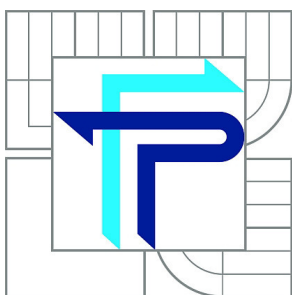


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA PODNIKATELSKÁ  
ÚSTAV INFORMATIKY**

FACULTY OF BUSINESS AND MANAGEMENT  
INSTITUTE OF INFORMATICS

# **ANALÝZA DATABÁZOVÉHO SYSTÉMU SPOLEČNOSTI A NÁVRH NA JEHO ZLEPŠENÍ**

ANALYSIS OF COMPANY'S DATABASE SYSTEM AND DESIGN OF INNOVATION

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**RADOVAN HLAVATÝ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. JIŘÍ KŘÍŽ, Ph.D.**

BRNO 2013

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Hlavatý Radovan**

---

Manažerská informatika (6209R021)

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 bakalářskou práci s názvem:

**Analýza databázového systému společnosti a návrh na jeho zlepšení**

v anglickém jazyce:

**Analysis of Company's Database System and Design of Innovation**

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:

CONNOLLY, T. M. a C. E. BEGG. Database systems: A Practical Approach to Design, Implementation, and Management. 4. vydanie. Harlow: Pearson Education, 2005. ISBN 0-321-21025-5.

HOFFER, J. A., V. RAMESH a H. TOPI. Modern database management. 10. vydanie. Upper Saddle River: Prentice Hall, 2010. ISBN 0-13-608839-2.

KŘÍŽ, J. a P. DOSTÁL. Databázové systémy. Brno: Akademické nakladatelství CERM, 2005. ISBN 80-214-3064-8.

SHARMA, N., L. PERNIU et al. Database fundamentals. 1. vydanie. Markham: IBM, 2010.

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

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2012/2013.

L.S.

---

doc. RNDr. Bedřich Půža, CSc.  
Ředitel ústavu

---

doc. Ing. et Ing. Stanislav Škapa, Ph.D.  
Děkan fakulty

V Brně, dne 26.05.2013

## **Abstrakt**

Táto bakalárska práca sa zaoberá analýzou aktuálneho stavu obchodnej spoločnosti, pôsobiacej v Slovenskej republike v oblasti posudkov a monitorovania životného prostredia. Analýza je zameraná na databázový systém spoločnosti a predkladá návrh možných riešení na jeho vylepšenie pri zohľadnení súčasných prostriedkov.

## **Abstract**

This bachelor thesis is focused on analysis of the current situation of the company, operating in the Slovak republic in the field of assessment and monitoring of the environment. The analysis is mostly focused on company's database system and proposal for possible solutions for its improvement taking into account the current resources.

## **Kľúčové slová**

databáza, databázový systém, SQL, SQL Server, MS Access, prenášanie dát

## **Key words**

database, database system, SQL, SQL Server, MS Access, data upsizing

## **Bibliografická citácia práce**

HLAVATÝ, R. *Analýza databázového systému spoločnosti a návrh na jeho zlepšení*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2013. 53 s.  
Vedoucí bakalářské práce Ing. Jiří Kříž, Ph.D.

## **Čestné prehlásenie**

Týmto čestne prehlasujem, že táto bakalárska práca je originál a je spracovaná samostatne. V bakalárskej práci som neporušil žiadne autorské práva, všetky informácie a poznatky čerpané z iných ako vlastných zdrojov sú riadne odcitované.

V Brne, dňa 28. mája 2013

.....

## **Pod'akovanie**

V prvom rade by som sa chcel poďakovať vedúcemu mojej bakalárskej práce Ing. Jiřímu Křížovi za jeho čas, odborné rady a systematické vedenie, ktoré mi poskytol pri vypracovávaní bakalárskej práce. Rovnako sa chcem poďakovať zamestnancom analyzovanej spoločnosti za ich ochotu odpovedať na otázky a poskytnutie potrebných informácií. V neposlednom rade ďakujem aj svojim rodičom za morálnu podporu a užitočné rady.

# Obsah

<b>ÚVOD</b> .....	<b>10</b>
<b>CIELE PRÁCE</b> .....	<b>11</b>
<b>1 TEORETICKÉ VÝCHODISKÁ</b> .....	<b>12</b>
<b>1.1 DATABÁZA</b> .....	<b>12</b>
<b>1.2 DATABÁZOVÝ MODEL</b> .....	<b>12</b>
<b>1.3 SYSTÉM RIADENIA BÁZY DÁT (SRBD)</b> .....	<b>12</b>
<b>1.4 RELAČNÝ SYSTÉM RIADENIA BÁZY DÁT</b> .....	<b>13</b>
1.4.1 ENTITA .....	13
1.4.2 ATRIBÚT .....	13
1.4.3 DOMÉNA .....	14
1.4.4 N-TICA .....	14
1.4.5 STUPEŇ .....	14
1.4.6 KARDINALITA .....	14
1.4.7 RELAČNÁ DATABÁZA .....	14
1.4.8 NORMALIZAČNÉ FORMY .....	14
1.4.9 VÄZBY MEDZI ENTITAMI .....	15
<b>1.5 MICROSOFT ACCESS</b> .....	<b>16</b>
<b>1.6 MICROSOFT SQL SERVER</b> .....	<b>17</b>
1.6.1 MECHANIZMUS USKLADNENIA DÁT (DATABASE ENGINE) .....	18
1.6.2 ANALYTICKÉ SLUŽBY .....	18
1.6.3 REPORTOVACIE SLUŽBY .....	18
1.6.4 INTEGRAČNÉ SLUŽBY .....	18
1.6.5 OZNAMOVACIE SLUŽBY .....	18
1.6.6 EDÍCIE SQL SERVERA .....	18
<b>2 ANALÝZA</b> .....	<b>21</b>
<b>2.1 ZÁKLADNÉ ÚDAJE O FIRME</b> .....	<b>21</b>
<b>2.2 PROFIL FIRMY</b> .....	<b>21</b>
<b>2.3 TECHNOLOGICKÁ INFRAŠTRUKTÚRA FIRMY</b> .....	<b>23</b>
2.3.1 INTERNETOVÉ PRIPOJENIE .....	23
2.3.2 INTRANET .....	23
<b>2.4 DATABÁZOVÝ SYSTÉM</b> .....	<b>24</b>
2.4.1 DATABÁZA OBJEKTOV (TECH-TAB) .....	25
2.4.2 DATABÁZA KVANTITY PODZEMNÝCH VÔD (GW) .....	29
2.4.3 DATABÁZA KVANTITY POVRCHOVÝCH VÔD (SW_DUN) .....	29
2.4.4 DATABÁZA KLIMATICKÝCH ÚDAJOV (KLIMA) .....	30
2.4.5 DATABÁZA PÔDNEJ VLHKOSTI (SOIL_MOISTURE) .....	31
2.4.6 VEĽKOSŤ DATABÁZ .....	31
<b>2.5 IDENTIFIKOVANÉ PROBLÉMY</b> .....	<b>32</b>
<b>3 NÁVRH RIEŠENIA</b> .....	<b>34</b>
<b>3.1 MIGRÁCIA DATABÁZ Z MICROSOFT ACCESS NA MS SQL SERVER</b> .....	<b>35</b>
3.1.1 ČASOVÁ NÁROČNOSŤ .....	35
3.1.2 FINANČNÁ NÁROČNOSŤ .....	36

<b>3.2 MIGRÁCIA DATABÁZ Z MICROSOFT ACCESS NA MS SQL SERVER A IMPLEMENTÁCIA</b>	
<b>WEBOVÉHO ROZHRAŇIA.....</b>	<b>36</b>
3.2.1 ČASOVÁ NÁROČNOSŤ .....	37
3.2.2 FINANČNÁ NÁROČNOSŤ .....	37
<b>3.3 PREROBENIE ŠTRUKTÚRY DATABÁZ V ROZHRAŇÍ MS SQL SERVER A IMPLEMENTÁCIA</b>	
<b>WEBOVÉHO ROZHRAŇIA NA PRÍSTUP K DATABÁZAM.....</b>	<b>38</b>
3.3.1 ČASOVÁ NÁROČNOSŤ .....	39
3.3.2 FINANČNÁ NÁROČNOSŤ .....	39
<b>3.4 ZHRNUTIE .....</b>	<b>39</b>
<b>3.5 VÝBER VHODNEJ VARIANTY .....</b>	<b>41</b>
3.5.1 VÝBER TABULIEK .....	41
3.5.2 INŠTALÁCIA SQL SERVERA.....	42
3.5.3 PRENESENIE A ÚPRAVA DÁT.....	42
3.5.4 KONTROLA DÁT.....	44
3.5.5 PREPOJENIE S ACCESS, DOLAĐOVANIE A TESTOVANIE.....	44
3.5.6 SKÚŠOBNÁ PREVÁDZKA .....	45
<b><u>4 PRAKTICKÁ REALIZÁCIA.....</u></b>	<b><u>46</u></b>
<b>4.1 VÝBER TABULIEK.....</b>	<b>46</b>
<b>4.2 INŠTALÁCIA SQL SERVERA .....</b>	<b>46</b>
<b>4.3 PRENESENIE A ÚPRAVA DÁT .....</b>	<b>47</b>
<b><u>ZÁVER.....</u></b>	<b><u>49</u></b>
<b><u>ZOZNAM POUŽITEJ LITERATÚRY .....</u></b>	<b><u>51</u></b>
<b><u>ZOZNAM OBRÁZKOV .....</u></b>	<b><u>53</u></b>
<b><u>ZOZNAM GRAFOV .....</u></b>	<b><u>53</u></b>
<b><u>ZOZNAM TABULIEK.....</u></b>	<b><u>53</u></b>
<b><u>ZOZNAM PRÍLOH.....</u></b>	<b><u>53</u></b>
<b><u>PRÍLOHA Č. 1: SCHÉMY .....</u></b>	<b><u>1</u></b>
<b><u>PRÍLOHA Č. 2: REPORT Z PRENÁŠANIA DÁT .....</u></b>	<b><u>6</u></b>

## Úvod

Dnešnú dobu, ktorú môžeme nazvať aj informačnou, je možné charakterizovať obrovským množstvom dát, ktoré ľudia v rôznych formách generujú. Dôležitosť údajov v posledných desaťročiach zaznamenala výrazný nárast. Rozvoj informačných technológií prispel k tomu, že sa stali dominantným prostriedkom pre uchovávanie, šírenie a používanie rôznych druhov informácií.

V najrozvinutejších krajinách sa do popredia dostáva pojem znalostnej spoločnosti, ktorej základnou črtou je intenzívne využívanie informačných technológií vo všetkých oblastiach života. Z ekonomického hľadiska to prispieva nielen k zvyšovaniu produktivity práce, ale aj kvality výstupov, pretože prístup k informáciám umožňuje posudzovanie a spracovanie informácií zo širšieho hľadiska.

Aby bolo možné obrovské množstvo údajov uchovávať a spracovávať boli vytvorené databázové systémy. Moderné databázové systémy umožňujú tieto dáta nielen uchovávať, ale zabezpečujú k nim aj jednoduchý prístup a dohľadateľnosť. Zo začiatku disponovali databázovými systémami najmä veľké spoločnosti, avšak dnes sú už nevyhnutnosťou na každej úrovni a nájdeme ich všade, od medzinárodných korporácií až po samostatných podnikateľov.

Aj obchodná spoločnosť, ktorej databázový systém som v mojej práci analyzoval, pracuje s veľkým množstvom rôznorodých údajov. Dôležitou súčasťou jej činnosti je spracovávanie údajov pochádzajúcich z monitorovania životného prostredia. Databázový systém, ktorý si obchodná spoločnosť vytvorila v prvých rokoch fungovania nebol už dlhšie obdobie aktualizovaný a práca s ním často nie je najefektívnejšia. Zlepšenie databázového systému, by spoločnosti malo priniesť rýchlejší prístup k údajom, vyššiu stabilitu aplikácie a bezpečnejšie uchovávanie dát. Práve analýza a návrh možných riešení na zlepšenie tohto systému je hlavnou témou mojej bakalárskej práce.

## **Ciele práce**

Cieľom bakalárskej práce je vylepšenie a zefektívnenie databázového systému v relatívne malej obchodnej spoločnosti, ktorá sa zaoberá monitorovaním životného prostredia a spracovaním rozsiahleho množstva údajov a správ z monitorovania prírodného prostredia. K tomuto cieľu je potrebné vykonať analýzu súčasného stavu databázového systému, v ktorom sú údaje uložené a navrhnúť kroky k optimalizácii jeho štruktúry a zvýšenia rýchlosti práce celého systému. Riešenie treba navrhnúť vo viacerých variantoch s poukázaním na ich výhody a nevýhody.

# **1 Teoretické východiská**

## **1.1 Databáza**

Databáza je virtuálne úložisko dát, ktoré sú usporiadané podľa určitej organizačnej štruktúry. Môžeme si ju predstaviť ako organizovaný súbor tabuliek, ktorých usporiadanie podporuje vykonávanie špecifických požiadaviek ako napríklad kombinovanie, preusporiadanie údajov a efektívne vyhľadávanie v nich podľa zadaných kritérií. Od logického modelu usporiadania údajov v databázach sa odvíjajú systémové procesy, jeho činnosť, ale aj požiadavky na hardware, na ktorom bude daný systém bežať (Sudolská, 2005).

## **1.2 Databázový model**

Pri tvorení databázy sa snažíme zachytiť obraz reálneho sveta, z ktorého údaje a informácie získavame. Na zjednodušenie tohto obrazu sa používa databázový model. Databázový model je teda zjednodušený pohľad na svet, ktorý je možné pretransformovať na potrebné údaje a informácie a tieto zaznamenať do samotnej databázy. Je to prostriedok na popis databázy (Kříž, 2005). Poznáme viacero druhov databázových modelov a to: hierarchický a sieťový boli používané v minulosti, no v súčasnosti najpoužívanejšie je relačný a začína sa používať aj objektovo orientovaný (Sudolská, 2005).

## **1.3 Systém riadenia bázy dát (SRBD)**

Zatiaľ, čo databáza predstavuje iba nejaké úložisko dát, SRBD predstavuje programy a softwarové nástroje, pomocou ktorých k dátam, uloženými v databáze, pristupujeme, upravujeme ich, organizujeme a udržiavame (Sharma, 2010).

SRBD je potrebný najmä z hľadiska riešenia zložitejších situácií, ktoré sa pri práci s databázou môžu vyskytnúť. Jedná sa napríklad o prístup, upravovanie alebo mazanie údajov v jednom súbore viacerými užívateľmi. Databázový systém rieši vzniknuté

situácie tak, aby sa užívatelia navzájom neovplyvňovali a aby sa predchádzalo konfliktom a nekonzistentnosti, ktorá by v tomto prípade mohla na úrovni dát nastat'. Databázový systém takisto zabezpečuje štandardizovaný prístup k dátam, nástroje na ich zálohovanie a prípadnú obnovu (Sharma, 2010).

## **1.4 Relačný systém riadenia bázy dát**

V dnešnej dobe je najpoužívanejším a najprepracovanejším databázovým systémom práve relačný databázový systém. Systém je postavený na relačnom dátovom modeli z roku 1970, ktorého autorom je E. F. Codd. Nespornou výhodou tohto systému je jeho jednoznačná a jednoduchá logická štruktúra. Všetky dáta sú usporiadané do relácií (tabuliek), z ktorých každá je pomenovaná a skladá sa z jednotlivých atribútov (stĺpcov), ktoré predstavujú jednotlivé vlastnosti entity. Údaje sú zapisované tak, že ku každému atribútu v n-tici (riadku) je priradená práve jedna hodnota (Sharma, 2010; Kříž, 2005).

### **1.4.1 Entita**

V terminológii databáz nazývame entitou jednotlivé základné stavebné prvky, ktoré používame na vyhotovenie databázového modelu. Každá entita je opísaná svojimi charakteristickými vlastnosťami (Sudolská, 2005). Entitou môže byť napríklad zákazník alebo monitorovací objekt.

### **1.4.2 Atribút**

Spomínané charakteristiky entít nazývame atribúty. V prípade zákazníka, by atribútmi mohli byť jeho meno, priezvisko, adresa atď. V prípade pozorovacieho objektu pracujeme napr. s technickými údajmi objektu (hĺbka, priemer), alebo meranými veličinami (hladina vody, chemické zloženie vody a pod.). Údaje zapisované ako atribúty sa vo väčšine prípadov vyberajú z určitej množiny, ktorú nazývame doménou atribútu (Sudolská, 2005).

### **1.4.3 Doména**

Doménou je množina povolených hodnôt, ktoré môže jeden alebo viac atribútov nadobudnúť. Doména definuje aký dátový typ môže byť použitý pre daný atribút a takisto interval hodnôt. Na príklade zákazníka sa môže jednať o atribút mesto, ktorý sa bude skladať so znakov o maximálnej dĺžke 25. Ak by sme hovorili o dátume jeho narodenia, použili by sme dátový typ dátum, kde vyberáme z množiny možných dátumov (Connolly, 2005).

### **1.4.4 N-tica**

Predstavuje riadok tabuľky. Jednotlivé elementy relácie sú zobrazené ako riadky tabuľky, pričom na ich poradí nezáleží – je jedno, či riadky navzájom vymeníme, stále sa bude jednať o tú istú reláciu (Connolly, 2005).

### **1.4.5 Stupeň**

Stupeň relácie určujeme na základe počtu atribútov, ktoré obsahuje. Tzn. entitu, ktorá má 5 vlastností bude označovať ako relácia 5-teho stupňa (Connolly, 2005).

### **1.4.6 Kardinalita**

Kardinalita relácie nám pre zmenu vyjadruje počet n-tíc v relácií. So zmenou počtu záznamov v tabuľke sa mení aj kardinalita relácie (Connolly, 2005).

### **1.4.7 Relačná databáza**

Relačná databáza je súbor normalizovaných relácií, z ktorých každá je jednoznačne určená názvom. Použité relácie musia byť náležite štruktúrované. Postup a pravidlá, ktorými sa dopracujeme k požadovanej štruktúre označujeme pojmom normalizácia (Connolly, 2005).

### **1.4.8 Normalizačné formy**

Tento pojem sa používa pre správny návrh tabuliek, pri ktorom sa riadime istými pravidlami a dodržiavaním štandardov (noriem). Na základe štruktúry tabuliek rozlišujeme 4 základné normalizačné formy (Kříž, 2005).

## 1. Normalizačná forma

Prvá normalizačná forma hovorí, že všetky atribúty tabuľky sú atomické, t.j. nedajú sa ďalej deliť (Kříž, 2005). Ako príklad môžeme uviesť atribút Meno zákazníka. Ak uvedieme meno a priezvisko zákazníka do jedného stĺpca, daná tabuľka nespĺňa kritérium prvej normalizačnej formy, pretože tento atribút môžeme rozdeliť na dva stĺpce, a to Meno a Priezvisko.

## 2. Normalizačná forma

Tabuľka vyhovuje druhej normalizačnej forme práve vtedy, ak spĺňa kritéria prvej normalizačnej formy a navyše každý z jej atribútov, ktorý nie je primárnym kľúčom je na primárnom kľúči plne závislý (Kříž, 2005).

## 3. Normalizačná forma

Tabuľka je v tretej normalizačnej norme, ak spĺňa kritéria prvej, zároveň aj druhej normalizačnej formy a navyše neobsahuje žiadny atribút, ktorý nie je primárnym kľúčom, ktorý by bol závislý na inom kľúči (Kříž, 2005). Napríklad, ak v tabuľke Zákazník máme okrem iného atribúty Mesto a PSČ. Táto tabuľka nie je v tretej normálnej forme pretože mesto je závislé na potenciálnom kľúči PSČ. Na to aby sme dostali tabuľku do tretej normálnej formy, musíme osamostatniť atribúty Mesto a PSČ v ďalšej tabuľke, kde PSČ bude primárnym kľúčom.

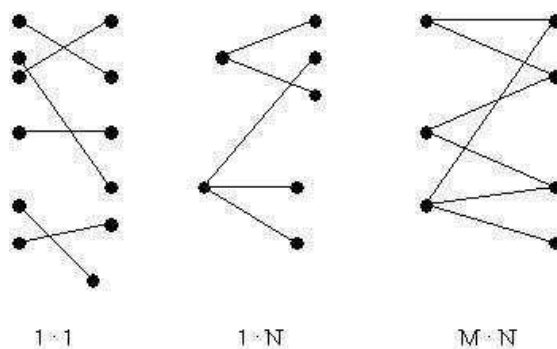
### Boyce-Coddova normalizačná forma

Na splnenie štvrtej normalizačnej formy musí tabuľka spĺňať kritéria všetkých predošlých normalizačných foriem a navyše musí riešiť viachodnotové závislosti (Sharma, 2010; Kříž, 2005).

### **1.4.9 Vázby medzi entitami**

Samotné údaje uložené v databáze nemajú pre nás veľkú výpovednú hodnotu a preto, tak ako v reálnom svete aj v databázach berieme do úvahy vzťahy, ktoré medzi entitami

existujú. Na základe počtu prvkov, medzi ktorými existuje väzba rozlišujeme tri základné typy týchto väzieb: 1:1, 1:N a M:N – **Obrázok č. 1** (Sudolská, 2005).



**Obrázok č. 1: Tri základné typy väzieb medzi entitami**  
(Zdroj: Sudolská, 2005)

## 1.5 Microsoft Access

Access je systém riadenia bázy dát od spoločnosti Microsoft, ktorý integruje relačné databázy, užívateľské prostredie a vývojárske nástroje. Access je súčasťou obsiahlejších verzií MS Office, no môže byť dokúpený aj samostatne. Access je určený prevažne pre tvorbu menej rozsiahlych databáz v malých a stredne veľkých podnikoch, kde sa využíva napríklad na spravovanie údajov o zákazníkoch, obchodoch a pod. Užívateľské prostredie je prispôbené tak, aby sa s ním naučili rýchlo pracovať aj nenároční používatelia bez predchádzajúcich skúseností s databázami, no ktorí majú skúsenosti s inými programami z balíku MS Office. Pre náročnejších užívateľov je v Accesse možnosť programovania a prispôbovania rôznych funkcií v jazyku Visual Basic for Applications (VBA), s ktorého podporou sa môžeme stretnúť aj v prostredí ďalších programov balíka MS Office. Vďaka použitiu VBA, či makier je možné rozšíriť jednoduché riešenia na takmer profesionálne, s pokročilou automatizáciou, riešením chýb a overovaním dát. Pri práci s dátami je veľmi nepohodlné a neprehľadné pristupovať priamo do tabuliek. V Accesse je možné vytvoriť formuláre a reporty, ktoré poskytujú ucelenejší pohľad na dáta. Oba objekty majú spoločné, že je možné nadefinovať ktoré dáta z daného záznamu, resp. záznamov je potrebné vidieť, tak aby sa koncový užívateľ nestratil v neprehľadnom množstve údajov (Groh, 2007).

Formulár – je databázový objekt, ktorý slúži na vytvorenie užívateľského rozhrania pre databázovú aplikáciu. Užívatelia týmto spôsobom pristupujú k dátam uloženým v databáze, pričom pracujú práve s jedným záznamom. Formulár užívateľom ponúka tri základné funkcie pre prácu so záznamami:

- prehliadanie
- upravovanie
- pridávanie

Prehľadnosť a ľahšiu orientáciu je možné zlepšiť ešte aj vhodne zvoleným, užívateľsky prívetivým grafickým dizajnom a vysvetľujúcimi popismi (Microsoft, 2013a).

Report – databázový objekt, ktorý slúži na zobrazenie záznamov uložených v databáze. Užívateľom poskytuje ucelený prehľad údajov, ktoré môžu byť z jednej, ale aj viacerých tabuliek. Report je možné prispôbiť podľa toho aké dáta potrebuje užívateľ vidieť, pričom je používaný aj na zobrazovanie sumárnych dát. Tak ako pri formulári je možné vďaka vhodnému grafickému dizajnu a popismi docieľiť efektívne a prehľadné zobrazenie dát (Microsoft, 2013b).

Na ukladanie databáz používa Access svoj vlastný formát \*.accdb (v starších verziách \*.mdb), avšak umožňuje pracovať aj s rôznymi inými databázovými formátmi, ako je napríklad Microsoft SQL Server, ale aj databázové riešenia od Oracle alebo Sybase. V tomto prípade slúži Access ako front-end aplikácia a tabuľky sú uložené na jednom zo spomínaných databázových riešení a predstavujú takzvaný back-end. Komplexnejšie riešenia je možné dosiahnuť prepojením Accessu s ďalšími programami ako Excel alebo Outlook (Groh, 2007).

## **1.6 Microsoft SQL Server**

SQL Server je predovšetkým relačný systém riadenia bázy dát, avšak ponúka mnoho ďalších funkcií, ktoré rozširujú základné povinnosti relačného systému. Medzi najvýznamnejšie môžeme zaradiť bohaté možnosti reportovania, analýzu dát alebo dolovanie dát (Wood, 2007).

### **1.6.1 Mechanizmus uskladnenia dát (Database engine)**

Mechanizmus uskladnenia dát typu OLTP (Online Transaction Processing) je hlavným komponentom SQL Servera a počas vývoja prešiel mnohými zmenami, ktoré sa v súčasnej verzii SQL Server 2012 prejavujú efektivitou a vyladenosťou. Tento fakt je veľmi dôležitý, pretože ako hlavný komponent je mechanizmus zodpovedný za efektívne ukladanie, získavanie a manipuláciu s dátami (Wood, 2007).

### **1.6.2 Analytické služby**

Analytické služby zabezpečujú OLAP (Online Analytical Processing) a dolovanie dát pre aplikácie Business Intelligence (BI). Poskytujú prostredie na detailnú analýzu multidimenzionálne štruktúrovaných dát, ktoré boli vytvorené na základe denormalizovaných dát a dát agregovaných z rôznych zdrojov. Na základe takejto analýzy môže firma odhaliť skryté väzby a zákonitosti (Wood, 2007).

### **1.6.3 Reportovacie služby**

Reportovacie služby sú postavené na riešení webových služieb a slúžia na návrh, prístupnosť a manažovanie dynamických reportov s webovým základom, ale aj tradičných papierových reportov (Wood, 2007).

### **1.6.4 Integračné služby**

SQL Server Integration Services (SSIS) predstavujú podnikový nástroj zabezpečujúci vybranie, transformáciu a načítanie dát (Wood, 2007).

### **1.6.5 Oznamovacie služby**

Oznamovacie služby umožňujú vytvárať aplikácie, ktoré generujú a odosielajú notifikácie viazané na dáta, respektíve na konkrétne udalosti. Oznámenia môžu byť doručované formou emailu alebo inými vlastnými metódami (Wood, 2007).

### **1.6.6 Edície SQL Servera**

Microsoft SQL Server má 3 hlavné edície prispôbené na rôzne potreby zákazníkov, pričom každá z edícií je dostupná v 32 aj 64 bitovej verzii. Spomínané edície sú:

- Standard edition
- Business Intelligence edition
- Enterprise edition

Okrem toho však nájdeme v ponuke aj ďalšie špecializované edície, ktoré sú určené pre spoločnosti so špecifickejšími požiadavkami. Nájdeme tu napríklad:

- Express edition
- Web edition
- Developer edition

(Mistry, 2012).

### **Enterprise edition**

Edícia Enterprise je navrhnutá tak, aby pokryla aj tie najvyššie požiadavky veľkých datacentier zabezpečením nekompromisného výkonu a stálej prístupnosti pre aplikácie, schopnosťou, vizualizovanými prostriediami a centralizovanými alebo externe orientovanými riešeniami BI (Mistry, 2012).

### **Business Intelligence edition**

Edícia Business Intelligence poskytuje firmám celú škálu funkcií a schopností pokročilých reportov a analýz pre náročné aplikácie BI. Je určená pre organizácie, ktoré potrebujú pokročilé nástroje BI, ale ktoré nevyžadujú výkon a prispôbitelnosť transakčného spracovania (OLTP) (Mistry, 2012).

### **Standard edition**

Edícia Standard je platforma na manažovanie dát, ktorá je vhodná pre stredné a menšie organizácie alebo riešenia pre oddelenia väčších korporácií, ktorým stačí obmedzená funkcionálnosť pre BI aplikácie. Neobsahuje natoľko pokročilé funkcie a vychytávky ako dve vyššie verzie, no zameriava sa na ľahkosť používania a manažovanie (Mistry, 2012).

### **Express edition**

Edícia Express je voľne dostupná a je určená najmä neprofesionálnym developerom, ľuďom, ktorí si chcú SQL Server vyskúšať, ale môže sa hodiť aj malým firmám alebo podnikateľom, ktorí potrebujú spravovať jednoduchšie databázy. Okrem výrazného obmedzenia pokročilých funkcií má táto edícia aj obmedzenia na využiteľnosť hardvéru. Podporuje maximálne jeden procesor, 1 GB operačnej pamäte a maximálna veľkosť databáze je limitovaná do 10 GB (Mistry, 2012).

### **Web edition**

Edícia Web je cenovo dostupnejšia ako edície Enterprise a Standard a zameriava sa na poskytovateľov služieb, ktorí hostujú aplikácie s webovým rozhraním. Na rozdiel od edície Express nie je obmedzená veľkosť databáz, sú podporované až štyri procesory a 64 GB operačnej pamäte. Chýbajú tu najmä pokročilejšie funkcie, obsiahnuté vo vyšších edíciách.

### **Developer edition**

Edícia Developer, ako je zrejme už z názvu, je určená vývojárom. Funkcionalita a možnosti sú zhodné s edíciou Enterprise, ale môže byť použitá iba na účel vývoja, testovania alebo predvádzania. Bez nutnosti opätovnej inštalácie je možné prejsť z tejto edície na edíciu Enterprise.

## **2 Analýza**

### **2.1 Základné údaje o firme**

Názov: Obchodná spoločnosť XY

Sídlo: Ulica č,

PSČ Bratislava

Slovenská republika

### **2.2 Profil firmy**

V rámci mojej práce som sa zameril na analýzu technologického a softvérového vybavenia malej obchodnej spoločnosti. Spoločnosť má 20-ročnú históriu a zamestnáva vysokošpecializovaných odborníkov prírodovedného zamerania. Počas svojej existencie zamestnávala 5-14 ľudí na trvalý pracovný pomer a krátkodobo do 30 externých odborníkov ročne.

Firmu XY som si zvolil preto, lebo svoju činnosť má úzko previazanú s používaním databáz. Mojou úlohou bolo analyzovať používaný databázový systém, identifikovať problémy pri práci s ním a prípadne navrhnúť kroky na jeho zlepšenie.

Firma XY je malou súkromnou spoločnosťou ponúkajúcou pomerne široké služby v oblasti využívania a ochrany vodnej zložky životného prostredia. Spoločnosť ponúka špeciálne hydrogeologické služby zamerané na využívanie a ochranu podzemných a povrchových vôd, projektovanie, manažment a rozvoj vodných zdrojov, numerické modelovanie prúdenia povrchových a podzemných vôd, monitorovanie prírodného prostredia a environmentálny prieskum súvisiaci s vodou, ako aj posudzovanie vplyvov na životné prostredie (SEA, EIA).

Ako podporné aktivity firma realizuje prednášky, vydáva publikácie a vyvíja špecializovaný software, účelové databázy a geografické informačné systémy (GIS).

V súčasnosti má firma sedem zamestnancov, ktorí majú stredne pokročilé až pokročilé znalosti práce s počítačom. Na kancelárske práce, písanie textov a spracovávanie tabuliek sa používajú programy prevažne na platforme Windows. Na sieťové služby sa využíva Linux. V rámci firmy bol vyvinutý komplexný databázový systém v prostredí MS Access, ktorého niektorým častiam však chýba užívateľské rozhranie. To nájdeme iba v najpoužívanejších čiastkových databázach.

Samotné údaje v databázach spoločnosť prevažne objednáva u externých dodávateľov, prípadne ich dostáva od objednávateľov. Merania, analýzy a pozorovania v rámci monitoringu každý dodávateľ poskytuje v inom formáte a dokonca sa bežne vyskytuje aj prípad nekonzistentnosti údajov u jedného dodávateľa. Z tohto dôvodu je veľmi zložité zautomatizovať import dát do firemnej databázy a preto sa tento proces do veľkej miery prevádza aj manuálne. Takéto importovanie je časovo veľmi náročné a pri jeho realizácii je často potrebné overovať aj nejasnosti a chyby v údajoch. Táto skutočnosť ovplyvnila a ovplyvňuje aj samotnú štruktúru databáz, pretože z dôvodu nejasností sú vytvárané aj tabuľky resp. atribúty, ktoré by za normálnych okolností neboli potrebné. Ich použitie je mnohokrát iba krátkodobé a pri následnej práci s dátami majú zanedbateľný význam.

Problém predstavuje aj to, že určité hydrologické údaje, ako je napríklad prietok vody, podliehajú úpravám, ktoré sú realizované až po skončení hydrologického roka. Znamená to, že údaje ktoré sú poskytované denne sa na konci roka revidujú a po odsúhlasení sa zasielajú opravené a overené údaje. Nakoľko je potrebné vo firme pracovať s týmito predbežnými a neoverenými dennými údajmi aj počas roka, vo firme sa to rieši vytvorením ďalších pomocných tabuliek alebo databáz.

Ďalšou nevýhodou je označovanie jednotlivých pozorovacích objektov, pretože označenie objektov u dodávateľov sa líši, a väčšinou je rozdiel aj pri označení v teréne. V rámci spoločnosti bol tento problém vyriešený tabuľkami slúžiacimi na prevod rozličných identifikátorov na vlastné označenie, ktoré sa používajú pri importe údajov. Podobne, ani tieto tabuľky nemajú pri ďalšej práci s dátami žiadny význam.

K databázovému systému nie sú stanovené žiadne prístupové práva a v princípe časť užívateľov, ktorí sa vyznajú v databázovom systéme, môže do neho zasahovať. Ani

zálohy databázového systému nie sú automatické a robia sa spôsobom vytvorenia duplicitného súboru k istému dátumu alebo inkrementálnym backupom. Ani tieto zálohy nie sú pravidelné a pri nečakanom zlyhaní systému je nutné znova naimportovať dáta, ktoré boli pridané od poslednej zálohy. Rovnakým nebezpečenstvom je aj náhodné vymazanie alebo prepísanie údajov.

## **2.3 Technologická infraštruktúra firmy**

### **2.3.1 Internetové pripojenie**

Firma má zabezpečené dátovo neobmedzené pripojenie na internet o rýchlosti 5120/384 Kb/s postavené na technológii DSL od poskytovateľa Telekom a.s. Na bežnú prevádzku firmy uvedená kapacita postačuje. Problémy sa vyskytujú najmä v prípade uploadov rozsiahlejších súborov alebo pri distribúcii súborov viacerým odberateľom naraz. Vtedy môžu nastať výpadky pripojenia a čas potrebný na stiahnutie súborov je príliš dlhý. Takéto prípady sa však vyskytujú zriedka.

### **2.3.2 Intranet**

Na prvej úrovni je NAT, na ktorý sú pripojené zariadenia, ktoré majú prístup iba k internetu, ale kvôli bezpečnosti nemajú prístup k intranetu. Medzi týmito zariadeniami je 100 Mb/s sieť. Napojený je na ňu web server a WiFi-router slúžiaci ako prístupový bod. Web server beží neustále 24h denne, pretože poskytuje službu DHCP a prístup na webové stránky spoločnosti. Na tomto serveri sa nachádza Firewall.

Vnútropodniková sieť pozostáva z dvoch 1Gb switchov spojených na priamo, do ktorých sú zapojené koncové zariadenia v hviezdicových topológiách. V intranete sa nachádza aj dátový server NAS, ktorý poskytuje dáta pomocou technológie CIFS. Server značky Synology netreba v bežnej prevádzke obsluhovať, beží na ňom systém Samba, postavený na Linuxe. Tento server slúži zároveň ako print server, takže spracováva všetky dotazy na tlač a priraduje ich jednotlivým tlačiarňam.

## 2.4 Databázový systém

Spoločnosť nedisponuje SQL databázovým serverom. Všetky údaje, s ktorými pracuje, sú uložené na disku v niekoľkých databázach. Na správu databáz sa používa program Microsoft Access, ktorý má firma zakúpený vo verzii pre developerov. V rámci MS Access firma vyvinula vlastné prostredie na správu čiastkových databáz. Prostredie umožňuje pripájať rôzne databázy, importovať predpripravené údaje do definitívnych štruktúr, prehliadanie údajov, tlač formulárov, vykresľovanie grafov, ako aj export údajov pre vopred stanovené účely.

Spoločnosť nemá ucelený firemný informačný systém. Účtovníctvo, agenda zamestnancov, zoznam odberateľov a dodávateľov sú spravované ekonomickým softvérom Datalock, ktorý má databázy tiež postavené na báze MS Access.

Čiastkové databázy boli pri vyvíjaní zámerne rozčlenené podľa predmetu monitoringu kvôli rýchlejšej práci s množstvom údajov, ktoré sú limitované hardvérovým a programovým vybavením firmy, ako aj možnosťami programu MS Access.

Rozdelenie databáz podľa predmetu pozorovania je nasledovné:

- kvantita vôd
  - povrchová voda
  - podzemná voda
- kvalita vôd
  - povrchová voda
  - podzemná voda
- monitoring pôdy
  - fyzikálno-chemické charakteristiky pôd
  - pôdna vlhkosť
- klimatické údaje
- monitoring lesa
- biologický monitoring

Osobitne bola vytvorená databáza objektov a technických informácií o monitorovacích plochách a pozorovacích objektoch, na ktorých sa realizujú jednotlivé merania. Táto databáza je prepojená na ostatné čiastkové databázy.

V nasledujúcej časti sa zameriam na popis štruktúry vybraných databáz a prepojenie medzi nimi. Tabuľky nachádzajúce sa v databázach stručne charakterizujem a ku každej z nich uvediem výpis kľúčov – primárny kľúč (PK) a cudzie kľúče (FK).

#### **2.4.1 Databáza objektov (tech-tab)**

V databáze sú uložené základné údaje o pozorovaných objektoch a lokalitách (Príloha č. 1, Obr. 1). Údaje v tejto databáze charakterizujú jednotlivé objekty, na ktorých prebieha monitoring a na ktorých sú realizované merania. Databáza obsahuje označenie objektov v teréne, informáciu o vlastníkoch respektíve správcoch objektov, údaje o ich geografickej polohe (súradnice x, y) a nadmorskej výške. V databáze sú rozlíšené typy objektov (napr. vodočet, plocha, pozorovací vrt), ďalej sú uvedené podrobné informácie o technickom prevedení pozorovacích objektov a informácie o metodike merania a kvalite údajov.

Tabuľky:

Základnou tabuľkou je tabuľka **OBJ**, kde nájdeme hlavné údaje o objekte ako jeho názov, typ, správcovskú organizáciu, lokalitu alebo odkaz na zemepisnú polohu a nadmorskú výšku. Na túto tabuľku sú naviazané ďalšie tabuľky, ktoré obsahujú doplnujúce, podrobnejšie informácie.

PK: obj\_id

FK: org\_id – previazanie na tabuľku **ORG**, ktorá predstavuje číselník organizácií vlastniacich alebo spravujúcich daný objekt

xy\_id – odkazuje sa na tabuľku zemepisných súradníc x, y (tabuľka **XY**)

z\_id – odkazuje sa na tabuľku nadmorských výšok (tabuľka **Z**), ktoré sa môžu v čase meniť (na rozdiel od x, y), pričom nadmorská výška nemusí byť pre všetky objekty známa

Pre prípad viacúrovňových pozorovacích objekt (v jednom pozorovacom objekte nachádza viac pozorovaných úrovní) je zavedená pomocná tabuľka **OBJ\_SUB**.

PK: zložený z obj\_id a obj\_ids

FK: obj\_id z tabuľky OBJ

z\_id z tabuľky Z

S tabuľkou OBJ sú previazané ďalšie tri tabuľky, **OBJ\_W**, **OBJ\_R** a **OBJ\_S**, ktoré sú s tabuľkou OBJ prepojené atribútom obj\_id a obsahujú dodatočné informácie (atribúty) k jednotlivým typom objektov.

Tabuľka **OBJ\_W** je tabuľka pozorovacích objektov (vrtov), ktorá obsahuje podrobné technické informácie o vrte, akými sú napr. materiál, priemer, spôsob zabudovanie, nadmorské výšky alebo výšky nad terénom jednotlivých častí objektu.

PK: obj\_id z tabuľky OBJ

FK: obj\_id

Tabuľka **OBJ\_S** je tabuľka monitorovacích plôch, ktorá obsahuje zoznam pozorovacích objektov nachádzajúcich sa na príslušnej monitorovacej ploche.

PK: obj\_id z tabuľky OBJ

FK: obj\_id

Tabuľka **OBJ\_R** je tabuľka monitorovaných miest na vodných tokoch, v ktorej sa nachádzajú informácie na akej rieke sa miesto nachádza, riečny kilometer a v prípade širokých vodných tokov aj umiestnenie monitorovaného miesta (pravá alebo ľavá strana, stred).

PK: obj\_id z tabuľky OBJ

FK: obj\_id

riv\_id odkazuje na číselník (**RIV**), v ktorom je zoznam riek

Pre tabuľky **OBJ\_S** a **OBJ\_R** sa použil rovnaký postup na rozlíšenie viacerých podobjektov alebo druhu monitorovanej informácie ako pri tabuľke **OBJ**. Tieto informácie sú zachytené v tabuľkách **OBJ\_S\_SUB** a **OBJ\_R\_SUB**.

Tabuľka **OBJ\_S\_SUB**

- PK: zložený z obj\_id a obj\_ids
- FK: obj\_id z tabuľky **OBJ\_S**  
obj\_id a obj\_ids z tabuľky **OBJ\_SUB**  
xy\_id z číselníku zemepisných súradníc XY

Tabuľka **OBJ\_R\_SUB**, obsahuje oproti tabuľke **OBJ\_R** aj záznam o monitorovanej zložke.

- PK: zložený z obj\_id a obj\_ids
- FK: obj\_id z tabuľky **OBJ\_S**  
obj\_id a obj\_ids z tabuľky **OBJ\_SUB**

S tabuľkou pozorovacích objektov **OBJ\_W** sú prepojené dve pomocné tabuľky, ktoré obsahujú ďalšie detailné informácie o technickej realizácii pozorovacieho objektu. V tabuľke **OBJ\_W\_PZM** sa nachádzajú informácie o hĺbke pozorovacieho vrtu, materiáli filtra, dĺžke kalníka, a pod.

- PK: zložený z obj\_id a obj\_ids
- FK: obj\_id z tabuľky **OBJ\_W**  
obj\_id a obj\_ids z tabuľky **OBJ\_SUB**

V tabuľke **OBJ\_W\_PZM\_SCR** je informácia o filtračnej časti pozorovacieho objektu.

- PK: obj\_id, obj\_ids a PERFod
- FK: obj\_id a obj\_ids z tabuľky **OBJ\_W\_PZM**

V tabuľke **MON** sa nachádzajú informácie, podľa ktorých sú jednotlivé objekty rozdelené do skupín podľa účelu monitorovania alebo podľa účelu vyhodnotenia. Nachádza sa tu aj údaj o období zaradenia daného objektu do tej-ktorej účelovej skupiny.

PK: táto tabuľka nemá definovaný primárny kľúč  
FK: obj\_id z tabuliek OBJ a OBJ\_SUB  
obj\_ids z tabuľky OBJ\_SUB  
mon\_id z tabuľky MON\_REM

V tabuľke **MON\_REM** sú uvedené jednotlivé účelové skupiny objektov (napr. podľa zákazky, územného členenia, reprezentatívnosti) spolu s krátkou charakteristikou.

PK: mon\_id  
FK: org\_id z tabuľky ORG

V databáze sa ďalej nachádza tabuľka **DATA\_KOD** obsahujúca zoznam typov dát, metódu akým spôsobom boli údaje získané alebo stanovené, kvalitu údajov a pod. Táto tabuľka je prepojená na iné databázy.

PK: src\_dbk  
FK: org\_id z tabuľky ORG.

Ďalšou tabuľkou je už spomínaný číselník **RIV**, ktorý reprezentuje zoznam vodných tokov. Okrem primárneho kľúča obsahuje iba údaj o názve rieky.

PK: riv\_id

Tabuľka **XY**, ktorá bola spomenutá už vyššie, obsahuje zemepisné súradnice objektu. Nachádza sa tu aj záznam o dátume a metóde stanovenia súradníc.

PK: xy\_id  
FK: obj\_id z tabuľky OBJ  
mth\_id z tabuľky MTH, kde je uvedená metóda stanovenia súradníc.

Ďalšou tabuľkou so súradnicami je tabuľka **Z**, ktorá obsahuje údaje o nadmorskej výške objektu. Keďže pri nadmorskej výške je väčšia pravdepodobnosť, že sa údaj bude meniť, v tabuľke sa uvádza aj časové obdobie platnosti súradnice. Rovnako je tu uvedená informácia o metóde stanovenia súradnice.

PK: z\_id  
FK: obj\_id z tabuľky OBJ  
mth\_id z tabuľky MTH

Poslednou tabuľkou, ktorú tu nájdeme je tabuľka **MTH** obsahujúca zoznam metód, ktorými boli súradnice x, y a z stanovené. Je prepojená s tabuľkami XY a Z.

PK: mth\_id

#### 2.4.2 Databáza kvantity podzemných vôd (GW)

V tejto databáze sa nachádzajú údaje o meraniach hladín podzemných vôd (Príloha č. 1, Obr. 2). Pre účely vyhodnotenie jednotlivých časových radoch meraní je dôležité mať aj informácie o pozorovanom objekte, preto tu nájdeme odkazy na technickú databázu (databázu objektov) **tech-tab**, v ktorej sú detailné informácie o jednotlivých objektoch (Príloha č. 1, Obr. 1).

Prevažná väčšina tabuliek, ktoré sa nachádzajú v tejto databáze, má pôvod v databáze objektov. Tieto tabuľky boli popísané v rozbere databázy **tech-tab**. Z tejto databázy tu nájdeme OBJ, OBJ\_SUB, OBJ\_W, OBJ\_W\_PZM, OBJ\_W\_PZM\_SRC, MON, XY, Z, DATA\_KOD. Tabuľka s údajmi o hladinách podzemných vôd je tabuľka **HED**. Okrem údajov o hladine v nadmorskej výške tu nájdeme dátum merania, použitú metódu merania, hĺbku hladiny od odmerného bodu a údaj o teplote vody.

PK: zložený z obj\_id, obj\_ids, dyt, src\_dbk  
FK: obj\_id z tabuľky OBJ  
obj\_id a obj\_ids z tabuľky OBJ\_SUB  
src\_dbk z tabuľky DATA\_KOD, ktorá obsahuje zoznam metód použitých pri meraní

#### 2.4.3 Databáza kvantity povrchových vôd (SW\_DUN)

Databáza kvantity povrchových vôd je obdobná databáza ako databáza kvantity podzemných vôd (Príloha č. 1, Obr. 3). Údaje o meraniach hladín povrchových vôd a údaje o prietoku povrchových vôd sa nachádzajú v tabuľkách **HED\_FIN** a **DIS\_FIN**.

Rovnako sa tu nachádzajú tabuľky pripojené z databázy objektov (Príloha č. 1, Obr. 1), ktoré boli popísané vyššie: OBJ, OBJ\_SUB, OBJ\_R, OBJ\_R\_SUB, RIV, MON, XY, Z, MON\_REM a MTH.

Tabuľka **HED\_FIN** obsahuje údaje časových radov meraných hladín povrchových vôd.

PK: zložený z ipt, dyt, src\_dbk a oprava  
FK: ipt z tabuľky OBJ, kde je uvedené ako obj\_id  
src\_dbk z tabuľky DATA\_KOD

Tabuľka **DIS\_FIN** obsahuje údaje časových radov stanovených prietokov.

PK: zložený z ipt, dyt, src\_dbk a oprava  
FK: ipt z tabuľky OBJ, kde je uvedené ako obj\_id  
src\_dbk z tabuľky DATA\_KOD

#### **2.4.4 Databáza klimatických údajov (Klima)**

V tejto databáze sa nachádzajú údaje základných meteorologických charakteristík, ako sú merania teploty vzduchu a zrážkové úhrny (Príloha č. 1, Obr. 4). Tieto údaje slúžia ako podporné údaje pre hodnotenie vývoja vlhkosti pôdy. Aj v tejto databáze väčšina tabuliek pochádza z databázy **tech-tab** (Príloha č. 1, Obr. 1). Nachádzajú sa tu tabuľky OBJ, OBJ\_SUB, OBJ\_S, OBJ\_S\_SUB, XY, Z, MTH, DATA\_KOD. Samotné údaje sú obsiahnuté v tabuľkách **T-TEPLOTA\_VZDUCHU** a **T-ZRAZKY**.

Tabuľka **T-TEPLOTA\_VZDUCHU** obsahuje časové rady s priemernými dennými údajmi z meraní teploty vzduchu.

PK: zložený z obj\_id, obj\_ids a datum  
FK: obj\_id z tabuľky OBJ  
mtd\_dbk z tabuľky DATA\_KOD

Tabuľka **T-ZRAZKY** obsahuje časové rady s údajmi denných zrážkových úhrnov.

PK: zložený z ipt a datum  
FK: mtd\_dbk z tabuľky DATA\_KOD

### 2.4.5 Databáza pôdnej vlhkosti (soil\_moisture)

V poslednej z analyzovaných databáz sa nachádzajú údaje z meraní pôdnej vlhkosti (Príloha č. 1, Obr. 5). Aj táto databáza obsahuje tabuľky napojené z databázy objektov **tech-tab** (Príloha č. 1, Obr. 1). Nachádzajú sa tu tabuľky OBJ, OBJ\_SUB, OBJ\_S, OBJ\_S\_SUB, OBJ\_W, OBJ\_W\_PZM, OBJ\_W\_PZM\_SRC, XY, Z, MON, MON\_REM, MTH a DATA\_KOD. Údaje sú ukladané v dátových tabuľkách **VLHKOST-ALL** a **VLHKOST\_HP-ALL**.

V tabuľke **VLHKOST-ALL** sa nachádzajú časové rady údajov o pôdnej vlhkosti, ktoré boli namerané na jednotlivých monitorovacích plochách v objektoch na meranie pôdnej vlhkosti. Zaznamenávané údaje sú reprezentované označením lokality, dátumu merania a hĺbkou merania, ku ktorej sa konkrétna hodnota viaže.

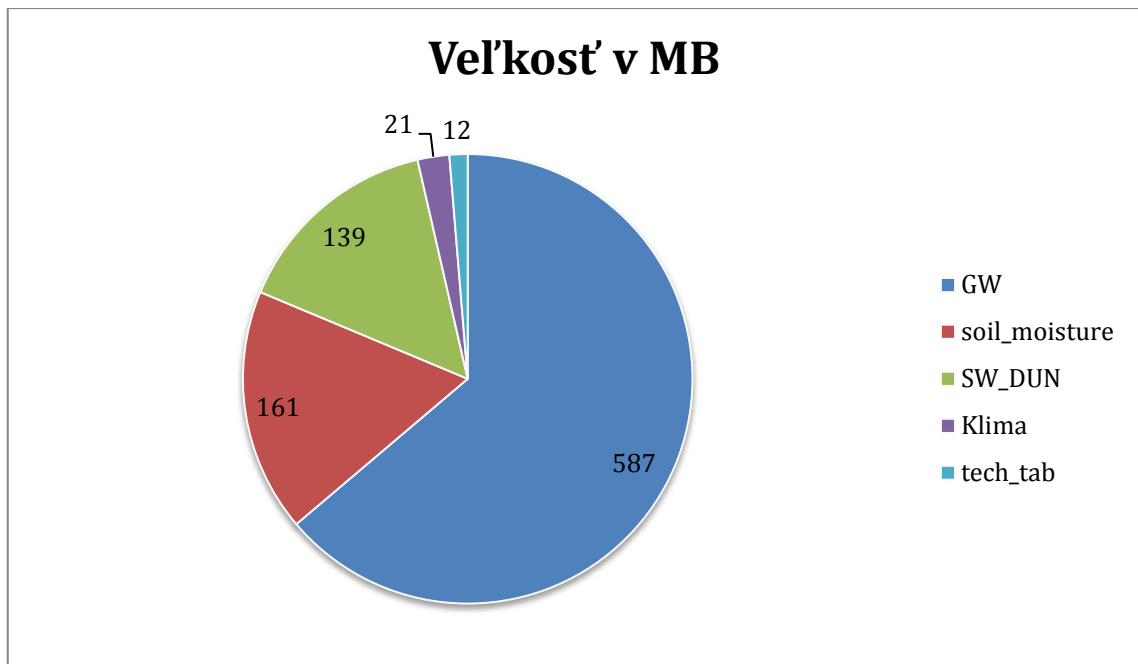
PK: zložený z ipt, dyt, depth  
FK: ipt z tabuľky OBJ  
mth z tabuľky DATA\_KOD

Druhá tabuľka **VLHKOST\_HP-ALL** obsahuje časové rady údajov o hĺbke hladiny podzemnej vody pre jednotlivé monitorovacie plochy, ktoré sú merané v pozorovacích objektoch na meranie hladiny vody.

PK: zložený z ipt a datum  
FK: ipt z tabuľky OBJ  
mth z tabuľky DATA\_KOD

### 2.4.6 Veľkosť databáz

Všetky spomínané databázy majú veľkosť nepresahujúcu 1 GB, pričom niektoré z nich majú len desiatky MB. Najväčšou databázou je databáza kvantity podzemných vôd (**GW**) o veľkosti 587 MB, za ňou nasledujú databázy pôdnej vlhkosti (**soil\_moisture**) a kvantity povrchových vôd (**SW\_DUN**) o veľkostiach 161 MB a 139 MB. Na predposlednom mieste je databáza klimatických údajov (**Klima**) zaberajúca 21 MB a za ňou ako najmenšia databáza objektov (**tech-tab**) s 12 MB. Súčet veľkostí databáz je 920 MB.



**Graf č. 1: Veľkosť databáz**  
(Zdroj: Vlastné spracovanie, 2012)

## 2.5 Identifikované problémy

V tejto časti zhrniem problémy a nedostatky, ktoré som po dôkladnej analýze databázového systému našiel.

V prvom rade by som spomenul hlavný problém, ktorým, sú aj podľa informácií od zamestnancov pomalé reakcie pri práci s databázami. Ako som už spomínal, toto bol hlavný dôvod, prečo bola databáza rozdelená na viacero čiastkových databáz, no ani toto riešenie daný problém úplne neodstránilo. Pomalá odozva sa ešte vystupňuje, ak sa s databázami snaží pracovať naraz viac užívateľov, kedy to často vedie aj k pádu aplikácie. Vzhľadom na to, že spoločnosť nemá veľký počet zamestnancov táto skutočnosť nemá až takú veľkú váhu, avšak situácie, kedy aspoň dvaja zamestnanci naraz pristupovali k dátam sa vyskytujú a tento fakt môže výrazne ovplyvniť efektivitu práce práve v čase, keď je potrebné niečo rýchlo dokončiť.

Ako ďalší problém by som videl samotnú štruktúru databázového systému, keďže tá bola prispôbená pre lepšiu prácu s dátami. To znamená, že keď si zamestnanec otvorí tabuľku, chce vidieť len potrebné údaje a najlepšie bez toho, aby sa musel v dátach horizontálne pohybovať. Preto nájdeme v databáze väzby medzi tabuľkami typu 1:1.

Databáza ďalej nespĺňa normálové formy a veľakrát nájdem väzby, ktoré nie sú definované alebo chýbajú. V databáze **tech-tab** (Príloha č. 1, Obr. 1) v tabuľke MON, som nenašiel definovaný primárny kľúč, čo je takisto chyba, keďže definícia primárneho kľúča je jedným zo základných pravidiel pri tvorbe databáz.

V tabuľkách sa nachádzajú aj zbytočné atribúty, ktoré mali pri tvorbe databáz istý účel, no pri neskoršej práci nemali uplatnenie, respektíve nebol čas sa nimi zaoberať. Takýmto atribútom je napríklad položka oprava v tabuľkách HED\_FIN a DIS\_FIN v databáze povrchových vôd (Príloha č. 1, Obr. 3). Pre prácu s dátami nie je tento atribút nevyhnutný, takže kvôli nedostatku času sa využíva len veľmi zriedka.

Databázy takisto obsahujú veľa tabuliek slúžiacich ako pomocné tabuľky pri importe údajov, ktorý zvyčajne prebieha manuálne a takisto ako pomocné tabuľky vytvorené zamestnancami spoločnosti, či už z dôvodu, aby mali zálohu, keby sa niečo nepodarilo pri importovaní alebo editovaní údajov v databáze, alebo s údajmi, ktoré dočasne potrebujú. Tieto tabuľky nemajú žiadne ďalšie využitie a nachádzajú sa tu len z nedôslednosti, pôvodne mali byť zmazané a z toho dôvodu som ich vyššie ani nespomínal.

Ďalším problémom je aj spomínané zálohovanie, ktoré nie je zautomatizované a prevádza sa manuálne v nepravidelných intervaloch, ktoré neraz presahujú mesiace. Ak by sa vyskytol nejaký problém vyžadujúci obnovu zo zálohy, je veľmi pravdepodobné, že by bolo znova nutné previesť import obrovského množstva dát, ktoré od poslednej zálohy pribudli. Takýto import dát by bol časovo veľmi náročný a vzniknutá situácia by mala negatívny dopad na plnenie úloh.

### 3 Návrh riešenia

Na základe predošlej analýzy súčasného stavu databázového systému firmy v tejto kapitole navrhнем ako vyriešiť niektoré problémy, ktoré boli v systéme identifikované. Samotná firma žiadala navrhnúť riešenie problému s celkovou rýchlosťou systému, čo je problém, ktorý sa asi najvýraznejšie prejavuje pri každodennej práci s databázami. Pri návrhu treba zohľadniť fakt, že ide o malú firmu, a preto musíme rátať s obmedzenými finančnými prostriedkami na investície do databázového systému. Ako vhodné riešenie sa javí jednoznačne implementácia databázového servera, ktorý by vyriešil hneď niekoľko problémov popísaných v predchádzajúcej kapitole.

Prínosy, ktoré by sa takýmto riešením dosiahli:

- zvýšenie rýchlosti reakcií systému pri práci s databázami
- možnosť pristupovať k databázam viacerým užívateľom súčasne, bez výrazného poklesu odozvy systému
- pokročilejšie funkcie zálohovania

Presun na databázový server však nie je všetko, treba pri ňom zohľadniť, do akej miery je potrebné pozmeniť súčasné riešenie, respektíve aké časti ponechať a čo všetko prepracovať. Realizácia niektorých činností môže zaberať značný čas, stáť nemalé peniaze alebo iné nefinančné prostriedky. Popíšem preto viacero možných riešení, z ktorých sa po konzultácií so zástupcami spoločnosti pokúsím na záver zrealizovať to, ktoré sa bude javiť ako najvhodnejšie.

Po preštudovaní problematiky databázového serveru a migrácie údajov som sa dopracoval k trom možným riešeniam:

1. Migrácia databáz z Microsoft Access na MS SQL Server
2. Migrácia databáz z Microsoft Access na MS SQL Server a implementácia webového rozhrania na prístup k databázam
3. Prerobenie štruktúry databáz v rozhraní MS SQL Server, tak aby spĺňali normálové formy a štandardy a implementácia webového rozhrania na prístup k databázam

### **3.1 Migrácia databáz z Microsoft Access na MS SQL Server**

Ako prvé uvádzam najjednoduchšie riešenie, ktoré by si vyžiadalo aj najmenej prostriedkov vynaložených firmou. Realizáciu tohto riešenia rozdelím do šiestich krokov:

1. Výber tabuliek a nutná úprava štruktúry
2. Inštalácia a konfigurácia SQL Servera
3. Prenesenie a úprava dát
4. Kontrola dát
5. Prepojenie s Access, doladovanie a testovanie
6. Skúšobná prevádzka

Po úspešnej realizácii dostaneme SQL Server, na ktorom budú uložené dáta spojené do jednej databázy. Access, ktorý je nainštalovaný na užívateľských počítačoch, bude naďalej slúžiť ako rozhranie prístupu k databázam. V Accesse by naďalej bola uložená celá logická časť aplikácie, formuláre, dotazy a reporty. Takýmto riešením získame výhody SQL Servera, akými sú hlavne vyššia rýchlosť a spoľahlivosť, bez toho, aby sa výraznejšie zmenilo prostredie pre používateľov. Užívatelia nebudú potrebovať špeciálne zaškolenie a bude im stačiť krátka inštrukcia, ak dôjde k nejakým minimálnym zmenám.

#### **3.1.1 Časová náročnosť**

Veľmi dôležitá je voľba vhodného termínu, pretože charakter vykonávanej práce v spoločnosti je cyklický, čo znamená, že každý rok sú určité obdobia, keď je práce výrazne viac a dokončujú sa rôzne správy. Na druhej strane nám to dáva príležitosť využiť na implementáciu obdobie, keď je práce menej. Na to, aby sme vedeli vybrať vhodný termín potrebujeme vedieť, koľko času implementácia zaberie. Po konzultácii a premyslení si krokov, by som implementáciu tohto riešenia pravdepodobne odhadoval na 2 až maximálne 3 týždne, ak zohľadním aj dostatočnú časovú rezervu. Táto možnosť by teda mohla byť realizovaná z vlastných zdrojov firmy, pretože jeden zo zamestnancov je technicky dostatočne kvalifikovaný na realizáciu spomenutých krokov. Takisto by sa počas obdobia s menšou vyťaženosťou mohol plne venovať

implementácii, pretože by bolo možné vyčleniť dostatočný čas, tak aby sa nemusel podieľať na iných úlohách.

### **3.1.2 Finančná náročnosť**

Náklady na projekt pozostávajú z nákupu hardvéru, softvéru a nákladov na odpracované hodiny. Cenu hardvéru odhadujem na 800 EUR. Zakúpenie licencií na softvér by vyšlo do 500 EUR. Najvýraznejšou položkou budú náklady na pracovnú silu. Firma má pracovný týždeň 42,5 hodín, čo znamená, že počet hodín, ktorý by zamestnanci strávili na implementácii projektu by bol od 480 do 700. Tento počet sa môže líšiť v závislosti na dĺžke trvania a takisto od počtu zamestnancov pracujúcich na projekte. Priemerná hodinová mzda je 20 EUR, z čoho náklady na pracovnú silu vychádzajú na 9600 až 14 000 EUR.

### **3.2 Migrácia databáz z Microsoft Access na MS SQL Server a implementácia webového rozhrania**

Ďalšie navrhované riešenie, ku ktorému som dospel po konzultáciách s firmou je časovo i finančne náročnejšie. Realizáciu tohto riešenia je možné zhrnúť do nasledovných desiatich krokov:

1. Konzultácie s externistami
2. Výber tabuliek a nutná úprava štruktúry
3. Inštalácia a konfigurácia SQL Servera
4. Prenesenia a úprava dát
5. Kontrola dát
6. Vývoj webového rozhrania
7. Implementácia rozhrania
8. Testovania a doladovanie
9. Skúšobná prevádzka
10. Zaškolenie zamestnancov

Postup pri implementácii tohto riešenia má sčasti spoločné kroky s prvým návrhom. Dáta budú, tak ako predtým presunuté na SQL Server, čím sa práca s nimi zrýchli a takisto sa zvýši spoľahlivosť, pričom štruktúra databáz sa výrazne nezmení. Veľkou

zmenou prejde prístupovanie k dátam. To už nebude, tak ako doteraz realizované cez program Access, respektíve jeho súčasti, ale cez webové rozhranie, ku ktorému užívatelia budú potrebovať iba internetový prehliadač. Takáto zmena má veľkú výhodu v tom, že nie je potrebné mať žiadny špeciálny program na prístup k databázam a ten je teda možný z akéhokoľvek počítača, ktorý má prístup k serveru. Na druhej strane zmena rozhrania znamená, že zamestnanci budú musieť byť oboznámení s novým rozhraním a tým pádom je nutné odborné školenie. Ďalšou výhodou vynechania Accessu je ešte vyššia rýchlosť pri práci s databázami.

### **3.2.1 Časová náročnosť**

Problematiku výberu vhodného termínu som rozobral už pri predošlej alternatíve riešenia, preto sa ňou tak podrobne už zaoberať nebudem. V tomto riešení je veľmi podobná a odlišuje sa hlavne v tom, že realizáciu bude potrebné zadať externistom, pretože firma nemá dostatočne kvalifikovaných zamestnancov, ktorí by boli schopní implementovať toto riešenie. Dĺžka trvania tejto realizácie navyše presahuje dobu, po ktorú by sa technicky kvalifikovaní zamestnanci mohli plne venovať tejto činnosti. Dĺžku trvania implementácie, ktorá bude vo veľkej miere závisieť aj od šikovnosti a skúseností zodpovedných pracovníkov, odhadujem na 4 až 6 pracovných týždňov. Avšak riziko neplánovaného predĺženia ovplyvňuje viacero faktorov, ktorými sú spomínaní externí pracovníci, ale aj väčšie množstvo realizovaných krokov.

### **3.2.2 Finančná náročnosť**

Ako som už spomínal, predpokladané náklady na túto možnosť budú vyššie ako pri predošlom riešení. Náklady na hardvér a softvér zostanú pravdepodobne nezmenené na úrovni 800 a 500 EUR, no výrazne sa zvýšia náklady na pracovnú silu, hlavne kvôli dlhšiemu obdobiu implementácie projektu. Cenu odpracovaných hodín vlastnými zamestnancami firmy a externými pracovníkmi vyčíslim zvlášť, keďže sa hodinová sadzba líši. Zamestnanci firmy majú hodinovú sadzbu na úrovni 20 EUR a v tomto prípade nebude nutné, aby projektu venovali toľko času ako v prvom riešení. Pravdepodobný počet hodín investovaných do projektu bude tvoriť úvodná konzultácia a prípadné konzultácie pri implementácii riešenia. Spolu by to predstavovalo asi 200 hodín, z čoho dostaneme náklady na úrovni 4000 EUR.

Na základe prieskumu trhu som prišiel na odhadovanú hodinovú sadzbu externých pracovníkov vo výške 15 EUR. Počet odpracovaných hodín by bol od 800 až do 1200, z čoho dostaneme náklady 12000 až 18000 EUR. Celkové náklady na pracovnú silu by v takomto prípade boli 16000 až 22000 EUR.

### **3.3 Prerobenie štruktúry databáz v rozhraní MS SQL Server a implementácia webového rozhrania na prístup k databázam**

Tretie riešenie je najobsiahlejšie a takisto z časového aj finančného hľadiska najnáročnejšie. Realizácia by prebiehala v nasledovných dvanástich krokoch:

1. Konzultácie s externistami
2. Návrh novej štruktúry databázy
3. Inštalácia a konfigurácia SQL Servera
4. Implementácia databázy
5. Spísanie dokumentácie
6. Import dát
7. Kontrola dát
8. Vývoj webového rozhrania
9. Implementácia rozhrania
10. Testovanie a doladovanie
11. Skúšobná prevádzka
12. Zaškolenie zamestnancov

Pri tomto riešení nenájdeme už žiadne kroky spoločné s prvým riešením, no spoločná časť s druhým návrhom je práve vytvorenie webového rozhrania. Toto riešenie je najkomplexnejšie a vyžaduje v podstate nanovo vytvoriť databázový systém firmy, ktorý bude mať s pôvodným spoločné viac-menej iba uložené údaje. Externí odborníci by navrhli na základe analýzy procesov a dátového toku kompletne novú štruktúru databázy, ktorá by spĺňala normálové formy a zaužívané štandardy. Výsledok tohto riešenia bude veľmi podobný tomu, ktorý dostaneme pri druhom návrhu, avšak lepšie optimalizované databázy by mohli byť nielen prehľadnejšie a zaberat' menej miesta, ale pravdepodobne by sa zvýšila aj rýchlosť práce s nimi. Databáza by bola uložená na SQL Serveri a užívatelia by k dátam pristupovali cez webové rozhranie.

### **3.3.1 Časová náročnosť**

Spoločnou črtou tretieho a druhého návrhu riešenia je riešenie externým odborníkom, z čoho vyplýva menšie pracovné zaťaženie zamestnancov spoločnosti, ktorí budú zodpovední najmä za konzultácie a prípadné pripomienky k návrhom riešení. Vzhľadom na obsiahlosť projektu odhadujem dobu trvania na 8 až 12 pracovných týždňov. Veľa bude závisieť od toho, ako rýchlo pochopia externí pracovníci problematiku a od efektivity ich práce. Riziko výskytu nepredvídateľných problémov je vyššie, pretože je potrebné vykonať väčší počet rôznych úkonov.

### **3.3.2 Finančná náročnosť**

Náklady na hardvér a softvér sa pri tomto riešení takisto nezmenia a ostanú na úrovni 800 a 500 EUR. Najdlhšia doba realizácie zo všetkých troch možností výrazne ovplyvní náklady na pracovnú silu. Zamestnanci firmy zodpovední za konzultovanie a kontrolu priebehu projektu by mohli takto stráviť 250 až 400 pracovných hodín, čo pri hodinovej sadzbe 20 EUR predstavuje náklady vo výške 5000 až 8000 EUR. Hodinová sadzba externých pracovníkov by bola 15 EUR, tak ako v predošlom prípade. Pri 1600 až 2400 odpracovaných hodinách by sa čiastka vyšplhala na 24000 až 36000 EUR. Celkové náklady na pracovnú silu by predstavovali 29000 až 44000 EUR.

## **3.4 Zhrnutie**

V predchádzajúcich kapitolách som načrtol tri možnosti riešenia, z ktorých každá sa líši nielen po finančnej stránke, ale aj úrovňou zmien databázového systému firmy. V nasledujúcom texte stručne zhrniem výhody a nevýhody jednotlivých riešení.

Prvé riešenie, v ktorom prenesieme súčasné databázy na SQL Server a naďalej budeme k dátam pristupovať cez Access je časovo nenáročné a finančne najpriateľnejšie. Pomerne jednoduchou úpravou získame výhody SQL Servera akými sú rýchlosť, stabilita a takisto aj pokročilé zálohovacie funkcie. Ďalšou výhodou je možnosť použitia súčasného prostredia vytvoreného v prostredí Access, na ktoré sú používatelia zvyknutí. Avšak prístup cez program Access je zároveň aj jednou z nevýhod daného riešenia, pretože táto skutočnosť môže brzdiť výkon celého systému a stabilita naprogramovaných súčastí je tiež otázna.

Výsledkom časovo aj finančne náročnejšieho druhého riešenia bude SQL Server, na ktorom budú prenesené súčasné databázy, v takmer nezmenenom stave, a navyše aj webové rozhranie na prístup k dátam, ktoré takto nahradí program Access. Výhodami takéhoto riešenia sú už spomínané lepšie vlastnosti SQL Servera (rýchlosť, stabilita, pokročilé zálohovanie). Vďaka použitiu webového rozhrania bude celý takto vytvorený systém rýchly a stabilný. Webové rozhranie poskytuje väčšiu flexibilitu a nie je potreba inštalovať žiadny špecializovaný software, pretože na klientskom počítači stačí prehliadač. Nevýhodami tohto riešenia sú najmä vyššia časová a finančná náročnosť a takisto so zmenou prostredia prichádza na rad aj zaškolenie zamestnancov, ktorí si naň budú musieť chvíľu zvykať.

Tretie riešenie je najkomplexnejšie a zároveň aj najdrahšie a s najdlhšou realizáciou. Ako jediné zahŕňa aj návrh novej štruktúry databáz a jej naprogramovanie v prostredí SQL Servera. Inak sa viac-menej zhoduje s druhým riešením a teda výsledkom budú databázy na SQL Serveri a webové rozhranie bude zabezpečovať prístup k nim. Najväčšími výhodami bude rýchlosť a stabilita takéhoto riešenia, keďže všetko sa dá zladať tak, aby to fungovalo bez kompromisov, no otázne je nakoľko bude takýto prírastok rýchlosti pozorovateľný, respektíve reálne využiteľný. Nová štruktúra databázy by reálne odstránila problémy s nespĺňaním normálových foriem a odborových štandardov, avšak táto úprava je z hľadiska koncového užívateľa iba „kozmetická“. Výhody SQL Servera a webového rozhrania, ktoré som spomenul pri druhom riešení platia aj tu. Na druhú stranu asi najväčšou nevýhodou tohto riešenia je dĺžka implementácie a cena. Pre takto malú firmu pravdepodobne nebude rentabilné investovať do projektu odhadovanú sumu.

**Tabuľka č. 1: Porovnanie navrhovaných riešení**

	Výhody	Nevýhody
<b>1. riešenie</b>	<ul style="list-style-type: none"> <li>- relatívne nízke náklady</li> <li>- krátka implementácia</li> <li>- výhody SQL Servera</li> <li>- známe užívateľské rozhranie</li> </ul>	<ul style="list-style-type: none"> <li>- použitie Access</li> <li>- nevyužitý celý potenciál SQL Servera</li> </ul>
	<b>Cena: 10900-15300 EUR</b>	<b>Trvanie: 2-3 týždne</b>

<b>2. riešenie</b>	<ul style="list-style-type: none"> <li>- výhody SQL Servera</li> <li>- webové rozhranie - prístup cez webový prehliadač</li> <li>- rýchlosť a stabilita</li> </ul>	<ul style="list-style-type: none"> <li>- vyššie náklady</li> <li>- dlhšia implementácia</li> <li>- otázky reálny prínos vyššej rýchlosti</li> <li>- nutnosť oboznámiť sa s novým prostredím</li> </ul>
	<b>Cena: 17300-23300 EUR</b>	<b>Trvanie: 4-6 týždňov</b>
<b>3. riešenie</b>	<ul style="list-style-type: none"> <li>- výhody SQL Servera</li> <li>- maximálna rýchlosť a stabilita vďaka vyladenosti</li> <li>- webové rozhranie - prístup cez webový prehliadač</li> <li>- spĺňanie normálových foriem a štandardov</li> </ul>	<ul style="list-style-type: none"> <li>- veľmi vysoké náklady</li> <li>- dlhá implementácia</li> <li>- otázky reálny prínos pre koncového užívateľa</li> <li>- nutnosť oboznámiť sa s novým prostredím</li> </ul>
	<b>Cena: 30300-45300 EUR</b>	<b>Trvanie: 8-12 týždňov</b>

(Zdroj: Vlastné spracovanie, 2013)

### 3.5 Výber vhodnej varianty

Možné riešenia som následne konzultoval vo firme. Po zvážení výhod, nevýhod a finančnej stránky sa zástupcovia firmy priklonili k prvému, najlacnejšiemu riešeniu, ktoré by prinieslo asi najväčší úžitok a tým pádom by rentabilita projektu bola vysoká. Toto riešenie určite neprinesie najväčší nárast rýchlosti a z technického hľadiska možno nie je najoptimálnejšie, ale za vynaložené finančné prostriedky ponúkne firme najväčší reálny prínos. V nasledujúcich podkapitolách podrobnejšie rozpišem jednotlivé kroky tejto varianty.

#### 3.5.1 Výber tabuliek

V prvom kroku je potrebné vybrať tabuľky s dátami, ktoré chceme na databázový server presunúť. Tento krok je dôležité konzultovať s pracovníkmi spoločnosti. Bolo by vhodné vybrať iba tie tabuľky, s ktorými sa reálne pracuje a eliminovať presun zbytočných, ktoré, ako som už spomínal, vznikli z nedôslednosti alebo sa nepoužívajú z iných dôvodov. Pri tejto príležitosti by som odporúčal vypracovať aj návrh na

čiastočné upravenie štruktúry databáz, ktorý by zahŕňal opravu časovo menej náročných chýb nespĺňajúcich normalizačné formy alebo iné štandardy. Jednalo by sa hlavne o definovanie primárnych kľúčov, odstránenie väzieb 1:1, ale aj zmazanie nepoužívaných atribútov. Časovo odhadujem, že by bolo možné tento krok zrealizovať za 2 až 4 dni, pričom by na jeho realizácii pracovali dvaja zamestnanci.

### **3.5.2 Inštalácia SQL Servera**

Po výbere dát a návrhu na zmeny by nasledovala inštalácia databázového servera. Tento krok zahŕňa spojzdenie a fyzické zapojenie počítača do podnikovej siete, inštaláciu operačného systému, MS SQL Servera a jeho konfiguráciu tak, aby spolupracoval s ostatnými zariadeniami. Pre technicky kvalifikovaného zamestnanca by realizácia tohto kroku mohla predstavovať 1 deň.

### **3.5.3 Prenesenie a úprava dát**

Po výbere tabuliek a implementácii SQL Servera nasleduje ďalší krok, v ktorom by sa preniesli dáta z Accessu na SQL Server. Dáta sa nachádzajú v súboroch formátu \*.mdb, čo je formát, ktorý používali staršie verzie Accessu. Po tom, ako som si túto problematiku naštudoval, som zistil, že samotný Microsoft poskytuje na riešenie migrácie dát dva nástroje.

Prvým z nástrojov je zabudovaný Upsizing Wizard, ktorý je priamo súčasťou Accessu a druhým je samostatný program SQL Server Migration Assistant (SSMA), voľne dostupný na webových stránkach spoločnosti Microsoft. Na základe dohľadných skutočností a vyjasnení rozdielov medzi týmito nástrojmi by som ako vhodné riešenie odporúčal použiť Upsizing Wizard. Ten, na rozdiel od SSMA, nielenže presunie dáta na SQL Server, ale aj vytvorí väzby z Accessu na tabuľky presunuté na server. Použitie Upsizing Wizard napriek čiastočnej automatizácii odvedie asi len 40% práce (Haught, 2013). Použitie a celkový prínos Upsizing Wizard sa líši v závislosti od dát, no určite si bude vyžadovať asistenciu programátora, ktorý bude riešiť prevod dát, na ktoré kapacita automatického nástroja nestačí. Časté problémy, s ktorými sa pri prevode môžeme stretnúť a nie sú riešené automaticky sú zhrnuté do nasledujúcej tabuľky (**Tabuľka č. 2**).

**Tabuľka č. 2: Problémy pri prenášaní dát**

<b>Problém</b>	<b>Popis</b>
<b>Neštandardné mená tabuliek/polí</b>	Jet engine a SQL používajú rozdielne štandardy pri názvoch. Upsizing Wizard niektoré nájde, no nie všetky, a tieto potom nebudú fungovať v existujúcom kóde.
<b>Rozdiely v jazyku SQL</b>	Access/Jet používajú vlastný dialekt SQL jazyka, ktorý sa líši od ANSI SQL podporovaného SQL Serverom. Veľa dotazov postavených na Jet engine nemôže byť bez prepísania na SQL Serveri spustených.
<b>Problémy pri prevode dátového typu</b>	Access/Jet majú vlastné štandardy pre dátové typy, ktoré sa v niektorých prípadoch líšia od SQL Servera. Upsizing wizard môže pri konvertovaní dátových typov navrhnúť nejaké možnosti, ale všetky zmeny si vyžadujú kontrolu.
<b>Rozdielne architektúry</b>	Microsoft Upsizing Wizard nedokáže prepísať aplikáciu, tak aby riadne pracovala s klient/server modelom SQL Servera. Takmer všetky aplikácie pre Access/Jet sú dizajnované na prácu so súborovým modelom Jet engine. Takýto dizajn sa dobre neprispôsobí klient/server modelu, následkom čoho môže byť slabý výkon.
<b>Kód sa nekonvertuje</b>	Upsizing Wizard neprevádza žiadne časti aplikácií naprogramovaných vo VBA. Vo výsledku tak dostaneme chyby zapríčinené tým, že časti aplikácie odkazujú na SQL Server, no náš kód stále odkazuje na Access/Jet databázu.
<b>Položky, ktoré sa neprenášajú</b>	Microsoft Upsizing Wizard nekonvertuje žiadne s nasledujúcich objektov: skryté objekty, bezpečnostné nastavenia, Format a InputMask vlastnosti, vlastnosti záhlavia tabuliek/polí, makrá, moduly a iné

(Zdroj: Haught, 2003)

Dôležité rozhodnutie, ktoré treba v tomto kroku urobiť je akým spôsobom bude Access prepojený s SQL Serverom. Access používa vlastný mechanizmus uskladnenia dát nazvaný Microsoft Jet Database Engine. Pri presunutí dát máme dve možnosti, buď budeme pristupovať k serveru naďalej pomocou Jet engine alebo priamo na SQL Server. Prvá možnosť je tá jednoduchšia, pretože formuláre, reporty, makrá a celá logická časť ostávajú uložené v súčasnej databáze. Takže všetky doteraz naprogramované objekty budú naďalej fungovať bez nutnosti do nich zasahovať alebo len s minimálnou úpravou kódu. Avšak všetko bude naďalej bežať cez Jet engine, ktorý bude mať na starosti preloženie všetkých dotazov na SQL Server z natívneho formátu do príkazov kompatibilných s SQL. Tento medzikrok má nepriaznivý vplyv na výkon celého systému, ale nepredpokladám, že by veľmi spomalil prístup k dátam. Keďže spoločnosť má veľké množstvo doprogramovaných súčastí, navrhujem zostať pri prvom variante, tak aby sa tieto súčasti nemuseli programovať nanovo.

#### **3.5.4 Kontrola dát**

Po tom ako sme dáta úspešne presunuli na SQL Server je nutné ich dodatočne skontrolovať, pretože kvalita dát je jedným z najdôležitejších faktorov správnej databázy. Je nutné overiť, či sme preniesli všetky tabuľky, ktoré sme mali v pláne. V ďalšom kroku kontroly by sme sa mali preveriť správnosť prevodu dátových typov (najmä pri dátumoch), s ktorými, ako už bolo spomenuté v tabuľke (**Tabuľka. č. 2**), majú softvérové nástroje časté problémy. V neposlednom rade treba skontrolovať aj úplnosť a správnosť väzieb medzi tabuľkami. V tomto prípade môžeme porovnávať schémy novovytvorenej a starej databázy, ktoré prehľadne graficky zobrazia existujúce väzby.

#### **3.5.5 Prepojenie s Access, dolad'ovanie a testovanie**

Pokiaľ sme overili správnosť dát postupujeme ďalším krokom, v ktorom vytvoríme prepojenie Accessu s SQL Serverom, modifikujeme kód a testujeme. V prípade, že použijeme vstavaný Upsizing Wizard ušetríme prácu s prepojovaním, pretože to tento nástroj pri prenášaní dát urobí automaticky za nás. Ak sme použili samostatne dostupný SSMA program, bude treba pripojenie na SQL Server nakonfigurovať, s tým že SQL

Server bude slúžiť ako back-end, na ktorom budú uložené dáta a užívatelia k nim budú pristupovať cez tzv. front-end, ktorým bude v našom prípade Access.

Ako už bolo spomenuté v **Tabuľke č. 2** užívateľsky naprogramované súčasti databázy ako sú formuláre, makrá a reporty treba po prenesení dát mierne modifikovať, tak aby pracovali s dátami na SQL Serveri. Po modifikovaní kódu a doprogramovaní niektorých súčastí treba pripravenú aplikáciu otestovať, či všetky súčasti fungujú, tak ako bolo zamýšľané. Ak sa vyskytnú nejaké chyby, treba nájsť časť kódu, ktorú treba upraviť a program vyladiť.

V tomto kroku sa prevedie aj nastavenie automatického zálohovania. Prvé zálohovanie bude prebiehať formou kompletnej kópie databázy a pri ďalších zálohách bude používaná diferenciálna metóda zálohy. Pri tejto metóde sa prenesú iba zmeny, ktoré sa udiali od poslednej zálohy.

### **3.5.6 Skúšobná prevádzka**

Posledným krokom pri implementácii je skúšobná prevádzka, v ktorej sa bude simulovať klasická plná prevádzka. Zamestnanci dostanú za úlohu robiť bežné úkony s databázou a ak náhodou vzniknú nejaké problémy budú to hlásiť developerovi, ktorý následne problémy odstráni. Nemenej dôležitou úlohou bude zaškolenie zamestnancov pri práci s databázami, ktoré vzhľadom na to, že zmeny budú len minimálne, ak vôbec nejaké, môže prebiehať v rámci skúšobnej prevádzky.

## **4 Praktická realizácia**

Po zhrnutí a rozpísaní jednotlivých krokov potrebných na realizáciu som sa potom pokúsil o realizáciu niektorých krokov potrebných pre implementáciu. Skúšobnú realizáciu som uskutočnil na osobnom počítači nie priamo v prostredí firmy, ale som mal k dispozícii kópiu databáz. V nasledujúcej kapitole popisujem výsledky, ku ktorým som sa dopracoval. Realizoval som prenos všetkých piatich analyzovaných databáz.

### **4.1 Výber tabuliek**

V prvom kroku som si prešiel databázy a na základe predošlej analýzy som po konzultácii vybral tabuľky, ktoré firma používa a plánuje preniesť na SQL Server. Do štruktúry databáz som však nezasahoval, keďže tento krok by musel podniknúť niektorý zo zamestnancov.

### **4.2 Inštalácia SQL Servera**

Na začiatku realizácie druhého kroku bolo nutné najprv stiahnuť samotný SQL Server, ktorý je dostupný v edícii Express zadarmo. Táto edícia má síce menej funkcií a viaceré obmedzenia, čo však pri tejto skúšobnej realizácii nie je limitujúce. Na základe zverejnených informácií by som túto verziu odporučil použiť aj pri samotnej realizácii vo firme, pretože v súčasnej situácii firma plnú funkcionality hlavných edícií ani nevyužije. Navyše obmedzenia, ktoré sa týkajú maximálnej veľkosti databázy 10 GB a využitia iba jedného procesora, nie sú v našom prípade limitujúce, pretože veľkosť popisovaných databáz nepresahuje ani 1 GB.

Pri inštalácii som postupoval jednotlivými krokmi inštalačného sprievodcu, kde bolo nutné spraviť základnú konfiguráciu SQL Servera, ktorá by sa však v podmienkach firmy odlišovala. Inštalácia prebehla bezproblémovo a už po asi pol hodine som mal vytvorený SQL Server s prednastaveným menom MSSQLSERVER, pripraveným na použitie. Na prihlasovanie na SQL server som použil Windows Authentication. V prípade skutočnej realizácie by bolo nutné stráviť viac času s pokročilejšou konfiguráciou, ale ako som sa už zmienil touto problematikou som sa pri mojom pokuse nezaoberal.

### 4.3 Prenesenie a úprava dát

Prenesenie dát som sa rozhodol realizovať cez nástroj Upsizing Wizard, ktorý je súčasťou Accessu. V Accesse som si otvoril ako prvú databázu **tech-tab** a v hlavnom menu som v záložke Database Tools, v časti Move Data spustil položkou SQL Server spomínaný nástroj. Po prvých dvoch krokoch, keď sa overovala dostupnosť zadaného serveru, som sa stretol s prvým problémom. Verzia Accessu 2007, ktorý som mal nainštalovaný nebola kompatibilná s najnovším SQL Serverom 2012. Keďže firma má ešte staršiu verziu kancelárskeho balíka MS Office 2003, znamená to, že by museli investovať dodatočné finančné prostriedky na zakúpenie licencií Access 2010. Licencie na Access by predstavovali sumu asi 600 EUR.

Po nainštalovaní Accessu 2010 som sa znova pokúsil o prenesenie dát databázy **tech-tab**. Tento krát prebehlo overenie dostupnosti Serveru v poriadku a mohol som pokračovať v ďalších krokoch sprievodcu. V ďalšom kroku sa mi zobrazila ponuka na výber tabuliek, ktoré chcem z existujúcej databázy preniesť do novej na SQL Serveri. Po vybratí tabuliek na základe predchádzajúcej analýzy som v ďalších krokoch nastavil aké vlastnosti tabuliek sa majú preniesť a zvolil som možnosť automaticky pripojiť tabuľky hneď po prenesení na SQL Server, tak aby sa s nimi dalo naďalej pracovať z Accessu.

Nasledovalo prenášanie tabuliek, ktoré malo trvanie v rádoch minút. Po skončení prenášania Access vygeneroval správu, v ktorej špecifikoval, ktoré tabuľky a atribúty boli prenesené a takisto aj chyby, ktoré sa pri prenose vyskytli (**Príloha č. 2**). Prvou chybou, ktorá bola v reporte bol chýbajúci primárny kľúč v tabuľke MON, ďalšia chyba sa týkala rozdielne definovaných typov pri cudzích kľúčoch. Táto chyba sa vyskytla pri troch tabuľkách a to DATA\_KOD, kde sa jednalo o atribút org\_id, XY a Z, pri atribúte mth\_id. Poslednou chybou, ktorá bola zaznamenaná v reporte boli prázdne políčka pri atribúte, u ktorého nebolo povolené nechať prázdny záznam, táto chyba bola nájdená v tabuľke OBJ\_SUB pri atribúte nam. Napriek chybám bol prenos dát do novej databázy úspešný, avšak nepreniesli sa vzťahy, v ktorých sa chyby vyskytli, respektíve sa upravili vlastnosti tabuliek. Access automaticky pripojil novovytvorené tabuľky z SQL Servera a navyše ponechal aj pôvodné tabuľky, ktorých meno bolo rozšírené

o príponu „local“. Tieto tabuľky sa môžu po kontrole dát zmazať, aby zbytočne neplietli užívateľov a nezaberali miesto na disku.

Ako som už spomínal pri podrobnom popise riešenia v predchádzajúcej kapitole, v prvom kroku by sa zásahy do štruktúry mali týkať práve správneho nadefinovania kľúčov, ktorých následkom je neúplné prenesenie vzťahov v databáze.

Pri prenášaní ostatných databáz som postupoval trochu inak. Keďže databáza na SQL Serveri už bola vytvorená a nemá zmysel držať tabuľky oddelené vo viacerých databázach, na začiatku sprievodcu prenášaním som zvolil, aby sa tabuľky prenášali do už existujúcej databázy. Pri prenose týchto databáz sa nevyskytli už žiadne chyby. Jedinou skutočnosťou, ktorú je potrebné spomenúť bolo, že kvôli nalinkovaným tabuľkám z databázy **tech-tab** sa nepreniesli žiadne väzby (*pozn. tabuľky z ostatných databáz mali väzby jedine s tabuľkami z databázy tech-tab, väzby medzi vlastnými tabuľkami nie sú*). Tieto väzby bolo potrebné po prenose dodefinovať priamo v prostredí SQL Serveru.

Na vytvorenie väzby je potrebné pridať cudzí kľúč, no v niektorých prípadoch je potrebné zmeniť dátový typ atribútu, pretože sa nezhodujú. Takýto prípad sa mi vyskytol v tabuľke ORG, ktorú som chcel prepojiť s tabuľkou DATA\_KOD. Ako cudzí kľúč v tabuľke DATA\_KOD slúži atribút org\_id. Pri tomto atribúte som najprv zmenil jeho dĺžku z pôvodných 10 znakov na 8, tak ako to bolo definované v tabuľke ORG. SQL kód, ktorým som želanú zmenu dosiahol je uvedený na **Obrázku č. 2**.

```
ALTER TABLE dbo.data_kod
ALTER COLUMN org_id nvarchar(8);

ALTER TABLE dbo.data_kod ADD CONSTRAINT FK_org
FOREIGN KEY (org_id) REFERENCES dbo.org (org_id);
```

**Obrázok č. 2: Príklad kódu na pridanie cudzieho kľúča**  
(Zdroj: Vlastné spracovanie, 2013)

## Záver

Hlavným cieľom bakalárskej práce je zefektívnenie databázového systému malej obchodnej spoločnosti. K tomuto cieľu bolo potrebné vykonať analýzu spoločnosti, v ktorej bolo zmienené fungovanie firmy, jej technologická infraštruktúra, ale hlavne detailne popísaný súčasný stav databázového systému. Na konci analýzy boli uvedené identifikované problémy, ktoré je potrebné pre zefektívnenie databázového systému vyriešiť. Jednalo sa hlavne o slabý výkon systému, nestabilitu aplikácie a nedostatočnú úroveň zálohovania dát.

V ďalšej kapitole som vypracoval návrh riešenia identifikovaných problémov. Boli vypracované tri varianty riešenia, z ktorých každý riešil vyššie spomínané problémy, pričom sa líšili rozsahom zmien potrebných na zefektívnenie databázového systému. Od rozsahu zmien sú priamo závislé aj doba trvania realizácie a výška finančných prostriedkov, ktoré treba vynaložiť. Spoločnou črtou všetkých variantov bolo prenesenie dát na SQL Server, ktorého pokročilé funkcie a výkon vo veľkej miere rieši spomínané problémy. Jednotlivé varianty riešenia, ich výhody a nevýhody, finančnú a časovú náročnosť, som podrobne popísal v tretej kapitole. Výsledky som prehľadne zobrazil v sumarizujúcej tabuľke.

V prvom riešení by sa ponechal prístup k databázam cez program Access, čo by umožnilo naďalej používať existujúce naprogramované prostredie, pričom databázy by boli uložené na SQL Serveri. Toto riešenie je najlacnejšie a najmenej časovo náročné.

Druhé riešenie by kompletne zmenilo prístup k databázam, ktorý by bol zabezpečený webovou aplikáciou. Čas potrebný na naprogramovanie aplikácie by výrazne predĺžil celú realizáciu a vyššia by bola aj finančná náročnosť takéhoto riešenia.

Tretie riešenie by bolo najkomplexnejšie a jednalo by sa o úplné preprogramovanie databázového systému, vrátane jeho štruktúry. Tento fakt by sa výrazne odrazil na finančnej a časovej náročnosti.

Po konzultácií so zástupcami spoločnosti bolo vybrané prvé, najmenej náročné riešenie. Jednotlivé kroky tohto variantu som v mojej práci podrobne rozpísal a analyzoval som

možné problémy, s ktorými treba pri implementácii vopred rátať. V poslednom kroku som popísal aj praktickú realizáciu vybraných krokov.

Na záver konštatujem, že vytýčený cieľ bakalárskej práce som splnil. Po detailnej analýze databázového systému spoločnosti som navrhol variantné riešenia, z ktorých som po konzultácii so zástupcami spoločnosti a zohľadnení časového a finančného hľadiska prakticky realizoval najjednoduchší variant.

## Zoznam použitej literatúry

CONNOLLY, Thomas M. and Carolyn E. BEGG, 2005. Database systems: A Practical Approach to Design, Implementation, and Management. 4. vydanie. Harlow: Pearson Education. ISBN 0-321-21025-5.

DEWSON, Robin, 2012. Beginning SQL Server 2012 for Developers. 3. vydanie. New York: Apress Media. ISBN 978-1-4302-3750-1.

GROH, Michael R., Joseph C. STOCKMAN et al, 2007. Access 2007 Bible. Indianapolis: Wiley Publishing. ISBN 0-470-04673-2.

HAUGHT, Dan. Strategic Initiatives: Evolving Microsoft Access Applications to Microsoft SQL Server. FMS Professional Solutions Group [online]. 2013 [cit. 2013-04-28]. Dostupné z:

<http://www.fmsinc.com/FMSUpsize/docs/EvolvingMicrosoftAccessApplications.pdf>

HOFFER, J. A., V. RAMESH a H. TOPI, 2010. Modern database management. 10. vydanie. Upper Saddle River: Prentice Hall. ISBN 0-13-608839-2.

KŘÍŽ, Jiří a Petr DOSTÁL, 2005. *Databázové systémy*. Brno: Akademické nakladatelství CERM. ISBN 80-214-3064-8.

MICROSOFT. Introduction to forms. Microsoft.com [online]. 2013a [cit. 2013-04-20]. Dostupné z: <http://office.microsoft.com/en-001/access-help/introduction-to-forms-HA010343724.aspx>

MICROSOFT. Create a simple report. Microsoft.com [online]. 2013b [cit. 2013-04-20]. Dostupné z: <http://office.microsoft.com/en-001/access-help/create-a-simple-report-HA001230739.aspx>

MISTRY, Ross and Stacia MISNER. Introducing Microsoft SQL Server 2012. Redmond: Microsoft Press [online]. ISBN 978-0-7356-6515-6. 2012 [cit. 2013-05-02].

Dostupné tiež z:

[http://download.microsoft.com/download/F/F/6/FF62CAE0-CE38-4228-9025-FBF729312698/Microsoft\\_Press\\_eBook\\_Introducing\\_Microsoft\\_SQL\\_Server\\_2012\\_PDF.pdf](http://download.microsoft.com/download/F/F/6/FF62CAE0-CE38-4228-9025-FBF729312698/Microsoft_Press_eBook_Introducing_Microsoft_SQL_Server_2012_PDF.pdf)

SHARMA, Neeraj, Liviu PERNIU et al. Database fundamentals. 1. vydanie. Markham: IBM [online]. 2010 [cit. 2012-12-03]. Dostupné tiež z:  
[http://public.dhe.ibm.com/software/dw/db2/express-c/wiki/Database\\_fundamentals.pdf](http://public.dhe.ibm.com/software/dw/db2/express-c/wiki/Database_fundamentals.pdf)

SUDOLSKÁ, Miloslava. *Geografické informačné systémy* [online]. 2005 [cit. 2012-12-03]. Dostupné z <http://www.gis.sudolska.sk/databazymenu.html>

WOOD, Dan, Chris LEITER and Paul TURLEY, 2007. Beginning SQL Server 2005 Administration. Indianapolis: Wiley Publishing. ISBN 0-470-04704-6.

## **Zoznam obrázkov**

Obrázok č. 1: Tri základné typy väzieb medzi entitami .....	16
Obrázok č. 2: Príklad kódu na pridanie cudzieho kľúča.....	48

## **Zoznam grafov**

Graf č. 1: Veľkosť databáz .....	32
----------------------------------	----

## **Zoznam tabuliek**

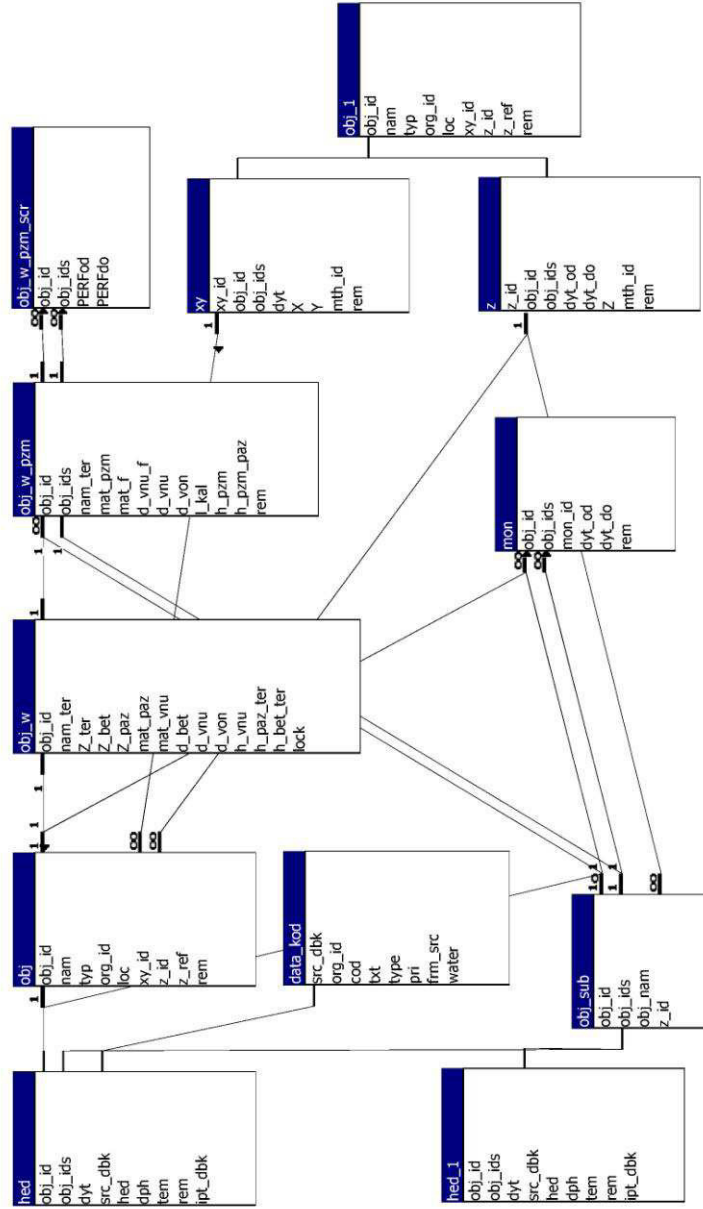
Tabuľka č. 1: Porovnanie navrhovaných riešení .....	40
Tabuľka č. 2: Problémy pri prenášaní dát.....	43

## **Zoznam príloh**

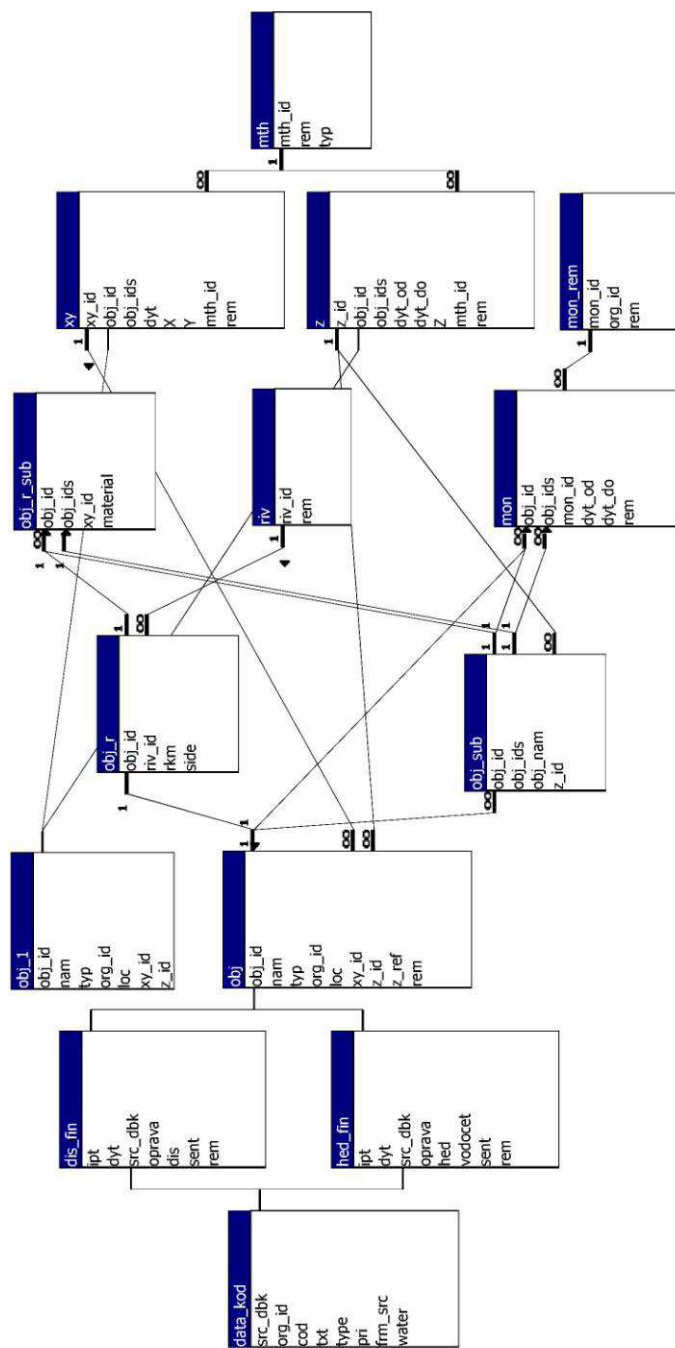
Príloha č. 1: Schémy .....	1
Príloha č. 2: Report z prenášania dát .....	6



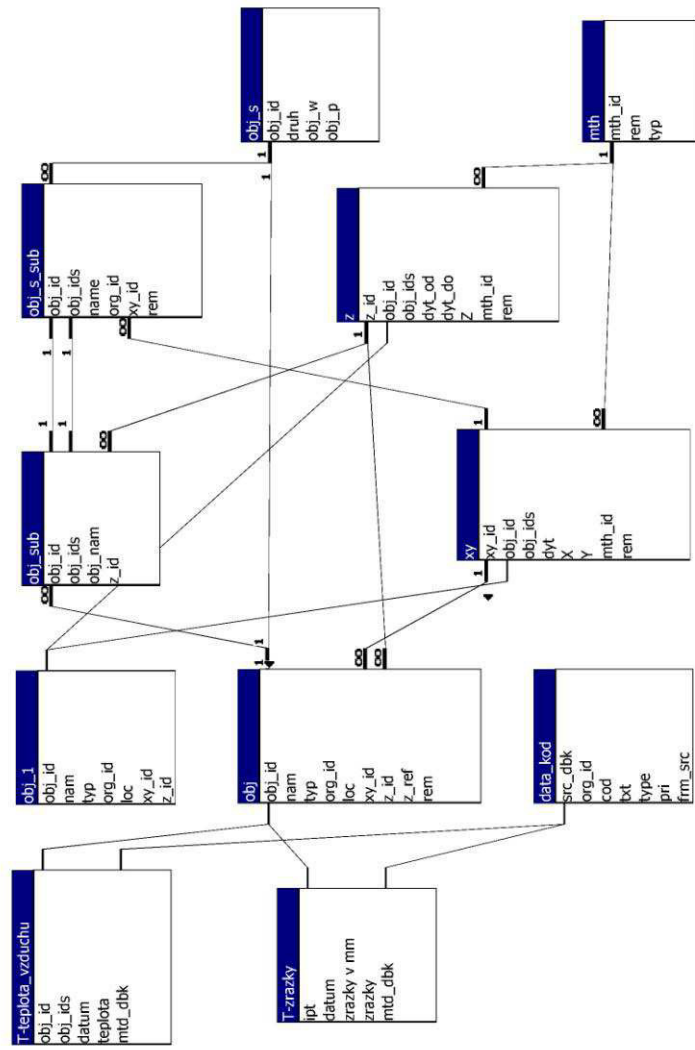
Obr. 2 Štruktúra databázy "GW"



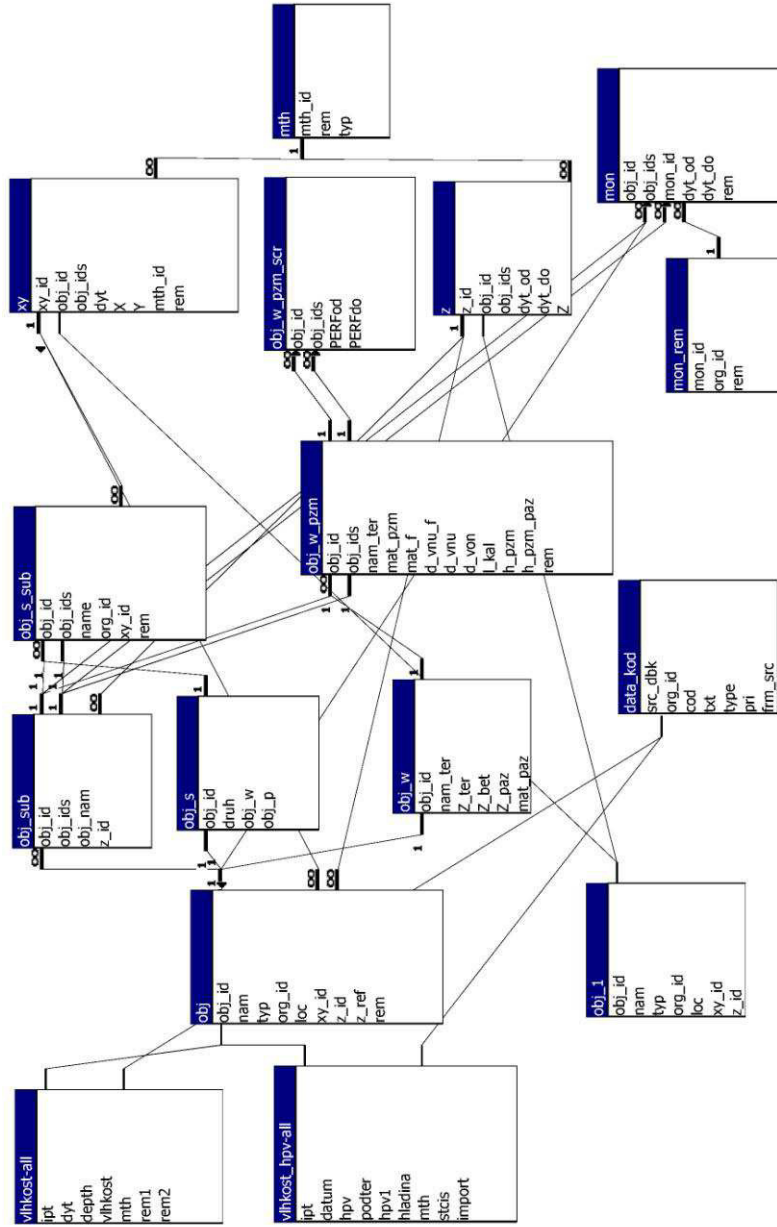
Obr. 3 Štruktúra databázy "SW\_DUN"



Obr. 4 Štruktúra databázy "Klima"



Obr. 5 Štruktúra databázy "soil\_moisture"



## Príloha č. 2: Report z prenášania dát

# Upsizing Wizard Report

## 1 Database

Microsoft Access Database: ...\\tech-tab.mdb  
SQL Server Database:

## Errors

### Table: mon

Attached table is not updatable because it has no unique index.

## Upsizing Parameters

### Table Attributes to Export

<b>Indexes</b>	<b>Table relationships:</b>
<b>Validation rules</b>	Upsized using DRI
<b>Defaults</b>	
<b>Structure only, no data</b>	<b>Timestamp fields added:</b>
	No tables

### Modifications to Existing Database

<b>Attach newly created SQL Server tables</b>	<b>Save password and user ID with attached tables</b>
---	---

### Client/Server Modifications

<b>Create a new Access client/server application.</b>	<b>Save password and user ID with application</b>
---	---

## Tables

### Table: data\_kod

	Microsoft Access	SQL Server
<b>Table Name:</b>	data_kod_local	data_kod
<b>Attached Table Name:</b>	data_kod	
<b>Aliasing Query:</b>		
<b>Validation Rule:</b>		

not

Relationship or Constraint Failed to Upsize: Server Error 1753: Column 'org.org\_id' is the same length or scale as referencing column 'data\_kod.org\_id' in foreign key 'data\_kod\_FK00'. Columns participating in a foreign key relationship must be defined with the same length and scale.  
Server Error 1750: Could not create constraint. See previous errors.

## Fields

	Microsoft Access	SQL Server
<b>Field Name:</b>	src_dbk	src_dbk

<b>Data Type:</b> Number (Integer)	smallint
<b>Field Name:</b> org_id	org_id
<b>Data Type:</b> Text(10)	nvarchar(10)
<b>Field Name:</b> cod	cod
<b>Data Type:</b> Text(1)	nvarchar(1)
<b>Field Name:</b> txt	txt
<b>Data Type:</b> Text(90)	nvarchar(90)
<b>Field Name:</b> type	type
<b>Data Type:</b> Number (Integer)	smallint
<b>Field Name:</b> pri	pri
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> frm_src	frm_src
<b>Data Type:</b> Number (Integer)	smallint
<b>Field Name:</b> water	water
<b>Data Type:</b> Number (Integer)	smallint

## Indexes

### Microsoft Access

**Name:** {D215308C-F680-4D4D-91CF-327C0EE6D065}  
**Fields:** org\_id  
**Type:** DuplicatesOK

**Name:** org\_id  
**Fields:** org\_id  
**Type:** DuplicatesOK

### SQL Server

"{D215308C-F680-4D4D-91CF-327C0EE6D065}"  
org\_id  
DuplicatesOK

org\_id  
org\_id  
DuplicatesOK

## Table: mon

### Microsoft Access

**Table Name:** mon\_local  
**Attached Table Name:** mon  
**Aliasing Query:**  
**Validation Rule:**

### SQL Server

mon

## Fields

### Microsoft Access

**Field Name:** obj\_id  
**Data Type:** Number (Long)

**Field Name:** obj\_ids  
**Data Type:** Number (Long)

**Field Name:** mon\_id  
**Data Type:** Text(10)

**Field Name:** dyt\_od  
**Data Type:** Date/Time

**Field Name:** dyt\_do  
**Data Type:** Date/Time

**Field Name:** rem  
**Data Type:** Memo

### SQL Server

obj\_id  
int

obj\_ids  
int

mon\_id  
nvarchar(10)

dyt\_od  
datetime

dyt\_do  
datetime

rem  
text

## Indexes

	Microsoft Access	SQL Server
<b>Name:</b>	{6984F7D3-3478-470C-A3D7-F4B809F84908}	"{6984F7D3-3478-470C-A3D7-F4B809F84908}"
<b>Fields:</b>	mon_id	mon_id
<b>Type:</b>	DuplicatesOK	DuplicatesOK
<b>Name:</b>	{DD362CF6-0906-4550-9CE9-B81DD24D7282}	"{DD362CF6-0906-4550-9CE9-B81DD24D7282}"
<b>Fields:</b>	obj_id	obj_id
<b>Type:</b>	DuplicatesOK	DuplicatesOK
<b>Name:</b>	{F274CDE8-5787-4039-B6D7-3A3428E8B14D}	"{F274CDE8-5787-4039-B6D7-3A3428E8B14D}"
<b>Fields:</b>	obj_id, obj_ids	obj_id, obj_ids
<b>Type:</b>	DuplicatesOK	DuplicatesOK
<b>Name:</b>	mon_id	mon_id
<b>Fields:</b>	mon_id	mon_id
<b>Type:</b>	DuplicatesOK	DuplicatesOK
<b>Name:</b>	obj%id	"obj%id"
<b>Fields:</b>	obj_id	obj_id
<b>Type:</b>	DuplicatesOK	DuplicatesOK

## Table: mon\_rem

	Microsoft Access	SQL Server
<b>Table Name:</b>	mon_rem_local	mon_rem
<b>Attached Table Name:</b>	mon_rem	
<b>Aliasing Query:</b>		
<b>Validation Rule:</b>		

## Fields

	Microsoft Access	SQL Server
<b>Field Name:</b>	mon_id	mon_id
<b>Data Type:</b>	Text(10)	nvarchar(10)
<b>Field Name:</b>	org_id	org_id
<b>Data Type:</b>	Text(8)	nvarchar(8)
<b>Field Name:</b>	rem	rem
<b>Data Type:</b>	Memo	text

## Indexes

	Microsoft Access	SQL Server
<b>Name:</b>	{0B8E2B5D-1A00-47B9-8FCA-F4E978A85062}	"{0B8E2B5D-1A00-47B9-8FCA-F4E978A85062}"
<b>Fields:</b>	org_id	org_id
<b>Type:</b>	DuplicatesOK	DuplicatesOK
<b>Name:</b>	org_id	org_id
<b>Fields:</b>	org_id	org_id
<b>Type:</b>	DuplicatesOK	DuplicatesOK

## Table: mth

	Microsoft Access	SQL Server
<b>Table Name:</b>	mth_local	mth
<b>Attached Table Name:</b>	mth	
<b>Aliasing Query:</b>		

**Validation Rule:**

**Fields**

	<b>Microsoft Access</b>	<b>SQL Server</b>
<b>Field Name:</b>	mth_id	mth_id
<b>Data Type:</b>	Text(8)	nvarchar(8)
<b>Field Name:</b>	rem	rem
<b>Data Type:</b>	Memo	text
<b>Field Name:</b>	typ	typ
<b>Data Type:</b>	Text(3)	nvarchar(3)

**Table: obj**

	<b>Microsoft Access</b>	<b>SQL Server</b>
<b>Table Name:</b>	obj_local	obj
<b>Attached Table Name:</b>	obj	
<b>Aliasing Query:</b>		
<b>Validation Rule:</b>		

**Fields**

	<b>Microsoft Access</b>	<b>SQL Server</b>
<b>Field Name:</b>	obj_id	obj_id
<b>Data Type:</b>	Number (Long)	int
<b>Field Name:</b>	nam	nam
<b>Data Type:</b>	Text(50)	nvarchar(50)
<b>Field Name:</b>	typ	typ
<b>Data Type:</b>	Text(1)	nvarchar(1)
<b>Field Name:</b>	org_id	org_id
<b>Data Type:</b>	Text(8)	nvarchar(8)
<b>Field Name:</b>	loc	loc
<b>Data Type:</b>	Text(150)	nvarchar(150)
<b>Field Name:</b>	xy_id	xy_id
<b>Data Type:</b>	Number (Long)	int
<b>Field Name:</b>	z_id	z_id
<b>Data Type:</b>	Number (Long)	int
<b>Field Name:</b>	z_ref	z_ref
<b>Data Type:</b>	Text(20)	nvarchar(20)
<b>Field Name:</b>	rem	rem
<b>Data Type:</b>	Memo	text

**Indexes**

	<b>Microsoft Access</b>	<b>SQL Server</b>
<b>Name:</b>	{05F63765-9E2B-4053-A413-E63AE0B9F9DB}	"{05F63765-9E2B-4053-A413-E63AE0B9F9DB}"
<b>Fields:</b>	xy_id	xy_id
<b>Type:</b>	DuplicatesOK	DuplicatesOK
<b>Name:</b>	{6FCC8B49-32EF-46E2-A999-D09D6955CDC3}	"{6FCC8B49-32EF-46E2-A999-D09D6955CDC3}"
<b>Fields:</b>	org_id	org_id
<b>Type:</b>	DuplicatesOK	DuplicatesOK
<b>Name:</b>	{B001DA82-A321-4D24-9EC7-	"{B001DA82-A321-4D24-9EC7-

0BDC5C25B710}	0BDC5C25B710}"
<b>Fields:</b> z_id	z_id
<b>Type:</b> DuplicatesOK	DuplicatesOK
<b>Name:</b> org_id	org_id
<b>Fields:</b> org_id	org_id
<b>Type:</b> DuplicatesOK	DuplicatesOK
<b>Name:</b> xy_id	xy_id
<b>Fields:</b> xy_id	xy_id
<b>Type:</b> DuplicatesOK	DuplicatesOK
<b>Name:</b> z_id	z_id
<b>Fields:</b> z_id	z_id
<b>Type:</b> DuplicatesOK	DuplicatesOK

## Table: obj\_r

	Microsoft Access	SQL Server
<b>Table Name:</b>	obj_r_local	obj_r
<b>Attached Table Name:</b>	obj_r	
<b>Aliasing Query:</b>		
<b>Validation Rule:</b>		

Fields	Microsoft Access	SQL Server
<b>Field Name:</b>	obj_id	obj_id
<b>Data Type:</b>	Number (Long)	int
<b>Field Name:</b>	riv_id	riv_id
<b>Data Type:</b>	Text(20)	nvarchar(20)
<b>Field Name:</b>	rkm	rkm
<b>Data Type:</b>	Number (Double)	float
<b>Field Name:</b>	side	side
<b>Data Type:</b>	Text(1)	nvarchar(1)

Indexes	Microsoft Access	SQL Server
<b>Name:</b>	{3335BB6E-686D-4CA7-9B3E-7E2AF8D8595F}	"{3335BB6E-686D-4CA7-9B3E-7E2AF8D8595F}"
<b>Fields:</b>	obj_id	obj_id
<b>Type:</b>	Unique	Unique
<b>Name:</b>	{DADF738D-4431-4F46-AC95-09001276132B}	"{DADF738D-4431-4F46-AC95-09001276132B}"
<b>Fields:</b>	riv_id	riv_id
<b>Type:</b>	DuplicatesOK	DuplicatesOK
<b>Name:</b>	riv_id	riv_id
<b>Fields:</b>	riv_id	riv_id
<b>Type:</b>	DuplicatesOK	DuplicatesOK

## Table: obj\_r\_sub

	Microsoft Access	SQL Server
<b>Table Name:</b>	obj_r_sub_local	obj_r_sub
<b>Attached Table Name:</b>	obj_r_sub	
<b>Aliasing Query:</b>		
<b>Validation Rule:</b>		

<b>Fields</b>	<b>Microsoft Access</b>	<b>SQL Server</b>
	<b>Field Name:</b> obj_id	obj_id
	<b>Data Type:</b> Number (Long)	int
	<b>Field Name:</b> obj_ids	obj_ids
	<b>Data Type:</b> Number (Long)	int
	<b>Field Name:</b> xy_id	xy_id
	<b>Data Type:</b> Number (Long)	int
	<b>Field Name:</b> material	material
	<b>Data Type:</b> Text(2)	nvarchar(2)

<b>Indexes</b>	<b>Microsoft Access</b>	<b>SQL Server</b>
	<b>Name:</b> {AA374B8A-E200-4CC2-87A5-1F0ADE03EA3E}	"{AA374B8A-E200-4CC2-87A5-1F0ADE03EA3E}"
	<b>Fields:</b> obj_id, obj_ids	obj_id, obj_ids
	<b>Type:</b> Unique	Unique
	<b>Name:</b> {F22B2003-11E9-4ED9-B08E-B322396862F5}	"{F22B2003-11E9-4ED9-B08E-B322396862F5}"
	<b>Fields:</b> obj_id	obj_id
	<b>Type:</b> DuplicatesOK	DuplicatesOK
	<b>Name:</b> xy_id	xy_id
	<b>Fields:</b> xy_id	xy_id
	<b>Type:</b> DuplicatesOK	DuplicatesOK

### Table: obj\_s

	<b>Microsoft Access</b>	<b>SQL Server</b>
<b>Table Name:</b>	obj_s_local	obj_s
<b>Attached Table Name:</b>	obj_s	
<b>Aliasing Query:</b>		
<b>Validation Rule:</b>		

<b>Fields</b>	<b>Microsoft Access</b>	<b>SQL Server</b>
	<b>Field Name:</b> obj_id	obj_id
	<b>Data Type:</b> Number (Long)	int
	<b>Field Name:</b> druh	druh
	<b>Data Type:</b> Text(3)	nvarchar(3)
	<b>Field Name:</b> obj_w	obj_w
	<b>Data Type:</b> Number (Long)	int
	<b>Field Name:</b> obj_p	obj_p
	<b>Data Type:</b> Number (Long)	int

<b>Indexes</b>	<b>Microsoft Access</b>	<b>SQL Server</b>
	<b>Name:</b> {FE6506E4-E952-466D-98D4-5994E1C56F41}	"{FE6506E4-E952-466D-98D4-5994E1C56F41}"
	<b>Fields:</b> obj_id	obj_id
	<b>Type:</b> Unique	Unique

### Table: obj\_s\_sub

<b>Microsoft Access</b>	<b>SQL Server</b>
-------------------------	-------------------

**Table Name:** obj\_s\_sub\_local  
**Attached Table Name:** obj\_s\_sub  
**Aliasing Query:**  
**Validation Rule:**

Fields	Microsoft Access	SQL Server
	<b>Field Name:</b> obj_id <b>Data Type:</b> Number (Long)	obj_id int
	<b>Field Name:</b> obj_ids <b>Data Type:</b> Number (Long)	obj_ids int
	<b>Field Name:</b> name <b>Data Type:</b> Text(50)	name nvarchar(50)
	<b>Field Name:</b> org_id <b>Data Type:</b> Text(50)	org_id nvarchar(50)
	<b>Field Name:</b> xy_id <b>Data Type:</b> Number (Long)	xy_id int
	<b>Field Name:</b> rem <b>Data Type:</b> Memo	rem text

Indexes	Microsoft Access	SQL Server
	<b>Name:</b> {2BB3F467-9617-45C5-A035-438F269A54FF} <b>Fields:</b> obj_id, obj_ids <b>Type:</b> Unique	"{2BB3F467-9617-45C5-A035-438F269A54FF}" obj_id, obj_ids Unique
	<b>Name:</b> {8328A550-357D-48FD-9A44-9184D1303471} <b>Fields:</b> xy_id <b>Type:</b> DuplicatesOK	"{8328A550-357D-48FD-9A44-9184D1303471}" xy_id DuplicatesOK
	<b>Name:</b> {B4B56EDC-BB0E-4D21-B409-C04592DA57F4} <b>Fields:</b> obj_id <b>Type:</b> DuplicatesOK	"{B4B56EDC-BB0E-4D21-B409-C04592DA57F4}" obj_id DuplicatesOK
	<b>Name:</b> org_id <b>Fields:</b> org_id <b>Type:</b> DuplicatesOK	org_id org_id DuplicatesOK
	<b>Name:</b> xy_id <b>Fields:</b> xy_id <b>Type:</b> DuplicatesOK	xy_id xy_id DuplicatesOK

### Table: obj\_sub

	Microsoft Access	SQL Server
<b>Table Name:</b>	obj_sub_local	obj_sub
<b>Attached Table Name:</b>	obj_sub	
<b>Aliasing Query:</b>		
<b>Validation Rule:</b>		

Fields	Microsoft Access	SQL Server
	<b>Field Name:</b> obj_id <b>Data Type:</b> Number (Long)	obj_id int

<b>Field Name:</b> obj_ids	obj_ids
<b>Data Type:</b> Number (Long)	int
<b>Field Name:</b> obj_nam	obj_nam
<b>Data Type:</b> Text(20)	nvarchar(20)
<b>Field Name:</b> z_id	z_id
<b>Data Type:</b> Number (Long)	int

## Indexes

Microsoft Access	SQL Server
<b>Name:</b> {0C6F53BB-9256-4ED4-9ACD-3264A39CF790}	"{0C6F53BB-9256-4ED4-9ACD-3264A39CF790}"
<b>Fields:</b> obj_id	obj_id
<b>Type:</b> DuplicatesOK	DuplicatesOK
<b>Name:</b> obj_id	obj_id
<b>Fields:</b> obj_id	obj_id
<b>Type:</b> DuplicatesOK	DuplicatesOK
<b>Name:</b> obj_ids	obj_ids
<b>Fields:</b> obj_ids	obj_ids
<b>Type:</b> DuplicatesOK	DuplicatesOK
<b>Name:</b> obj_nam	
<b>Fields:</b> obj_nam	
	Server Error 1505: The CREATE UNIQUE INDEX statement terminated because a duplicate key was found for the object name 'dbo.obj_sub' and the index name 'obj_nam'. The duplicate key value is (<NULL>).
	Server Error 3621: The statement has been terminated.
<b>Type:</b> DuplicatesOK	DuplicatesOK
<b>Name:</b> z_id	z_id
<b>Fields:</b> z_id	z_id
<b>Type:</b> DuplicatesOK	DuplicatesOK
<b>Name:</b> zobj_sub	zobj_sub
<b>Fields:</b> z_id	z_id
<b>Type:</b> DuplicatesOK	DuplicatesOK

## Table: obj\_w

Microsoft Access	SQL Server
<b>Table Name:</b> obj_w_local	obj_w
<b>Attached Table Name:</b> obj_w	
<b>Aliasing Query:</b>	
<b>Validation Rule:</b>	

## Fields

Microsoft Access	SQL Server
<b>Field Name:</b> obj_id	obj_id
<b>Data Type:</b> Number (Long)	int
<b>Field Name:</b> nam_ter	nam_ter
<b>Data Type:</b> Text(50)	nvarchar(50)
<b>Field Name:</b> Z_ter	Z_ter
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> Z_bet	Z_bet
<b>Data Type:</b> Number (Single)	real

<b>Field Name:</b> Z_paz	Z_paz
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> mat_paz	mat_paz
<b>Data Type:</b> Text(1)	nvarchar(1)
<b>Field Name:</b> mat_vnu	mat_vnu
<b>Data Type:</b> Text(8)	nvarchar(8)
<b>Field Name:</b> d_bet	d_bet
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> d_vnu	d_vnu
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> d_von	d_von
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> h_vnu	h_vnu
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> h_paz_ter	h_paz_ter
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> h_bet_ter	h_bet_ter
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> lock	lock
<b>Data Type:</b> Yes/No	bit

## Indexes

### Microsoft Access

**Name:** {97BE1391-6935-4C98-BC90-2C6ABE60E628}  
**Fields:** obj\_id  
**Type:** Unique

### SQL Server

"{97BE1391-6935-4C98-BC90-2C6ABE60E628}"  
obj\_id  
Unique

## Table: obj\_w\_pzm

### Microsoft Access

**Table Name:** obj\_w\_pzm\_local  
**Attached Table Name:** obj\_w\_pzm  
**Aliasing Query:**  
**Validation Rule:**

### SQL Server

obj\_w\_pzm

## Fields

### Microsoft Access

**Field Name:** obj\_id  
**Data Type:** Number (Long)

**Field Name:** obj\_ids  
**Data Type:** Number (Long)

**Field Name:** nam\_ter  
**Data Type:** Text(50)

**Field Name:** mat\_pzm  
**Data Type:** Text(1)

**Field Name:** mat\_f  
**Data Type:** Text(8)

### SQL Server

obj\_id  
int

obj\_ids  
int

nam\_ter  
nvarchar(50)

mat\_pzm  
nvarchar(1)

mat\_f  
nvarchar(8)

<b>Field Name:</b> d_vnu_f	d_vnu_f
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> d_vnu	d_vnu
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> d_von	d_von
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> l_kal	l_kal
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> h_pzm	h_pzm
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> h_pzm_paz	h_pzm_paz
<b>Data Type:</b> Number (Single)	real
<b>Field Name:</b> rem	rem
<b>Data Type:</b> Memo	text

## Indexes

Microsoft Access	SQL Server
<b>Name:</b> {C2A6B52F-D7A9-45F2-9773-90B5AF09D6E4}	"{C2A6B52F-D7A9-45F2-9773-90B5AF09D6E4}"
<b>Fields:</b> obj_id	obj_id
<b>Type:</b> DuplicatesOK	DuplicatesOK
<b>Name:</b> {E352BECA-FE70-4D4F-B0A0-C650639CCEA1}	"{E352BECA-FE70-4D4F-B0A0-C650639CCEA1}"
<b>Fields:</b> obj_id, obj_ids	obj_id, obj_ids
<b>Type:</b> Unique	Unique
<b>Name:</b> pie_id	pie_id
<b>Fields:</b> obj_ids	obj_ids
<b>Type:</b> DuplicatesOK	DuplicatesOK
<b>Name:</b> well_id	well_id
<b>Fields:</b> obj_id	obj_id
<b>Type:</b> DuplicatesOK	DuplicatesOK

## Table: obj\_w\_pzm\_scr

	Microsoft Access	SQL Server
<b>Table Name:</b>	obj_w_pzm_scr_local	obj_w_pzm_scr
<b>Attached Table Name:</b>	obj_w_pzm_scr	
<b>Aliasing Query:</b>		
<b>Validation Rule:</b>		

## Fields

	Microsoft Access	SQL Server
<b>Field Name:</b> obj_id	obj_id	obj_id
<b>Data Type:</b> Number (Long)		int
<b>Field Name:</b> obj_ids	obj_ids	obj_ids
<b>Data Type:</b> Number (Long)		int
<b>Field Name:</b> PERFod	PERFod	PERFod
<b>Data Type:</b> Number (Single)		real
<b>Field Name:</b> PERFdo	PERFdo	PERFdo
<b>Data Type:</b> Number (Single)		real

## Indexes

**Microsoft Access**  
**Name:** {AEF32E76-1EF7-4063-9F5D-E178B0720886}  
**Fields:** obj\_id, obj\_ids  
**Type:** DuplicatesOK  
**Name:** ÚSEKYperf  
**Fields:** obj\_id  
**Type:** DuplicatesOK

**SQL Server**  
"{AEF32E76-1EF7-4063-9F5D-E178B0720886}"  
obj\_id, obj\_ids  
DuplicatesOK  
"ÚSEKYperf"  
obj\_id  
DuplicatesOK

## Table: org

**Microsoft Access**  
**Table Name:** org\_local  
**Attached Table Name:** org  
**Aliasing Query:**  
**Validation Rule:**

**SQL Server**  
org

## Fields

**Microsoft Access**  
**Field Name:** org\_id  
**Data Type:** Text(8)  
**Field Name:** name  
**Data Type:** Text(150)  
**Field Name:** rem  
**Data Type:** Memo

**SQL Server**  
org\_id  
nvarchar(8)  
name  
nvarchar(150)  
rem  
text

## Indexes

**Microsoft Access**  
**Name:** id  
**Fields:** org\_id  
**Type:** DuplicatesOK

**SQL Server**  
id  
org\_id  
DuplicatesOK

## Table: riv

**Microsoft Access**  
**Table Name:** riv\_local  
**Attached Table Name:** riv  
**Aliasing Query:**  
**Validation Rule:**

**SQL Server**  
riv

## Fields

**Microsoft Access**  
**Field Name:** riv\_id  
**Data Type:** Text(20)  
**Field Name:** rem  
**Data Type:** Text(80)

**SQL Server**  
riv\_id  
nvarchar(20)  
rem  
nvarchar(80)

## Table: xy

**Microsoft Access**  
**Table Name:** xy\_local  
**Attached Table Name:** xy  
**Aliasing Query:**  
**Validation Rule:**

**SQL Server**  
xy

Relationship or Constraint Failed to Upsize: Server Error 1753: Column 'mth.mth\_id' is

'xy\_FK00'.

not the same length or scale as referencing column 'xy.mth\_id' in foreign key

Columns participating in a foreign key relationship must be defined with the same length and scale.

Server Error 1750: Could not create constraint. See previous errors.

## Fields

	Microsoft Access	SQL Server
<b>Field Name:</b>	xy_id	xy_id
<b>Data Type:</b>	Number (Long)	int
<b>Field Name:</b>	obj_id	obj_id
<b>Data Type:</b>	Number (Long)	int
<b>Field Name:</b>	obj_ids	obj_ids
<b>Data Type:</b>	Number (Long)	int
<b>Field Name:</b>	dvt	dvt
<b>Data Type:</b>	Date/Time	datetime
<b>Field Name:</b>	X	X
<b>Data Type:</b>	Number (Double)	float
<b>Field Name:</b>	Y	Y
<b>Data Type:</b>	Number (Double)	float
<b>Field Name:</b>	mth_id	mth_id
<b>Data Type:</b>	Text(10)	nvarchar(10)
<b>Field Name:</b>	rem	rem
<b>Data Type:</b>	Memo	text

## Indexes

	Microsoft Access	SQL Server
<b>Name:</b>	{FF1A7FD7-9A2C-4624-B1B2-4E687D3DE343}	"{FF1A7FD7-9A2C-4624-B1B2-4E687D3DE343}"
<b>Fields:</b>	mth_id	mth_id
<b>Type:</b>	DuplicatesOK	DuplicatesOK
<b>Name:</b>	mth_id	mth_id
<b>Fields:</b>	mth_id	mth_id
<b>Type:</b>	DuplicatesOK	DuplicatesOK
<b>Name:</b>	obj_id	obj_id
<b>Fields:</b>	obj_id	obj_id
<b>Type:</b>	DuplicatesOK	DuplicatesOK
<b>Name:</b>	xy_id	xy_id
<b>Fields:</b>	xy_id	xy_id
<b>Type:</b>	DuplicatesOK	DuplicatesOK

## Table: z

	Microsoft Access	SQL Server
<b>Table Name:</b>	z_local	z
<b>Attached Table Name:</b>	z	
<b>Aliasing Query:</b>		
<b>Validation Rule:</b>		

Relationship or Constraint Failed to Upsize: Server Error 1753: Column 'mth.mth\_id' is not the same length or scale as referencing column 'z.mth\_id' in foreign key 'z\_FK00'. Columns participating in a foreign key relationship must be defined with the same length and scale.

Server Error 1750: Could not create constraint. See previous errors.

<b>Fields</b>	<b>Microsoft Access</b>	<b>SQL Server</b>
	<b>Field Name:</b> z_id <b>Data Type:</b> Number (Long)	z_id int
	<b>Field Name:</b> obj_id <b>Data Type:</b> Number (Long)	obj_id int
	<b>Field Name:</b> obj_ids <b>Data Type:</b> Number (Long)	obj_ids int
	<b>Field Name:</b> dyt_od <b>Data Type:</b> Date/Time	dyt_od datetime
	<b>Field Name:</b> dyt_do <b>Data Type:</b> Date/Time	dyt_do datetime
	<b>Field Name:</b> Z <b>Data Type:</b> Number (Single)	Z real
	<b>Field Name:</b> mth_id <b>Data Type:</b> Text(10)	mth_id nvarchar(10)
	<b>Field Name:</b> rem <b>Data Type:</b> Memo	rem text

<b>Indexes</b>	<b>Microsoft Access</b>	<b>SQL Server</b>
	<b>Name:</b> {96C2E9B3-90D8-4CC5-AD5E-1B95E983D2E5} <b>Fields:</b> mth_id <b>Type:</b> DuplicatesOK	"{96C2E9B3-90D8-4CC5-AD5E-1B95E983D2E5}" mth_id DuplicatesOK
	<b>Name:</b> mth_id <b>Fields:</b> mth_id <b>Type:</b> DuplicatesOK	mth_id mth_id DuplicatesOK
	<b>Name:</b> obj_id <b>Fields:</b> obj_id <b>Type:</b> DuplicatesOK	obj_id obj_id DuplicatesOK
	<b>Name:</b> xy_id <b>Fields:</b> z_id <b>Type:</b> DuplicatesOK	xy_id z_id DuplicatesOK