");
            SrvConnection.Close();

            foreach (DataRow row in seznamDB.Rows)
            {
                listTables.Items.Add(row["table_name"]);
            }
        }
    }
}

private void Predikce_Click(object sender, EventArgs e)
{
    // CONNECTION ANALYSIS SERVER
    Server Anal_Server = new Server();
    string strAnal_Conn = "Data Source=" + Start.GlobalVars.ServerName +
        ";Provider=msolap;";
    Anal_Server.Connect(strAnal_Conn);

    // NOVÁ ANALYTICKÁ DATABÁZE
    Database Anal_Database = new Database();
    Anal_Database = null;

    Anal_Database =
        Anal_Server.Databases.FindByName("C#_PREDIKCE_DOTAZNIKY");
    if (Anal_Database != null)

```



```

    {
        Anal_Database.Drop();
    }

Anal_Database = Anal_Server.Databases.Add(
    Anal_Server.Databases.GetNewName("C#_PREDIKCE_DOTAZNIKY"));
Anal_Database.Update();

// NOVÝ DATASOURCE
RelationalDataSource Anal_DataSource = new RelationalDataSource();
Anal_DataSource = Anal_Database.DataSources.Add("DATASOURCE",
    "DATASOURCE");
Anal_DataSource.ConnectionString = "Provider=SQLOLEDB.1;" +
    Start.GlobalVars.SrvConnString;
Anal_DataSource.Update();

// NOVÝ DATASOURCE VIEW
DataSourceView Anal_View = new DataSourceView();
Anal_View = Anal_Database.DataSourceViews.Add(
    Anal_Database.DataSourceViews.GetNewName("DATASOURCE_VIEW"));
Anal_View.DataSourceID = "DATASOURCE";
Anal_View.Schema = new DataSet();

string commSQL_SelectALL = "SELECT * FROM [dbo].[" +
    train_table.Text + "];";
OleDbDataAdapter SQL_adapter = new OleDbDataAdapter(commSQL_SelectALL,
    "Provider=SQLOLEDB.1;" + Start.GlobalVars.SrvConnString);

System.Data.DataTable[] dataTables =
    SQL_adapter.FillSchema(Anal_View.Schema,
        SchemaType.Mapped, train_table.Text);
System.Data.DataTable dataTable = dataTables[0];

dataTable.ExtendedProperties.Add("TableType", "Table");
dataTable.ExtendedProperties.Add("DbSchemaName", "dbo");
dataTable.ExtendedProperties.Add("DbTableName", train_table.Text);
dataTable.ExtendedProperties.Add("FriendlyName", train_table.Text);

Anal_View.Update();

dataTable = null;
dataTables = null;
SQL_adapter = null;

// MINING STRUCTURE
if (Anal_Database.MiningStructures.ContainsName("Dotazniky_MS"))
{
    Anal_Database.MiningStructures["Dotazniky_MS"].Drop();
}

Microsoft.AnalysisServices.MiningStructure Anal_MiningStructure =
    Anal_Database.MiningStructures.Add("Dotazniky_MS", "Dotazniky_MS");

Anal_MiningStructure.Source = new DataSourceViewBinding(
    Anal_Database.DataSourceViews[0].ID);

```

```
Anal_MiningStructure.HoldoutMaxCases = 200;  
Anal_MiningStructure.HoldoutMaxPercent = 30;
```

```
ScalarMiningStructureColumn MS_ID =  
    Anal_MiningStructure.Columns.Add("ID", "MS_ID");  
MS_ID.IsKey = true;  
MS_ID.Type = MiningStructureColumnTypes.Double;  
MS_ID.Content = MiningStructureColumnContents.Key;  
MS_ID.KeyColumns.Add(train_table.Text, "ID", OleDbType.Double);
```

```
ScalarMiningStructureColumn MS_Inovacni_5 =  
    Anal_MiningStructure.Columns.Add(  
        "Inovacni_5", "MS_Inovacni_5");  
MS_Inovacni_5.Type = MiningStructureColumnTypes.Double;  
MS_Inovacni_5.Content = MiningStructureColumnContents.Discrete;  
MS_Inovacni_5.KeyColumns.Add(train_table.Text, "Inovacni_5",  
    OleDbType.Double);
```

```
ScalarMiningStructureColumn MS_Univerzity_5 =  
    Anal_MiningStructure.Columns.Add(  
        "Univerzity_5", "MS_Univerzity_5");  
MS_Univerzity_5.Type = MiningStructureColumnTypes.Double;  
MS_Univerzity_5.Content = MiningStructureColumnContents.Discrete;  
MS_Univerzity_5.KeyColumns.Add(train_table.Text, "Univerzity_5",  
    OleDbType.Double);
```

```
ScalarMiningStructureColumn MS_AkadVed_5 =  
    Anal_MiningStructure.Columns.Add(  
        "AkadVed_5", "MS_AkadVed_5");  
MS_AkadVed_5.Type = MiningStructureColumnTypes.Double;  
MS_AkadVed_5.Content = MiningStructureColumnContents.Discrete;  
MS_AkadVed_5.KeyColumns.Add(train_table.Text, "AkadVed_5",  
    OleDbType.Double);
```

```
ScalarMiningStructureColumn MS_Vlastnici_3 =  
    Anal_MiningStructure.Columns.Add(  
        "Vlastnici_3", "MS_Vlastnici_3");  
MS_Vlastnici_3.Type = MiningStructureColumnTypes.Double;  
MS_Vlastnici_3.Content = MiningStructureColumnContents.Discrete;  
MS_Vlastnici_3.KeyColumns.Add(train_table.Text, "Vlastnici_3",  
    OleDbType.Double);
```

```
ScalarMiningStructureColumn MS_FinInstitute_3 =  
    Anal_MiningStructure.Columns.Add(  
        "FinInstitute_3", "MS_FinInstitute_3");  
MS_FinInstitute_3.Type = MiningStructureColumnTypes.Double;  
MS_FinInstitute_3.Content = MiningStructureColumnContents.Discrete;  
MS_FinInstitute_3.KeyColumns.Add(train_table.Text, "FinInstitute_3",  
    OleDbType.Double);
```

```
ScalarMiningStructureColumn MS_UradMS_3 =  
    Anal_MiningStructure.Columns.Add(  
        "UradMS_3", "MS_UradMS_3");  
MS_UradMS_3.Type = MiningStructureColumnTypes.Double;  
MS_UradMS_3.Content = MiningStructureColumnContents.Discrete;
```

```

MS_UradMS_3.KeyColumns.Add(train_table.Text, "UradMS_3",
    OleDbType.Double);

ScalarMiningStructureColumn MS_Univerzity_16 =
    Anal_MiningStructure.Columns.Add(
        "Univerzity_16", "MS_Univerzity_16");
MS_Univerzity_16.Type = MiningStructureColumnTypes.Double;
MS_Univerzity_16.Content = MiningStructureColumnContents.Discrete;
MS_Univerzity_16.KeyColumns.Add(train_table.Text, "Univerzity_16",
    OleDbType.Double);

ScalarMiningStructureColumn MS_Konkurence_16 =
    Anal_MiningStructure.Columns.Add(
        "Konkurence_16", "MS_Konkurence_16");
MS_Konkurence_16.Type = MiningStructureColumnTypes.Double;
MS_Konkurence_16.Content = MiningStructureColumnContents.Discrete;
MS_Konkurence_16.KeyColumns.Add(train_table.Text, "Konkurence_16",
    OleDbType.Double);

ScalarMiningStructureColumn MS_Zakaznici_2 =
    Anal_MiningStructure.Columns.Add(
        "Zakaznici_2", "MS_Zakaznici_2");
MS_Zakaznici_2.Type = MiningStructureColumnTypes.Double;
MS_Zakaznici_2.Content = MiningStructureColumnContents.Discrete;
MS_Zakaznici_2.KeyColumns.Add(train_table.Text, "Zakaznici_2",
    OleDbType.Double);

ScalarMiningStructureColumn MS_Dodavatele_2 =
    Anal_MiningStructure.Columns.Add(
        "Dodavatele_2", "MS_Dodavatele_2");
MS_Dodavatele_2.Type = MiningStructureColumnTypes.Double;
MS_Dodavatele_2.Content = MiningStructureColumnContents.Discrete;
MS_Dodavatele_2.KeyColumns.Add(train_table.Text, "Dodavatele_2",
    OleDbType.Double);

ScalarMiningStructureColumn MS_Poradna_2 =
    Anal_MiningStructure.Columns.Add(
        "Poradna_2", "MS_Poradna_2");
MS_Poradna_2.Type = MiningStructureColumnTypes.Double;
MS_Poradna_2.Content = MiningStructureColumnContents.Discrete;
MS_Poradna_2.KeyColumns.Add(train_table.Text, "Poradna_2",
    OleDbType.Double);

ScalarMiningStructureColumn MS_T_INOVACE =
    Anal_MiningStructure.Columns.Add(
        "T_INOVACE", "MS_T_INOVACE");
MS_T_INOVACE.Type = MiningStructureColumnTypes.Text;
MS_T_INOVACE.Content = MiningStructureColumnContents.Discrete;
MS_T_INOVACE.KeyColumns.Add(train_table.Text, "T_INOVACE");

ScalarMiningStructureColumn MS_T_FINANCE =
    Anal_MiningStructure.Columns.Add(
        "T_FINANCE", "MS_T_FINANCE");
MS_T_FINANCE.Type = MiningStructureColumnTypes.Text;
MS_T_FINANCE.Content = MiningStructureColumnContents.Discrete;

```

```

MS_T_FINANCE.KeyColumns.Add(train_table.Text, "T_FINANCE");

ScalarMiningStructureColumn MS_T_HR = Anal_MiningStructure.Columns.Add(
    "T_HR", "MS_T_HR");
MS_T_HR.Type = MiningStructureColumnTypes.Text;
MS_T_HR.Content = MiningStructureColumnContents.Discrete;
MS_T_HR.KeyColumns.Add(train_table.Text, "T_HR");

ScalarMiningStructureColumn MS_T_KONTAKTY =
    Anal_MiningStructure.Columns.Add(
        "T_KONTAKTY", "MS_T_KONTAKTY");
MS_T_KONTAKTY.Type = MiningStructureColumnTypes.Text;
MS_T_KONTAKTY.Content = MiningStructureColumnContents.Discrete;
MS_T_KONTAKTY.KeyColumns.Add(train_table.Text, "T_KONTAKTY");

ScalarMiningStructureColumn MS_T_PORADNA =
    Anal_MiningStructure.Columns.Add(
        "T_PORADNA", "MS_T_PORADNA");
MS_T_PORADNA.Type = MiningStructureColumnTypes.Text;
MS_T_PORADNA.Content = MiningStructureColumnContents.Discrete;
MS_T_PORADNA.KeyColumns.Add(train_table.Text, "T_PORADNA");

Anal_MiningStructure.Update();

// MINING MODEL
if (Anal_MiningStructure.MiningModels.ContainsName("Dotazniky_MM"))
{
    Anal_MiningStructure.MiningModels["Dotazniky_MM"].Drop();
}

Microsoft.AnalysisServices.MiningModel Anal_MiningModel =
    Anal_MiningStructure.CreateMiningModel(true, "Dotazniky_MM");
Anal_MiningModel.Algorithm =
    MiningModelAlgorithms.MicrosoftDecisionTrees;

Microsoft.AnalysisServices.MiningModelColumn MM_T_INOVACE =
    Anal_MiningModel.Columns.Add("MM_T_INOVACE", "MM_T_INOVACE");
MM_T_INOVACE.SourceColumnID = "MS_T_INOVACE";
MM_T_INOVACE.Usage = MiningModelColumnUsages.PredictOnly;

Microsoft.AnalysisServices.MiningModelColumn MM_T_FINANCE =
    Anal_MiningModel.Columns.Add("MM_T_FINANCE", "MM_T_FINANCE");
MM_T_FINANCE.SourceColumnID = "MS_T_FINANCE";
MM_T_FINANCE.Usage = MiningModelColumnUsages.PredictOnly;

Microsoft.AnalysisServices.MiningModelColumn MM_T_HR =
    Anal_MiningModel.Columns.Add("MM_T_HR", "MM_T_HR");
MM_T_HR.SourceColumnID = "MS_T_HR";
MM_T_HR.Usage = MiningModelColumnUsages.PredictOnly;

Microsoft.AnalysisServices.MiningModelColumn MM_T_KONTAKTY =
    Anal_MiningModel.Columns.Add("MM_T_KONTAKTY", "MM_T_KONTAKTY");
MM_T_KONTAKTY.SourceColumnID = "MS_T_KONTAKTY";
MM_T_KONTAKTY.Usage = MiningModelColumnUsages.PredictOnly;

```

```

Microsoft.AnalysisServices.MiningModelColumn MM_T_PORADNA =
    Anal_MiningModel.Columns.Add("MM_T_PORADNA", "MM_T_PORADNA");
MM_T_PORADNA.SourceColumnID = "MS_T_PORADNA";
MM_T_PORADNA.Usage = MiningModelColumnUsages.PredictOnly;

Microsoft.AnalysisServices.MiningModelColumn del_MS_1 =
    Anal_MiningModel.Columns.FindByName("T_INOVACE");
Anal_MiningModel.Columns.Remove(del_MS_1);

Microsoft.AnalysisServices.MiningModelColumn del_MS_2 =
    Anal_MiningModel.Columns.FindByName("T_FINANCE");
Anal_MiningModel.Columns.Remove(del_MS_2);

Microsoft.AnalysisServices.MiningModelColumn del_MS_3 =
    Anal_MiningModel.Columns.FindByName("T_HR");
Anal_MiningModel.Columns.Remove(del_MS_3);

Microsoft.AnalysisServices.MiningModelColumn del_MS_4 =
    Anal_MiningModel.Columns.FindByName("T_KONTAKTY");
Anal_MiningModel.Columns.Remove(del_MS_4);

Microsoft.AnalysisServices.MiningModelColumn del_MS_5 =
    Anal_MiningModel.Columns.FindByName("T_PORADNA");
Anal_MiningModel.Columns.Remove(del_MS_5);

Anal_MiningModel.Update();
Anal_MiningModel.Process();

// PREDIKCE
string commDMX_string = "SELECT
    t.[ID],t.[Pocet_zamestnancu],t.[Obor],t.[Management]," +
    "t.[Pravni_forma],(Predict([Dotazniky_MM].[MM_T_FINANCE])) as
    [FINANCE]," +
    "(Predict([Dotazniky_MM].[MM_T_HR])) as [HR]," +
    "(Predict([Dotazniky_MM].[MM_T_INOVACE])) as [INOVACE]," +
    "(Predict([Dotazniky_MM].[MM_T_KONTAKTY])) as [KONTAKTY]," +
    "(Predict([Dotazniky_MM].[MM_T_PORADNA])) as [PORADNA] " +
    "From [Dotazniky_MM] PREDICTION JOIN OPENQUERY([DATASOURCE]," +
    "'SELECT
    ID],[Pocet_zamestnancu],[Obor],[Management],[Pravni_forma]," +
    "[Inovacni_5],[Univerzity_5],[AkadVed_5],[Vlastnici_3],[FinInstitute_3]," +
    "[UradMS_3],[Univerzity_16],[Konkurence_16],[Zakaznici_2],[Dodavatele_2]," +
    "[Poradna_2],[T_INOVACE],[T_FINANCE],[T_HR],[T_KONTAKTY],[T_PORADNA] " +
    "FROM [dbo].[ " + pred_table.Text + " ]') AS t ON " +
    "[Dotazniky_MM].[Inovacni_5] = t.[Inovacni_5] AND " +
    "[Dotazniky_MM].[Univerzity_5] = t.[Univerzity_5] AND " +
    "[Dotazniky_MM].[AkadVed_5] = t.[AkadVed_5] AND " +
    "[Dotazniky_MM].[Vlastnici_3] = t.[Vlastnici_3] AND " +
    "[Dotazniky_MM].[FinInstitute_3] = t.[FinInstitute_3] AND " +
    "[Dotazniky_MM].[UradMS_3] = t.[UradMS_3] AND " +
    "[Dotazniky_MM].[Univerzity_16] = t.[Univerzity_16] AND " +
    "[Dotazniky_MM].[Konkurence_16] = t.[Konkurence_16] AND " +

```

```

        "[Dotazniky_MM].[Zakaznici_2] = t.[Zakaznici_2] AND " +
        "[Dotazniky_MM].[Dodavatele_2] = t.[Dodavatele_2] AND " +
        "[Dotazniky_MM].[Poradna_2] = t.[Poradna_2] AND " +
        "[Dotazniky_MM].[MM_T_INOVACE] = t.[T_INOVACE] AND " +
        "[Dotazniky_MM].[MM_T_FINANCE] = t.[T_FINANCE] AND " +
        "[Dotazniky_MM].[MM_T_HR] = t.[T_HR] AND " +
        "[Dotazniky_MM].[MM_T_KONTAKTY] = t.[T_KONTAKTY] AND " +
        "[Dotazniky_MM].[MM_T_PORADNA] = t.[T_PORADNA]";

// CONNECTION ANALYSIS SERVER + DB
strAnal_Conn = "Data Source=" + Start.GlobalVars.ServerName +
    ";Provider=msolap;Initial Catalog=C#_PREDIKCE_DOTAZNIKY";

AdomdConnection conn_anal = new AdomdConnection();
conn_anal.ConnectionString = strAnal_Conn;
conn_anal.Open();

AdomdCommand comm_predikce = new AdomdCommand();
comm_predikce.Connection = conn_anal;
comm_predikce.CommandText = commDMX_string;

excel.Application excel_APP = new excel.Application();
excel.Workbook excel_WB = excel_APP.Workbooks.Add();
excel.Worksheet excel_WS = excel_WB.ActiveSheet;

AdomdDataReader reader = comm_predikce.ExecuteReader();

excel_WS.Cells[1, 1].value = "ID";
excel_WS.Cells[1, 2].value = "Počet zaměstnanců";
excel_WS.Cells[1, 3].value = "Obor";
excel_WS.Cells[1, 4].value = "Management";
excel_WS.Cells[1, 5].value = "Právní forma";
excel_WS.Cells[1, 6].value = "FINANCE";
excel_WS.Cells[1, 7].value = "HR";
excel_WS.Cells[1, 8].value = "INOVACE";
excel_WS.Cells[1, 9].value = "KONTAKTY";
excel_WS.Cells[1, 10].value = "PORADENSKÁ ČINNOST";
excel_WS.Range["A1", "J1"].Font.Bold = true;

int j = 2;
while (reader.Read())
{
    for (int i = 0; i < reader.FieldCount; i++)
    {
        excel_WS.Cells[j, i + 1].value = reader[i];
    }
    j += 1;
}

SaveFileDialog SFD_excel = new SaveFileDialog();
SFD_excel.Filter = "Excel 2007|*.xlsx";
SFD_excel.FileName = "predikce.xlsx";
SFD_excel.ShowDialog();

MessageBox.Show("Predikce DONE!");

```

```
        excel_WS.SaveAs(SFD_excel.FileName);
        excel_APP.Quit();
        conn_anal.Close();
        Anal_Server.Disconnect();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        System.Windows.Forms.Application.Exit();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        if (listTables.SelectedItem != null)
        {
            train_table.Text = listTables.SelectedItem.ToString();
        }
    }

    private void button3_Click(object sender, EventArgs e)
    {
        if (listTables.SelectedItem != null)
        {
            pred_table.Text = listTables.SelectedItem.ToString();
        }
    }
}
}
```