

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Zadání bakalářské práce

Student: **Jindřich Goldmann**
Studijní program: B2647 Informační a komunikační technologie
Studijní obor: 2612R025 Informatika a výpočetní technika
Téma: Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: CID International a.s.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. David Seidl, Ph.D.**

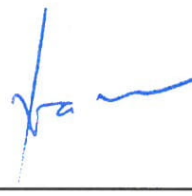
Konzultant bakalářské práce: Ing. Lukáš Adámek

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018



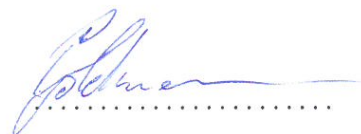
doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry



prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 17. dubna 2018



.....

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26 odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 17. dubna 2018


.....
 CID International a.s.
provozovna Ostrava
Nádražní 184/477
702 00 Ostrava
tel. 596 133 181

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, především mé rodině za podporu během celého studia.

Abstrakt

V této práci popisuji absolvování odborné praxe ve firmě CID International a.s., přičemž se pokusím stručně shrnout základní informace o společnosti, mém začlenění do její struktury, popis zadaných úkolů spolu s jejich řešeními a také použité technologické postupy. Jelikož jsem během praxe pracoval především na množině menších úkolů, které lze rozdělit do jednotlivých oblastí, popíši je včetně řešení a výsledků. Závěr práce je věnován celkovému zhodnocení, přínosu daných úloh pro jejich budoucí využití, uplatněné znalosti nabyté během mého dosavadního studia, ale také znalosti a zkušenosti, které jsem získal až právě díky praxi.

Klíčová slova: bakalářská práce, SQL, relační databáze, databázový projekt, optimalizace

Abstract

This bachelor thesis describes my participation in professional practice organized by CID International a.s. The content consists of elementary information about the company, my placement in its business structures, description of tasks with solutions and used technologies. Due to working primarily on small assignments, which could be covered by broader topics, I am going to describe these topics with examples and accomplished results. The conclusion deals with the final evaluation report, expected usage of the assignments, applied and missing knowledge.

Key Words: bachelor thesis, SQL, relational database, database project, optimization

Obsah

Seznam použitých zkratk a symbolů	8
Seznam obrázků	9
Seznam výpisů zdrojového kódu	10
1 Úvod	11
2 Představení společnosti a pracovní zaměření	12
2.1 Základní popis společnosti	12
2.2 Bližší informace k produktům společnosti	12
2.3 Zařazení studenta ve společnosti	12
2.4 Seznámení a první dny	12
3 Používaný Software a použité technologie	14
3.1 Úvod	14
3.2 SQL Server	14
3.3 Plan Explorer	14
3.4 SharePoint	14
3.5 LORI, CisAdmin a IS Market	15
3.6 XML	15
3.7 WCF	15
3.8 CLR	15
3.9 VPN a OpenVPN	15
4 Analýza problému	16
4.1 Tvorba procedur	16
4.2 Optimalizace klientských databází	21
4.3 Administrace klientských databází	23
4.4 Databázové testování	24
4.5 Tvorba metodiky	27
5 Závěr	28
5.1 Využité a scházející znalosti	28
5.2 Zhodnocení a dosažené výsledky	28
Zdroje	29

Seznam použitých zkratk a symbolů

SQL	– Structured Query Language
SSMS	– SQL Server Management Studio
T-SQL	– Transcript-Structured Query Language
B2B	– Business to Business
WCF	– Windows Communication Foundation
MSDN	– Microsoft Developer Network
CLR	– Common Language Runtime
DMV	– Dynamic Management Views

Seznam obrázků

1	Skalární funkce z dotazu	22
2	Ukázka testování	25
3	Výňatek z metodiky	27

Seznam výpisů zdrojového kódu

1	Ukázka parsovací procedury	17
2	Rekurzivní Common Table Expression	19

1 Úvod

Tato bakalářská práce si klade za cíl popsat průběh odborné praxe.

Jako formu realizace bakalářské práce jsem si zvolil absolvování odborné praxe ve společnosti CID International a.s. Učinil jsem tak především z důvodu, že jsem se chtěl seznámit s vývojem software v podnikající firmě. Zároveň jsem považoval pro mé následné uplatnění za velmi přínosné, že bych mohl tímto způsobem získat na závěr bakalářského studia cenné zkušenosti z průmyslu.

Začátek práce věnuje pozornost zejména představení společnosti a mému začlenění do její struktury. Jelikož se jedná o společnost relativně malou, rozhodl jsem se tuto část popsat podrobněji tak, aby čtenář měl možnost získat lepší povědomí o jejích aktivitách.

V následující kapitole popisují technologie a software, s kterým jsem po dobu praxe pracoval.

Na tuto část navazuje hlavní kapitola. V hlavní kapitole se zabývám zadanými úkoly a popisem jejich řešení. Vzhledem ke skutečnosti, že jsem v rámci praxe pracoval především na menších úkolech, je tato kapitola rozdělena do několika tématických okruhů.

Na závěr práce uvádím nejen výčet použitých znalostí získaných během studia spolu se znalostmi, které jsem díky praxi nabyl a také celkové zhodnocení a přínos praxe.

2 Představení společnosti a pracovní zaměření

2.1 Základní popis společnosti

Společnost CID International a.s. se zabývá vývojem a implementací informačních systémů v oblasti dopravy, spedice a skladování. Vyjma samotného vývoje jsou jejími vedlejšími činnostmi také poskytování konzultací, outsourcingu, školení a další.

Pozici na trhu si společnost vydobyla především dlouholetým poskytováním kvalitních a moderních produktů v oblasti, v níž se na tuzemském trhu konkurence ve větší míře nevyskytuje a zahraniční konkurence často nabízí ceny nesrovnatelně vyšší v poměru k nabízeným službám.

Společnost CID International a.s. je členem skupiny OLTIS Group a.s. OLTIS Group a.s. tvoří několik dceřiných firem, které působí jak v České republice, tak i v mnoha dalších evropských zemích. Oblastí, na kterou se celá skupina primárně zaměřuje je vývoj informačních systémů pro dopravu a logistiku.

Svou budoucnost společnost vidí v neustálém zapracovávání inovativních metod informačních technologií do svých produktů. Konkrétně se jedná o metody spojené s detailním reportingem a grafovou analýzou, tak aby zákazníci získali lepší přehled o svých firemních aktivitách.

2.2 Bližší informace k produktům společnosti

Portfolio společnosti je tvořeno zejména šesticí hlavních produktů, které jsou vyvíjeny jednotlivými týmy. Kromě zaměstnanců v týmech, které tvoří programátoři, analytici, konzultanti a specialisté, jsou ve firmě také administrátoři, obchodní zástupci a ředitelé úseků.

Stěžejním produktem je informační systém LORI sloužící pro silniční dopravu, spedici a sběrnou službu. Značnou výhodou tohoto systému je flexibilita a přenesitelnost. Zákazníkům je dodán základní systém, který se na základě konkrétních požadavků individualizuje.

2.3 Zařazení studenta ve společnosti

Ve firmě jsem pracoval na pozici databázový specialista Microsoft Server SQL ve vývojovém týmu LORI. Alternativně jsem spolupracoval s kolegy z jiných týmů na řešení problémů týkajících se databází. Mou pracovní náplní byly zejména tvorba procedur, analýza klientských databází včetně jejich následných úprav, tvorba databázové metodiky pro interní zaměstnance a administrace databází.

2.4 Seznámení a první dny

Na úvod praxe provedl obchodní zástupce společnosti školení ohledně firemních produktů, přičemž součástí školení byla také prezentace systému, na němž jsem měl následně pracovat. Dále jsem se seznámil s vnitropodnikovým projektovým systémem. Jedná se o platformu, která tvoří

jeden z produktů celé skupiny a byl upraven pro vnitřní použití. Slouží pro vykazování docházky, organizaci úkolů, sledování termínů a podobně.

3 Používaný Software a použité technologie

3.1 Úvod

Během praxe jsem používal technologie spjaté nějakým způsobem s SQL Serverem. V technologiích, které jsem před nástupem na praxi alespoň trochu znal, jsem si prohloubil a zdokonalil znalosti. Hojně jsem využíval také volně dostupných webových aplikací jako je například Statistics Parser.

3.2 SQL Server

Při práci jsem používal především program SQL Server Management Studio(dále jen SSMS) ve verzi 2017, který se užívá pro správu SQL Serveru. Na základě zakoupené verze a edice SQL Serveru při zákaznické databázi se v některých případech měnil můj přístup k řešení problémů, protože nabízené edice a verze se liší ve funkcionalitě a kompatibilitě. Většina zákazníků měla zakoupený SQL Server v edici Standart, ovšem našli se i tací, kteří měli pouze edici Express.

Často užívaným nástrojem během mé práce byl SSMS Server Profiler. Jedná se o nástroj pro sledování provozu v databázi. Mnohokrát jsem ho využil například pro ladění či optimalizaci databází.

3.3 Plan Explorer

Plan Explorer je volně dostupný program sloužící k analýzám SQL dotazů. Dle mého názoru nabízí lepší a přehlednější možnosti, než například základní zobrazení exekučního plánu v SSMS. Tento program je vyvíjen společností SentryOne a výhodou je, že tato společnost vyvíjí také komerční produkty a po vydání nové verze se jejich funkcionality posouvají do volně dostupných programů.

3.4 SharePoint

SharePoint je webová platforma od Microsoftu, která obsahuje mnoho funkcí jako je osobní cloud, intranet, firemní sociální síť, úložiště aplikací a další. Nejčastěji jsem tuto platformu používal jako prostor pro ukládání a následný přístup k firemním dokumentům.

3.5 LORI, CisAdmin a IS Market

LORI je software na kterém jsem primárně pracoval. Architektura je postavena na principu třívrstvého modelu, kde aplikační vrstvu tvoří desktopová aplikace realizována formou tenkého klienta. Tato aplikace komunikuje s aplikačním serverem a datovou vrstvu tvoří databázový systém.

CisAdmin je interní software ke správě LORI. Nabízí například nastavení práv k jednotlivým modulům LORI, vytváření filtrovacích oken a podobně.

IS Market je interní software ke správě požadavků, návrhu a vykonání jejich řešení a k vykazování odvedené práce.

3.6 XML

XML (eXtensible Markup Language) je značkovací jazyk. Umožňuje snadné vytváření konkrétních značkovacích jazyků pro různé účely a různé typy dat.

3.7 WCF

Windows Communication Foundation je sada knihoven sloužící pro vytváření servisně orientovaných aplikací.

3.8 CLR

Common Language Runtime je virtuální stroj .Net frameworku umožňující zpracování .Net programu.

3.9 VPN a OpenVPN

Virtuální privátní síť je vytvoření komunikačního kanálu mezi privátními sítěmi prostřednictvím veřejné sítě takovým způsobem, že počítače mezi sebou komunikují, jako kdyby byly uvnitř privátní sítě.

4 Analýza problému

4.1 Tvorba procedur

Prvním problémem, jemuž se chci věnovat je sestavení databázových procedur. Tuto část jsem rozdělil do podsekcí, které odpovídají oblastem k čemu dané procedury slouží. Z těchto oblastí vždy popíši alespoň jednu proceduru detailněji.

4.1.1 Obecné procedury

Zadáním první procedury bylo vytvořit pohled v informačním systému Lori, který bude sloužit k porovnání průměrné spotřeby podle dvou zdrojů a upozorní na výraznější výkyvy hodnot z těchto zdrojů.

Nejprve jsem začal úkol analyzovat ze širokého úhlu pohledu tak, abych věděl jak modul bude koncový uživatel používat, jak budou vypadat struktury a prezentace výsledků a v neposlední řadě odkud budou čerpána data. Obecně jsem se během praxe často doptával na to, jaké vstupy bude zákazník používat, a to především z důvodu, že LORI nabízí mnoho způsobů zadání například spotřeby jízdy, které naplňují odlišné databázové tabulky. Problém spočívá v tom, že zákazník zadává z nějakého důvodu jen některé vstupy. Orientační informace k vypracování jsem obdržel od firemního analytika a na mě byla konkrétní implementace a zasazení do systému.

Po získání základní představy o funkci procedury jsem začal přemýšlet o tom, jak proceduru začlenit do databázového systému LORI. Vytvořil jsem si tabulku, z níž byla data prezentována v modulu a procedura pak takto plnila tabulku po zadání vstupu. Následně jsem potřeboval zajistit, aby data z tabulky byla prezentována v LORI. V databázovém systému k tomu slouží tabulka nazvána tree, která má atributy uzpůsobené pro reprezentaci stromové struktury a lze tak pohled umístit na patřičnou pozici do LORI.

Jelikož procedura používá vstupy pro filtraci vrácených hodnot, vytvořil jsem tak oknový filtr se třemi vstupy. Tuto práci mi velice usnadnil interní program CisAdmin přes který lze taková filtrovací okna vytvářet. Také jsem jej použil k přidělení práv uživatelům pro práci s pohledem. Filtr automaticky naplní databázovou tabulku a získat tak hodnoty v proceduře je snadné.

Procedura se spouští pomocí univerzálního spouštěče procedur, do kterého jsem zanesl parametry mé procedury s vytvořeným filtrem. Samotnou proceduru jsem vytvořil tak, že jsem si rozdělil výpočet hodnot podle palubního počítače vozidla a podle fakturace. Získané hodnoty jsem porovnal, a pokud odpovídaly toleranci zadané uživatelem, tak bude v LORI řádek označen zelenou barvou. Na závěr jsem dočasnou tabulku překopíroval do tabulky pohledu.

Součástí tvorby každé procedury bylo její otestování. Jelikož se ladění pohledu nedá předem nijak fázovat, často jsem nad touto částí strávil poměrně dlouhou dobu.

Další proceduru, kterou bych na tomto místě rád zmínil je procedura sloužící pro automatizaci vkládání dat ze souboru XML. S XML dotazováním v SSMS jsem nikdy předtím nepracoval.

Zkušenosti s touto problematikou neměl ani nikdo z mých kolegů a tudíž bylo mým úkolem prozkoumat možnosti v tomto směru a následně vyhodnotit, kterou z nich bude nejvýhodnější zvolit. Použil jsem funkci value, která slouží pro získávání hodnot elementů a konstrukci nodes.

```
;WITH XMLNAMESPACES (DEFAULT 'http://schemas.ewals.com/communication')
INSERT INTO #Temp_Unloader(lIDCarrierInstruction, lIDConsignmentNumber)
SELECT
    a.b.value('id[1]', 'int') as idCarrierInstruction,
    c.d.value('consignmentNumber[1]', 'int') as idConsignmentNumber
FROM
    @xmlFile.nodes('//iPlanCommunicationMessage') a(b)
    CROSS APPLY a.b.nodes('content/data/consignments/consignment')

DECLARE @xmlFixedFile XML
SET @xmlFixedFile =
(
    SELECT * FROM OPENROWSET(BULK 'C:\data.xml', SINGLE_BLOB) AS data
)
EXEC csp_XmlExtractData @xmlFixedFile
```

Výpis 1: Ukázka parsovací procedury

Ze strany zákazníka mi byl zaslán ukázkový soubor typu xml, analýza jednotlivých elementů a ukázkový objednávací list. Od firemního analytika jsem dostal obecný seznam informací, které mám ze úkol ze souboru získat. Vytvořil jsem si tedy pro tyto potřeby odpovídající tabulky. Jelikož informační obsah souboru byl značně rozsáhlý, největší část jsem strávil analýzou, která data jsou podstatná a jaké pro ně použít datové typy.

4.1.2 Výpočet silniční daně

Byl mi přidělen úkol pro výpočet silniční daně. Tento úkol jsem zpracovával od samotného počátku včetně analýzy až po předání řešení konzultantovi.

Začal jsem tím, že jsem si z volně dostupných zdrojů nastudoval výpočet této daně. Po přečtení několika článků mi bylo zřejmé, že budu muset pohled propojit s uživatelsky definovanými sazebníky, a to hned z několika důvodů. Silniční daň se počítá zvlášť pro osobní vozidla a zvlášť pro ostatní typy vozidel. Pro každý typ vozidel je sazba daně dána určitými parametry konkrétního vozidla. Zároveň může být daň snížena popřípadě zvýšena, jestliže vozidlo podléhá určitým podmínkám. V nejkrajším případě může být plátcem daně od této konkrétní povinnosti zcela osvobozen.

Na základě tohoto jsem usoudil, že funkčnost pohledu musí být především flexibilní pro pozdější úpravy, jelikož se sazby a podmínky mohou často měnit společně s legislativou.

Vytvořil jsem si tři uživatelsky definované sazebníky. První dva sazebníky slouží k tomu, aby uživatel zadal sazbu pro osobní nebo ostatní vozidla dle zákona. Třetí sazebník slouží k zadání podmínek pro snížení, respektive zvýšení daně dle jednotlivých odstavců v zákoně.

Při nasazení jsem sazebníky vyplnil sám a následující správnost a aktuálnost těchto dat bude ponechána na starost uživateli. U třetího pohledu bude po zadání přidán záznam do číselníku k parametrům vozidla. Pokud by například nabyla účinnosti novela zákona vedoucí ke snížení daně u elektromobilů, došlo by k přidání parametru a naplnění vazební tabulky k jednotlivým autům.

Funkčnost procedury byla taková, že nejprve do dočasné tabulky vložím všechny parametry pro výpočet, a následně kurzorem projdu tuto tabulku. V kurzoru jsou ošetřeny jednotlivé podmínky pro slevu, či navýšení daně. Některé jsou jednoduché, jiné volají funkci pro doplnění hodnot. Jedná se o například o funkci pro dopočítání jak dlouho bylo auto v depozitu. K této proceduře jsem vytvořil i dokumentaci, která bude sloužit obchodním zástupcům jako podpora při nabízení rozšíření o tento pohled u nových zákazníků.

4.1.3 Vytvoření skriptu reprezentace práv k užívání systému

Dostal jsem za úkol vytvořit skript, jenž by do tabulky zobrazoval všechny moduly, podmoduly a funkce. Následně k nim měl skript také přiřazovat, které role s nimi mohou pracovat. V prvním sloupci tabulky je název modulu a v dalších sloupcích jsou operace, které s nimi role mohou provádět.

Názvy modulů jsem čerpal z tabulky s názvem *tree*, kde se nachází atribut *parent*, který určuje rodiče modulu. V tabulce jsem musel zařídit, aby názvy modul reprezentovaly strukturu rodič-potomek odsazením například sérií mezer. Účelem tohoto formatování je, aby na první pohled bylo zřejmé, kdo je rodič a kdo potomek.

Uvažoval jsem, jak provést rekurzi, která mi toto dohledá. Vytvořil jsem si Common Table Expression, která mi pomocí rekurze do dočasné tabulky vloží id záznamu, rodiče tohoto záznamu, level - určující pořadí na stejné úrovni a hloubku.

```

with cte as
(
  select
    lIdtree,
    lParent,
    llevel,
    0 as depth
  from
    p_tree
  where
    lidtree = @lidtree
  union all
  select
    p_tree.lIDTree,
    p_tree.lParent,
    p_tree.lLevel,
    depth + 1
  from
    cte
  join
    p_tree
    on p_tree.lParent = cte.lidtree
)
insert into
  #temp(lidtree, lparent, llevel, ldepth)
select *
from
  cte
order by
  lLevel

```

Výpis 2: Rekurzivní Common Table Expression

Následně jsem provedl formátování dle atributu lDepth. Pro načtení práv jsem použil kurzor, jenž prochází všechny moduly a pomocí konstrukce STUFF + FOR XML přiřadí každé operaci role, které ji mohou provádět. Toto jsem později přepsal spolu s funkcí pivot. Funkce pivot, zjednodušeně řečeno, převádí řádky na sloupce.

4.1.4 Promazávání záznamů

V souvislosti s importem dat z palubního počítače bylo potřeba řešit nárůst objemu dat průběžným odstraňováním těch neaktuálních. Poměrně dlouho jsme vedli diskuzi o tom, jak stará data si můžeme dovolit odstranit. Původně jsem měl vytvořit univerzální proceduru, která bude provádět postupné mazání.

Vycházel jsem z procedury pro mazání logu, kterou jsem upravil do dynamického SQL. Vstupními parametry procedury jsou počet záznamů smazaných v každém cyklu, po kolika cyklech se má provést checkpoint v případě simple recovery modelu, popřípadě po kolika cyklech se má provést záloha transakčního logu v případě full recovery modelu.

Jelikož jsou tabulky navzájem propojeny cizími klíči, tak se procedura pro jednotlivé tabulky bude volat v takovém pořadí, aby nedocházelo ke konfliktu. Určitě existují sofistikovanější řešení, nicméně v tomto případě bylo pro svou rychlost a jednoduchost vybráno toto.

Jednotné mazání vysokého počtu záznamů může lehce způsobit pád databáze. V tomto případě, kdy databáze sídlí v cloudu spolu s ostatními zákazníky, by to mohlo mít vážné důsledky. Ještě před implementací jsem řešení konzultoval s vedoucím analytikem a programátorem. Zjistilo se, že vysoký nárůst dat je způsoben jejich stahováním za poslední tři dny každé tři minuty. Toto číslo se snížilo na posledních šest hodin, tak aby byly zachovány duplicity v případě problému.

4.1.5 Vytvoření pohledů pro interaktivní map

Součástí tvorby nových funkcionalit LORI bylo vytvoření webové aplikace, která bude sloužit jako interaktivní mapa pro sledování polohy a historie objektů. Na tomto jsem se podílel vytvořením pohledů, které vybírají data z metadatabáze, která pak budou zobrazena v mapě.

Jelikož se na této aplikaci podíleli kolegové z jiných týmů a poboček, strávil jsem podstatnou část analýzou a konzultací. Probíral jsem s nimi například, jak řešit situaci, kdy dva objekty budou mít stejné ID, jaká jsou konvence jejich pojmenování a podobně. Samotný pohled funguje tak, že vybírá jako první objekt auta a k nim poslední známou polohu spolu s popisem místa. Přidání dalšího objektu (například vlaku) je přes konstrukci UNION ALL tak, aby nedocházelo ke kontrole duplicit.

Funkcí druhého pohledu je dohledat historii vozidel. Tento pohled vybírá objekty z prvního pohledu a přiřazuje jim trasu. Výhodou je, že SSMS nabízí zobrazení typu GEOGRAPHY a je schopen v prostoru zobrazit polygon těchto bodů.

4.1.6 Vytvoření validační procedury

Další typ procedury, kterou jsem vytvořil, byla procedura validační. Funkčnost je taková, že při zadání určitých hodnot do LORI, na které je tato validace navázána, dojde k naplnění dočasně vytvořených tabulek, které odpovídají tabulkám, s kterými modul pracuje. Při validaci pak

nepracuji přímo se zadanými daty, ale s jejich duplicitami v dočasných tabulkách. Předtím se validace prováděla přímo s konkrétními daty a způsobovalo to zpomalení.

Vytvořil jsem si tedy proceduru, která načte hodnoty z dočasných tabulek, provede validaci a v případě neshody vloží záznam do univerzální tabulky pro výsledek validace s popisem neshody. Konkrétně tato procedura ověřovala, že vždy jednou ročně v měsíc nástupu řidiče je mu připsán platový bonus.

4.1.7 Procedura kontroly termínů

Podobný typ procedury jsem vytvářel u kontroly termínů. V souvislosti s tímto, jsem navrhnul databázový konceptuální model, který mapuje tabulky zajišťující funkčnost kontroly termínů. Tento model bude sloužit kolegům k tomu aby věděli, do jakých tabulek vložit záznam o proceduře pro zajištění správné funkčnosti kontroly termínů. Samotná procedura provádí kontrolu vložení kurzovního lístku vždy první pracovní den v měsíci. V proceduře kontroluji zadaný lístek k víkendům a svátkům.

4.2 Optimalizace klientských databází

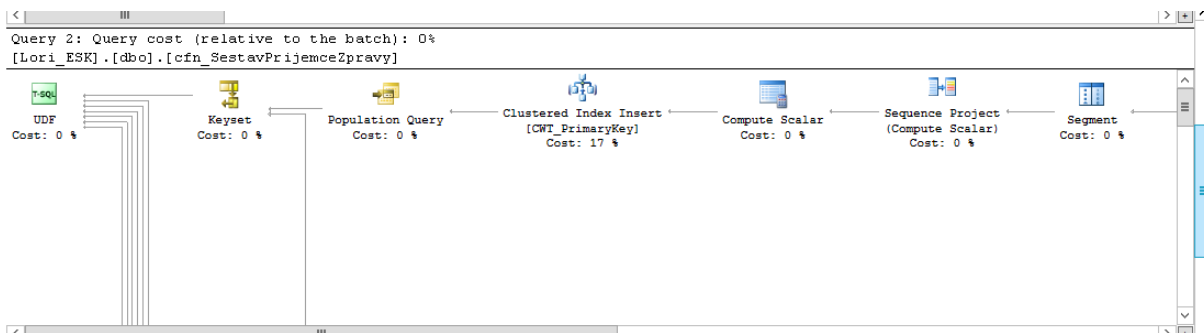
Před tím než jsem začal pracovat na optimalizaci databází, jsem od svého konzultanta dostal témata k nastudování. Například se jednalo o studium exekučního plánu včetně informací, na co se zaměřit.

Jedním z mých úkolů bylo snížit časovou náročnou dotazů na systémové databázi u dvou našich zákazníků. Postupoval jsem tak, že jsem nejprve potřeboval získat údaje k přihlášení ke klientskému serveru, které jsem si následně přidal do programu pro správu vzdálených připojení (Remote Desktop Connection Manager), přes který vždy na daný server přistupuji. Jelikož při práci na ostré databázi je třeba být obezřetný, tak jsem o této skutečnosti vždy obeznámil kolegu, jenž má daného zákazníka na starosti.

V případech, kdy jsem nepotřeboval pracovat na ostré databázi kvůli paměti typu cache, stáhl jsem si poslední zálohu ostré databáze, uložil na disk a obnovil. Jako nejlepší způsob mi přišlo uložení ze serveru na cloud a následné stažení. Přímé uložení trvá delší dobu.

Pro nalezení kritických míst jsem používal dva způsoby. Prvním z nich je použití SQL Server Profiler. Pomocí filtru jsem si nastavil, jaké typ výsledků chci vrátit. V mnoha případech mi tuto práci ulehčilo, že trace běží na serveru permanentně, a já si tak mohl projít záznam filtrovaných událostí například za posledních pět dní.

Ze začátku jsem měl problém s tím, že jsem nevěděl, na jaké události se zaměřit a jaké ignorovat. Například automatické úlohy, které probíhají v noci sice trvají dlouho, ale nijak neovlivní koncového uživatele, protože s databází v tu chvíli nikdo nepracuje. Dalším příkladem jsou dotazy pracující s datovým skladem. Po získání přehledu o tom, které události jsou pro mne relevantní jsem si vykopíroval dotaz a spustil jej spolu s exekučním plánem. Pokud mi to situace dovoľovala, tak jsem vždy používal možnost include actual execution plan místo display



Obrázek 1: Skalární funkce z dotazu

the estimated execution plan. Rozdíl je v tom, že první z nich uvádí přesný postup a hodnoty v jakém je dotaz vykonán a druhý uvádí pouhý odhad, který je často nepřesný. Nicméně výsledek display the estimated execution plan lze získat i bez spuštění dotazu a při některých dotazech, tak je jedinou možností.

Druhým způsobem bylo použití skriptu pro nalezení náročných dotazů, který jsem si podle potřeby upravoval pro dlouho trvající dotazy, dotazy s největším I/O a podobně. Tento skript extrahuje informace z DMV, který pracuje se statistikami týkajícími se dotazů uložených v paměti typu cache.

Jedním z dotazů, které jsem optimalizoval, byl dotaz, který trval přibližně 40 sekund a vracel 20000 záznamů. Jedná se o pohled v LORI. Problém byl v tom, že dotaz pracoval se dvěma skalárními funkcemi. Skalární funkce v dotazu jsou pro SQL Server něco jako černá skříňka do které nevidí. Tato funkce měla za úkol vyhledat a seřadit příjemce pomocí kurzoru na základě ID záznamů. Pokud to jde, tak je dobré se vždy kurzoru vyhnout. Obecně je SQL server optimalizován pro dotazy, nikoliv pro sekvenční procházení. Na druhou stranu jsou i případy, ve kterých kurzor lze obejít, ale dotaz se stane tak nepřehledný, že je lepší kurzor použít.

Tuto skalární funkci jsem přepsal pomocí konstrukce STUFF + FOR XML PATH přímo v dotazu. For XML path na konci dotazu vrací výsledky jako xml prvky. Funkce STUFF vkládá jeden řetězec do druhého a v tomto případě je použit pro odstranění první čárky. Časové zlepšení bylo více než o polovinu. U další skalární funkce GetUser, jejichž funkce byla dohledat uživatele, který provedl přidání či změnu záznamu jsem využil tabulkovou funkci z jiného databázového systému a nahradil ji v dotazu. Další změnu, kterou jsem provedl, bylo odstranění subselectů, které měly stejnou strukturu, akorát každý z nich se doptával na jiné sloupce. Konkrétně se jednalo o místo a čas nákladky. Subselecty jsem přidal pomocí outer apply do jednoho poddotazu.

Na systému LORI jsem příliš mnoho nových indexů nevytvářel. Často jsem do indexů přidával sloupce do include, tak aby nedocházelo k dohledání. Jedná se také o sloupce, které jsou v dotazech například obaleny funkcí ISNULL a nemůžou být z tohoto důvodu použity k index seek. V terminologii SQL se těmto predikátům říká sargable a non-sargable. Indexy jsem vytvářel především u nových struktur.

Od kolegy z jiného týmu jsem dostal za úkol vyřešit problém s vyčerpáním místa v databázi. Běžně takováto situace nenastává, ale tento zákazník disponoval pouze edicí Express ve verzi (z roku 2008), která umožňuje mít databázi pouze o velikosti 8 gb. V rychlosti jsem analyzoval velikost jednotlivých tabulek a dotazů, které nad nimi probíhají. U největší tabulky docházelo k tomu, že každý den se některé záznamy automaticky spouštěnou úlohou odmazaly a jiné přidaly. Tyto změny však byly v nepoměru a tudíž docházelo k permanentnímu nárůstu objemu dat.

S kombinací skriptů jsem zjistil, jaké je využití jednotlivých indexů tabulky a jak jsou velké. Tato tabulka jich obsahovala několik, a to především indexů vytvořených nástrojem Database Tuning Adviser. V rámci produktu LORI je použití tohoto nástroje zakázáno, nicméně někdy jsem se s ním setkal a ve většině případů jsou tyto indexy naprosto nesmyslné. Často obsahují několik sloupců v include. Zabrání se tak sice možnému dohledání (look up), ale nároky na správu takového indexu jsou nesrovnatelně vyšší. V dalších případech obsahují nadbytečné kombinace predikátů v indexech.

Vyřešil jsem to tedy takovým způsobem, že jsem si porovnal využití indexů a jejich velikost. Většinu z nich jsem pro nároky na jejich režii, velikost a nevyužitelnost odstranil.

Zpětně shledávám optimalizaci databází jako nejtěžší téma v průběhu praxe. Strávil jsem poměrně mnoho času analýzami dotazů a zvažováním, proč si SQL Server vybral právě tento exekuční plán, i když jsem jiný považoval za vhodnější. Jedná se například o situaci, kdy stejný dotaz s union all, byl pětkrát delší než s union, a to proto, že SQL Server z nějakého důvodu neprovedl exekuci dotazu paralelně. Obdobné situace byly během mé praxe poměrně časté.

4.3 Administrace klientských databází

V této oblasti mi byl zadán k vyřešení problém spojený s vyčerpáním diskového prostoru databázového serveru u dvou zákazníků. Mým úkolem bylo zjistit, proč k tomuto došlo a jak těmto problémům do budoucna zabránit, popřípadě jakou obecně zvolit zálohovací strategii. Veškeré kroky jsem konzultoval s firemním administrátorem, který má tyto věci na starosti.

Obdržel jsem vysvětlení, jak probíhá základní údržba databáze a jaké úlohy ji tvoří. První věcí, na níž jsem se zaměřil, byl častý výskyt shrink databáze. Tento algoritmus fyzicky zmenší databázový soubor, ale tím způsobem, že posouvá koncové stránky směrem dopředu, a až najde poslední volné místo, tak dojde k přesunutí. Zcela tak pozmění celou strukturu indexu a způsobí tím fragmentaci. V případě, že po této úloze neprovedeme rebuild databáze (čímž se v podstatě dostaneme do stavu před shrinkem), zůstane index pravděpodobně vysoce fragmentovaný a pozbyde na významu.

Zkontroloval jsem proto maintenance plány, abych zjistil v jakých krocích jsou úlohy prováděny. Automatický shrink musí být vždy zakázán. U shrinku vyvolaného například automaticky spouštěnou úlohou v noci to takový problém není, ale je nutno provést rebuild. Pro tento případ jsem neshledal žádná negativa, ale ani žádná pozitiva, čímž se nabízí otázka, zda má vůbec smysl shrink používat.

Další úlohou byla četnost zálohování, na což navazovala diskuze o tom, jak stará data jsou v případě poruchy ještě akceptovatelná. V tomto směru bylo na databázi většinou vše v pořádku. Všechny produkční databáze by měly být ve full recovery modelu. Standartem je provádění full backup jednou denně, popřípadě kombinace s diferenciální zálohou. Četnost zálohy transakčního logu jsem v jednom případě zmenšil ze tří hodin na půl hodiny. Dále jsem doporučil, aby zálohy byly vždy ukládány na jiný disk a minimalizovalo se tak riziko ztráty dat.

Poznatky, které jsem při administraci databází nabyt, jsem zpracoval do stručného dokumentu. Tento dokument slouží firemnímu administrátorovi, popřípadě člověku zodpovědnému za provoz databáze u zákazníka, jako návod k nastavení nově vytvořené databáze. Zároveň dokument lze využít jako určitou ochranu při řešení problému, vzniklého důsledkem špatného nastavení databáze externím správcem.

4.4 Databázové testování

Během praxe jsem dostal několik úkolů zaměřených na testování nově implementovaných řešení do databázového systému. Chci zde zmínit některá z těchto řešení a postup průběhu testování.

Spolu s dodáním samotného systému LORI byl ze strany zákazníka požadován modul pro výpočet parametrů cesty. Obecná funkčnost je, že po zadání jednotlivých souřadnicových bodů cesty je očekáváno vrácení parametrů, jako jsou časová délka cesty a vzdálenost. Pro výpočet těchto parametrů na základě souřadnic byla zakoupena licence německého dodavatele poskytujícího výpočetní server. Komunikaci s tímto serverem zajišťuje služba WCF, na níž jsou napojeny CLR.

Od programátora jsem dostal instalaci služby a CLR s jednotlivými třídami. Do databáze jsem si pomocí příkazu Create Assembly přidal tyto soubory a jim odpovídající procedury. Jelikož německá společnost dodává více produktů včetně webových aplikací, rozhodl jsem se testovat správnost výsledků vůči těmto aplikacím s předpokladem, že všechny používají stejné mapové podklady.

U prvních tří funkcí sloužících pro převod GPS souřadnic na adresu a pro výpočet trasy bez ohledu na parametry vozidla nebyl žádný problém. Potíže nastaly při testu čtvrté funkce, která slouží pro výpočet trasy s omezením, kterým může být výška či šířka vozidla, ale i další údaje. Webová aplikace Map and Guide nabízí mapu s bodovými značkami reprezentujícími konkrétní omezení. Mohl jsem si tak naplánovat trasu s omezením a vyzkoušet, jestli pro vozidlo s přesahujícími parametry dojde ke kalkulaci jiné trasy. Webová aplikace vždy alternativní trasu našla, zatímco výsledky ze serveru byly shodné pro různá vozidla na totožné trase s omezením.

Předpokládal jsem, že problém musí být na naší straně, jelikož při volání stejného požadavku by výsledky měly být shodné. Funkce jsou volány s parametrem GUID - unique identifier, který odkazuje na hodnoty ve vstupní tabulce. V případě volání procedury s parametry je princip obdobný, pouze s rozdílem, že hodnoty jsou vkládány do dvou vstupních tabulek a jedno volání se pozná na základě stejného GUID v těchto tabulkách.


```

67 -----
68 -- clrMG_VypocetTrasyOmezeni
69 -----
70 SET @GUID = '132B7BA8-FC8C-4F6C-8C1C-82C3FCC0AC7B'
71
72 EXECUTE clrMG_VypocetTrasyOmezeni @GUID, 1,
73         @dcKm OUTPUT,
74         @lCas OUTPUT,
75         @lError OUTPUT,
76         @szError OUTPUT,
77         'cz', 'CID', 1, 'Admin'
78
79 SELECT @dcKm, @lCas, @lError, @szError
80
81
82
83
84

```

100 %

Results Messages

	(No column name)	(No column name)	(No column name)	(No column name)
1	1.897	5	0	

Obrázek 2: Ukázka testování

Postupně jsem s programátorem upravoval dokument typu JSON, který se zasílá na server a testovali vrácená data. Nakonec jsem komunikoval s německým dodavatelem ohledně vyřešení problému. Ve výsledku bylo zjištěno, že stejný poskytovatel používá pro své mapové aplikace rozdílné mapové podklady. Zjednodušeně řečeno, některé mapy obsahují podstatně více informací oproti jiným.

Dalším projektem, který jsem testoval je nová mobilní aplikace Tablog, která rozšiřuje funkčnost LORI. Tato aplikace je určena řidičům pro vykazování úkolů, jakými mohou být např. nákladka, výkladka, škodní zápisy, pauzy a další. Aplikace komunikuje s jednotnou databází Microsoft Azure, z této se data přesouvají na základě rozlišení zákazníka do jeho metadatabáze a z metadatabáze do produkční. Ocenil jsem, že jsem se alespoň vzdáleně seznámil s Cloudem od Microsoftu a to především z důvodu, že nabízí velmi přehledné řešení u kterého se není potřeba starat o takové záležitosti, jakou je například škálovatelnost. V případě rozšíření si lze jednoduše dokoupit požadovaný výkon a není tak potřeba řešit konzultace. Zároveň jsou data bezpečně uložena na více kontinentech a minimalizuje se tak riziko ztráty.

Samotné testování probíhalo tak, že jsem zkoušel naplánovat různé přepravy v LORI, exportovat je do Tablogu a sledovat jejich funkčnost. V případě nejasností jsem komunikoval s vedoucím vývoje, zda se daný USE-CASE má chovat tímto způsobem, či nikoliv. Pro testování jsem používal poslední verzi Tablogu z HockeyApp a produkční verzi LORI, která umožňuje export do Tablogu. Mimo testování přeprav jsem testoval také nově naprogramované vlastnosti, jako například zobrazení přihlášení řidiče. V návaznosti na testování jsem absolvoval školení

nasazení Tablogu u nového zákazníka. Školení se týkalo popisu skriptů a nutných úprav, které v nich musí být provedeny.

4.5 Tvorba metodiky

Jelikož jsem v průběhu praxe narazil na mnoho nových témat, dostal jsem od svého konzultanta zadané, že vypracuji interní databázovou příručku. Ta bude následně sloužit těm zaměstnancům, kteří často pracují s SQL. Cílem příručky je poskytnout vysvětlení k nejčastějším problémům, se kterými se tito zaměstnanci při práci s SQL setkávají a popsat základní principy SQL Serveru. Může sloužit také jako zdroj odkazů, na nichž je možno nalézt detailnější popis řešené problematiky.

Metodiku jsem zpracovával tak, že při řešení určitého úkolu jsem si nejprve musel nastudovat související materiály, přičemž právě takto nabyté znalosti tvoří základ informací uváděných v metodice.

Některá témata bylo poměrně jednoduché popsat. Konkrétně se jedná o typy, strukturu indexů a podobně. Obecně se jedná o témata statické, které se nijak nemění v průběhu života databáze.

Komplikovanější naopak byla témata, která jsou svou povahou spíše individuální a jejich význam se vztahuje na konkrétní databázi. Jedná se například o případ, jak velký problém je fragmentace, popřípadě jestli vůbec problémem je.

Celkově je metodika rozpracována do zhruba třiceti stránek textu s příklady a odkazy na podrobnější popis.

Pořadí sloupců v indexu

- při vytvoření indexu s více sloupci je nutné vědět co nám pořadí říká
- př.index nad sloupci (a,b) znamená, že data ve sloupci b jsou organizována až na základě dat ve sloupci a. A proto při podmínce `WHERE b = “_”` nedojde vůbec k použití seek predikátu. Bude proveden scan, kde sloupec b bude běžným predikátem.
- obecně volíme pořadí tak, abychom eliminovali v dalším kroku co nejvíce výsledků(co nejvyšší selektivita)
- viz.[podrobnější vysvětlení](#)

Cover index a filtered index

- cover index je takový, který pokrývá všechny požadované sloupce z dotazu, tak aby nebylo potřeba dohledání(look-up) - viz.[cover index](#)
- filtered index obsahuje za svou deklaraci nějakou podmínku, která zmenšuje počet záznamů ze sloupce, který bude použit pro vytvoření indexu. Pokud např.sloupec obsahuje hodně null záznamů a my víme, že s nimi nikdy nebudeme pracovat, pak je vhodné toto do podmínky napsat a ušetříme si tak místo i výkon. - viz.[filtered index](#)

Obrázek 3: Výňatek z metodiky

5 Závěr

5.1 Využití a scházející znalosti

5.1.1 Využití teoretické a praktické znalosti získané v průběhu studia

V průběhu praxe jsem využil znalosti z mnoha předmětů, především z těch databázových. Jedná se o předměty Úvod do databázových systémů a Databázové a informační systémy, které mi daly praktické základy pro psaní dotazů a procedur. Předmět Administrace databázových systémů jsem měl bohužel až v letním semestru, kdy jsem probíranou látku z tohoto předmětu ve větší míře pro praxi nevyužil.

Dále jsem využil znalosti získané v předmětu Úvod do softwarového inženýrství, které jsem uplatnil při analýze zadaných úkolů. V neposlední řadě mi byly nápomocné také předměty jako Vývoj internetových aplikací, kde byly probírány základy různých technologií jako např. XML.

Znalosti z dalších předmětů jsem využil nepřímým způsobem. Látka probírána v předmětu Správa Windows systémů mi například pomohla získat přehled o oblasti, ve které jsem měl před praxí určité nejasnosti a v případě, kdy byla tato problematika řešena s kolegy, jsem již věděl, o co se jedná.

5.1.2 Scházející teoretické a praktické znalosti v průběhu odborné praxe

V první polovině praxe jsem nejvíce pociťoval absenci znalostí z oblasti administrace databáze. Musel jsem si informace k tomuto tématu dohledávat a zabralo mi čas, než jsem získal zodpovědnost tyto informace někomu dalšímu prezentovat, protože jsem neměl jistotu, že to tak skutečně je a mou navrhované úpravy nepřinesou nechtěné vlastnosti. Tímto jsem se ale naučil číst anglickou dokumentaci, zpracovat ji a být schopen ji vysvětlit.

Dále jsem věděl poměrně málo o metodách optimalizace databáze. Postupem času se mi však dařilo vytvářet vlastní metody, které jsem uplatnil ve vícero případech.

5.2 Zhodnocení a dosažené výsledky

Celkově absolvování bakalářské praxe hodnotím jako velmi přínosné. Cením si faktu, že jsem získal přehled o problémech spojených s vývojem vlastního software. Dále pro mě bylo naučné jednání s lidmi ať už na úrovni vývojového týmu nebo externích spolupracovníků v mateřském a cizím jazyce. V průběhu praxe mi bylo nabídnuto pokračování vzájemné spolupráce.

Zdroje

- [1] Microsoft, *Microsoft Development Network*, [ONLINE]
Dostupné z: <<https://msdn.microsoft.com/>>
- [2] McGehee, SQL Server Backup and Restore, Shawn McGehee, SQL Server Backup and Restore, Simple Talk Publishing, 2009. ISBN: 978-1-906434-74-8. [ONLINE]
Dostupné z: <<https://www.red-gate.com/simple-talk/RedGateBooks/>>
- [3] Microsoft, SQL Server Documentation, [ONLINE]
Dostupné z: <<https://docs.microsoft.com/en-us/sql/sql-server/>>
- [4] Brent Ozar, Články a komunita Brent Ozar Unlimited, [ONLINE]
Dostupné z: <<https://www.brentozar.com/blitz/>>
- [5] Sentry One, Plan Explorer User Guide, [ONLINE]
Dostupné z: <<https://www.sentryone.com/plan-explorer/>>
- [6] Oltis Group, Informační systémy pro dopravu a logistiku, [ONLINE]
Dostupné z: <<https://www.oltis.cz/>>