

XML v databázi Oracle

XML in Oracle database

Ondřej Chmelař



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Ondřej CHMELÁŘ
Osobní číslo: A10726
Studijní program: B 3902 Inženýrská informatika
Studijní obor: Informační a řídicí technologie

Téma práce: XML v databázi Oracle

Zásady pro vypracování:

1. Nastudujte funkce a způsoby práce s XML v databázi Oracle (schemata, export, import, procházení pomocí XPath výrazů).
2. Vymyslete vlastní příklady reprezentující práci s XML v databázi Oracle.
3. Vyzkoušejte funkčnost příkladů na školním serveru s databází Oracle.
4. Vytvořte českou uživatelskou příručku, která bude obsahovat popis práce s XML v databázi Oracle a její ukázky na Vámi vytvořených příkladech.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. MLÝNKOVÁ, Irena, et al. XML technologie : Principy a aplikace v praxi. Praha : Grada, 2008. 272 s. ISBN 978-80-247-2725-7.
2. YOUNG, Michael J.. XML : Krok za krokem. Brno : Computer Press, a. s., 2006. 471 s. ISBN 80-251-1070-2.
3. Oracle Database 10g Release 2 : XML DB [online]. Redwood Shores : Oracle Corporation, 2005 [cit. 2011-02-02]. Dostupné z WWW: http://download.oracle.com/otndocs/tech/xml/xmlldb/TWP_XML_DB_10gR2_long.pdf.
4. XML Path Language (XPath) : Version 1.0 [online]. [s.l.] : W3C, 1999 [cit. 2011-02-02]. Dostupné z WWW: <http://www.w3.org/TR/xpath>.
5. URMAN, Scott, et al. Oracle : Programování v PL/SQL. Brno : Computer Press, a. s., 2008. 720 s. ISBN 9788025118702.

Vedoucí bakalářské práce:

Ing. Kateřina Ježková
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

25. února 2011

Termín odevzdání bakalářské práce:

7. června 2011

Ve Zlíně dne 25. února 2011

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Cílem této bakalářské práce bylo vytvořit podporu výuky předmětu Teorie zpracování dat ve formě uživatelské příručky. Hlavním úkolem této uživatelské příručky je popis práce s jazykem XML (Rozšířitelný značkovací jazyk) a následné vysvětlení jak využívat XML při práci v databázovém systému Oracle. Postupně je vysvětlena syntaxe jazyka XML, tvorba dokumentů v tomto jazyce, schémat, dotazování se nad XML daty za pomoci jazyka XPath a ukázka jak využívat XML v databázi Oracle na jednotlivých příkladech. Ve vývojovém prostředí Oracle SQL developer byly vytvořeny a popsány příklady práce s XML v této databázi, které slouží pro pochopení, jak lze tento jazyk v databázi využívat.

Klíčová slova: XML, XML Schéma, XPath, databáze, element, atribut

ABSTRACT

The aim of this bachelor thesis was to create an electronic support to a subject called "Data Processing Theory." The goal was successfully realized with the support in a form of user guide. The purpose of this user guide is to describe how to work with the Extensible Markup Language (XML) as well as to explain the usage of XML while operating in the Oracle database system. The syntax of XML, building both documents and schemes in XML, querying data in XML using the XML Path language (XPath), and an illustration of the possibilities of XML in the Oracle database are all defined. In development environment of the Oracle SQL Developer, examples of work with XML in the Oracle database were created and described in order to understand the possibilities of using XML in the database.

Keywords: XML, XML scheme, XPath, database, element, attribute

PODĚKOVÁNÍ

Chtěl bych vyjádřit poděkování vedoucímu své bakalářské práce Ing. Kateřině Ježkové za uvedení do problému a za cenné rady a připomínky k obsahu práce. Také bych rád poděkoval mé rodině za morální a finanční podporu při studiu.

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST.....	10
1 DATABÁZOVÝ SYSTÉM ORACLE.....	11
1.1 HISTORIE A VÝVOJ ORACLE DB	11
1.2 DATABÁZE OBECNĚ.....	12
1.3 DATABÁZE V PODÁNÍ ORACLE	13
1.3.1 Databázový systém Oracle 10g Release 2	13
2 JAZYK XML A JEHO POUŽITÍ	15
2.1 REÁLNÉ VYUŽITÍ XML.....	15
2.2 FORMÁT XML.....	17
2.3 XML DOKUMENT	17
2.3.1 Syntaxe XML dokumentu	18
2.4 DEFINICE APLIKACE XML.....	19
2.5 XML SCHÉMA.....	20
2.5.1 DTD.....	20
2.5.2 XML Schema	21
2.6 XPATH.....	24
2.6.1 Datový model XPath	24
2.7 XSLT	25
2.8 XQUERY	26
2.9 APLIKAČNÍ ROZHRANÍ	27
2.9.1 Infoset.....	27
3 VYUŽITÍ XML V DATABÁZÍCH.....	29
3.1 ULOŽENÍ XML DAT V RELAČNÍ DATABÁZI.....	29
3.2 SQL/XML	30
3.3 XML V ORACLE DATABASE	30
3.3.1 Implementace XML na platformě Oracle Database 10g.....	31
3.3.2 Administrace XML na platformě Oracle Database 10g.....	32

II PRAKTICKÁ ČÁST	33
4 VÝVOJOVÉ PROSTŘEDÍ PRO TVORBU PŘÍKLADŮ	34
5 UŽIVATELSKÁ PŘÍRUČKA A PŘÍKLADY	37
5.1 UŽIVATELSKÁ PŘÍRUČKA	37
5.2 PŘÍKLADY V ORACLE	39
5.2.1 Příklady na tvorbu XML schéma	40
5.2.2 Příklady výběru dat pomocí XPath	41
5.2.3 Příklady na tvorbu XML z dat tabulky	42
5.2.4 Ostatní příklady v databázi Oracle	44
ZÁVĚR	46
LAST WORDS.....	CHYBA! ZÁLOŽKA NENÍ DEFINOVÁNA.
SEZNAM POUŽITÉ LITERATURY.....	50
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	51
SEZNAM OBRÁZKŮ	52
SEZNAM PŘÍLOH.....	53

ÚVOD

Dnešní doba je dobou vyspělých informačních technologií, které se rozvíjejí obrovskou rychlostí a bezprostředně ovlivňují naše životy. K době informačních technologií, kdy potřebujeme zpracovávat velká kvanta informací, neodmyslitelně patří elektronické databázové systémy. Ačkoliv si to běžný člověk možná tolik neuvědomuje, tak jsou naše životy s databázemi spojeny takřka na každém kroku. Sofistikované databázové systémy v současné době zajišťují a zabezpečují převody peněžních prostředků, je díky nim možno platit prostřednictvím platebních karet, využívat internetového bankovníctví, nakupovat na internetu, rezervovat si online letenky, ubytování v hotelu nebo nesčetné množství dalších věcí. Databázové systémy uchovávají rok od roku větší množství dat o světě kolem nás i o nás samotných. Jednou z nejlepších firem na trhu poskytující své služby v oblasti databází, tedy vývoj, prodej a údržbu databázových systémů je firma Oracle Corporation.

Databáze jednoduše slouží k usnadnění správy dat a k efektivnímu nakládání a práci s těmito daty. V případě, že je databáze navrhnutá kvalitně, tak nám umožňuje mezi všemi daty uloženými v databázi velmi jednoduché a rychlé vyhledání námi požadované informace a to i ve velmi rozsáhlých databázích. Stejně jako patentované databázové formáty se nám v dnešní době nabízí možnost využít v databázích možnosti značkovacího jazyka XML (Rozšiřitelný značkovací jazyk), který nám například umožňuje označení jednotlivých polí uvnitř databázového systému (např. tedy označení jmen, adres, telefonů v databázi adres). V případě označení každé části informace se nám naskýtá možnost zobrazení dat různými způsoby, dále jejich řazení, třídění, filtrování a jakékoliv jiné možnosti zpracování.

Tato bakalářská práce je zaměřena na využití technologií XML v databázovém systému firmy Oracle a měla by především sloužit k usnadnění a lepšímu pochopení jak využívat XML technologie v databázovém systému Oracle. Praktickým výstupem této bakalářské práce je uživatelská příručka, ve které je v jednotlivých kapitolách postupně vysvětlena základní tvorba jednoduchých XML dokumentů až po konkrétní využití XML v databázích názorně ukázaného na vytvořených příkladech. V příručce nejprve vysvětlena syntaxe jazyka XML, dále tvorba XML schémat, dotazování se nad XML daty za pomoci jazyka XPath a zakončení ve formě příkladů v databázi Oracle. Bakalářská práce i její příloha byla napsána ve snaze co nejsrozumitelněji a co nejjednodušeji vysvětlit využití XML v databázi Oracle.

I. TEORETICKÁ ČÁST

1 DATABÁZOVÝ SYSTÉM ORACLE

Oracle je systém řízení báze dat (Oracle database management system - DBMS), moderní multiplatformní databázový systém s velice pokročilými možnostmi zpracování dat, vysokým výkonem a snadnou škálovatelností. Databázový systém Oracle je vyvíjen firmou Oracle Corporation. Oracle Corporation je největší společností na světě, která dodává podnikový software firmám a organizacím všech velikostí. Tato společnost nabízí kromě podnikových aplikací a nástrojů na jejich vývoj také databázi, aplikační server a nástroje pro podnikovou spolupráci. [1]

1.1 Historie a vývoj Oracle DB

Historie tohoto databázového systému je tak bohatá, že bychom jí mohli věnovat vlastní publikaci. Pro náhled do historie si vystačíme se zásadními milníky na cestě k největší verzi databázového systému Oracle 11g. Tato cesta trvala více než třicet let, což je zárukou zkušeností a propracovanosti databázové architektury. [1]

V roce 1977 Larry Ellison, Bob Miner a Ed Oates založili firmu SDL (Software Development Laboratory), která se zabývala vývojem RDBS (Relational Database Management System) podle teorie doktora E. F. Codd pod názvem Oracle. O rok později se společnost přejmenovala na Relational Software Inc. Byla vyvinuta verze Oracle V1 pro počítač PDP-11 a pracující pod operačním systémem RSX. Tato verze byla napsána ve strojovém jazyce a nebyla nikdy oficiálně uvedena na trh. V roce 1980 společnost RSI představila komerční verzi Oracle – Oracle V2. Tato verze byla také napsána ve strojovém jazyce počítače PDP-11, ale pracovala pod operačním systémem Vax/VMS. O dva roky později byla představena verze Oracle V3. Oracle se stal prvním DBMS pro sálové počítače, minipočítače i počítače PC. V této verzi je zabudována podpora transakčního zpracování. Programový kód pro stranu serveru byl napsán v jazyce C. V roce 1983, se společnost přejmenovala na Oracle Corporation. O rok později byla představena verze Oracle V4. Oracle je dostupný pro více platforem, byla umožněna komunikace mezi PC a serverem. [1]

Verze Oracle V5 vydaná v roce 1985 byla postavena na architektuře klient – server, podpora VAX clusterů a distribuovaného zpracování. Rok 1988 přinesl Oracle V6 a o rok později vyšla verze Oracle 6.2, která využívala Oracle Parallel Server. V roce 1991 se podařilo na paralelně pracujících strojích dosáhnout výkonnosti 1000 TPS (Transactions Per Second). O rok později byla vydána verze Oracle 7 pro platformu Unix a pár let poté

pro platformu PC. Roku 1997 vyšla verze Oracle 8, která podporovala více uživatelů, větší množství dat a vyšší dostupnost. O dva roky později byla do verze Oracle8 integrována Java a verze byla pojmenována Oracle 8i. V roce 2000 se Oracle zařadil mezi ERP (Enterprise Resource Planning) aplikace a vydal verzi Oracle 8i Release2. Verze Oracle 9i byla vydána o rok později a verze Oracle 10g byla vydána v roce 2003. V současnosti se využívají verze Oracle 10g Release 2 a verze Oracle 11g.

Vedoucí pozici na trhu si společnost Oracle zasloužila řadou technických inovací a prvenství. Společnost Oracle byla jednou z prvních společností, které poskytly své obchodní aplikace prostřednictvím internetu, a také jednou z prvních, které zajistily, že jejich software je schopen spolupracovat. Výsledkem je minimalizace nákladů na integraci a současné zvýšení spolehlivosti. Společnost Oracle se rovněž zaměřila na zjednodušení zákaznických systémů a poskytování komplexních řešení, které uživatelům umožňují získat lepší přístup k datům za nižší náklady. [1]

1.2 Databáze obecně

Databáze je kolekce dat na disku, která je fyzicky uložena v jednom, nebo více souborech na databázovém serveru a která obsahuje souvztažné informace. Databáze sestává z různých fyzických a logických struktur. Nejdůležitější logickou strukturou databáze je tabulka. Tabulka je složena z řádků a sloupců, které obsahují souvztažná data. Aby databáze mohla uchovávat data, musí obsahovat alespoň tabulky. [7]

Databáze také poskytuje zabezpečení, která brání neautorizovanému přístupu k datům. Soubory, ze kterých je databáze složena, se dělí do dvou kategorií: databázové soubory a nedatabázové soubory. Rozdíl je v datech, která jsou v souborech uložena. Databázové soubory obsahují databázová data a metadata, nedatabázové soubory obsahují inicializační parametry, protokolovací informace atd. Databázové soubory jsou pro fungování databáze klíčové. [7]

1.3 Databáze v podání Oracle

Oracle je relační systém řízení báze dat. Data jsou zde ukládány do tabulek, které jsou definovány sloupci. Ty mají různé názvy a u každého sloupce je definován typ dat, jaká v něm budou obsažena. Data jsou uložena v jednotlivých řádcích dané tabulky. Slovo relační u těchto databází znamená, že vazby, mezi tabulkami se zde vytvářejí tak, že v tabulkách, které chceme propojit, je v každé jeden sloupec, který nabývá stejných informačních hodnot jako jeden ze sloupců druhé tabulky. [1]

Oracle obecně nepodporuje přístup k datům jen podle relačního modelu, ale podporuje i objektově orientované struktury (např. abstraktní datové typy a metody). Objekty lze mezi sebou provázat nebo mohou obsahovat další objekty. Veškerá data jsou interními strukturami databáze logicky mapována na soubory. Tato logická rozdělení jsou označovány jako tabulkové prostory (tablespace). Každý tabulkový prostor se skládá z jednoho, nebo více souborů na disku (tzv. datových souborů). Datový soubor může patřit pouze k jednomu tabulkovému prostoru. Po přidání datového souboru k tabulkovému prostoru ho není možné z tabulkového prostoru odstranit ani přiřadit k jinému tabulkovému prostoru. Díky tomu, že je možné uložit datové soubory tabulkových prostorů na samostatné disky, je možné optimalizovat zpracování vstupních a výstupních požadavků na databázi. [1]

1.3.1 Databázový systém Oracle 10g Release 2

V tomto kroku bych rád přiblížil nějaké informace o databázovém systému Oracle 10g Release 2, protože tento systém je nainstalován na školním serveru a byly na něm odzkoušeny příklady, které jsou uvedeny v příručce. Tato databáze je též využívána studenty pro výuku předmětů spojených s tvorbou databází v databázovém systému Oracle.

Tato verze rozšiřuje možnosti zpracování ve výpočetních sítích a zvyšuje výkon, zlepšuje dostupnost aplikací a zjednodušuje správu. Verze 2 mj. zvyšuje škálovatelnost technologie RAC, takže nyní může cluster tvořit až 100 serverů. Zlepšené vyvažování zátěže v RAC dovoluje rychleji reagovat na měnící se vzorce využití serverů v clusteru. Tato verze obsahuje aplikační rozhraní pro clusterové služby. Rozšířené funkce Oracle Automatic Storage Management zjednodušují sdílení úložných prostředků v prostředí sítě. K dispozici jsou metody třídění dat a nové automatizované funkce Oracle Data Guard umožňující během několika sekund od poruchy přepnout provoz na záložní databázi, a to dokonce i v prostředí bez obsluhy. Verze Oracle Database 10g Release 2 umožňuje šifrovat

databázová data zálohovaná na disku nebo na pásce. Oracle Secure Backup je nový automatizovaný systém zálohování na pásky a poskytuje nejvyšší výkon zálohování databází Oracle. Lze jej použít nejen s platformou Oracle Database 10g, ale také se staršími verzemi, například Oracle 9i. Oracle Recovery Manager byl oproti starším verzím zdokonalen o šifrování zálohovaných dat. Několik nových vlastností Oracle Database 10g Release 2 zlepšuje podporu pro vývojáře aplikací, novinkami oproti předchozím verzím je například podpora standardu W3C XML Query pro přístup k datům v jazyce XML či vylepšená podpora systému Microsoft Windows prostřednictvím uložených procedur, které jsou implementovány v jazyce CLR (Common Language Runtime). Další novinkou je zdokonalená konzola Oracle Enterprise Manager Database Control, která zjednodušuje sledování topologie a propojení clusteru. [6]

2 JAZYK XML A JEHO POUŽITÍ

Formát XML definovalo konsorcium W3C jako formát pro přenos obecných dokumentů a dat. XML je zkratka pro eXtensible Markup Language, tj. rozšiřitelný značkovací jazyk. Návrh XML vychází ze staršího a obecnějšího standardu SGML (Standard Generalized Markup Language). Poznamenejme, že ze standardu SGML vycházel i formát dokumentů HTML (Hyper-Text Markup Language). Sada značek formátu HTML je pevná a slouží k vyjádření prezenční podoby dokumentu. Naproti tomu v XML sada značek pevná není, ale může být definována pro různé sady dokumentů různě. Definice sady značek může být součástí definice XML dokumentu, může být specifikována odkazem, nebo může být dohodnuta předem. [4]

Nad XML daty se můžeme dotazovat za pomoci dotazovacích jazyků, mezi které lze zařadit XPath, XPointer a XLink. Účelem těchto jazyků však není pouze dotazování, nýbrž také možnost vytvářet pohledy nad XML daty, transformace XML dat, případně možnost specifikovat množinu XML dat, která má být aktualizována. [4]

Po roce 1998 se objevila řada jazyků, které rozvíjely nové koncepce dotazování. Patří sem např. jazyky XML-QL, XQL a mnoho dalších. Vývoj v konsorciu W3C se posléze orientoval na jazyky XSLT 1.0, XSLT 2.0, XPath 1.0, XPath 2.0 a XQuery 1.0. XSLT byl vytvořen přímo pro transformace XML dat. Využívá výrazů v XPath a taktéž samotný jazyk XML. Vhodnější pro dotazování je ale jazyk XQuery, především díky své syntaxi orientované více uživatelsky. Jazyk XPath 1.0, resp. 2.0, je koncipován tak, že tvoří vlastní podmnožinu jazyka XQuery. [4]

2.1 Reálné využití XML

Ačkoli se XML může zdát jako zajímavý nápad, tak si někdo může lámat hlavu s tím, jak vlastně XML využít v reálném světě. V této části se dozvíte něco o praktickém využití XML. Popíšeme si, jakým způsobem se v současnosti XML používá a jak se jej různé společnosti chystají využívat. [3]

Příklady využití XML [3]:

- Ukládání do databází. Touto problematikou se budeme zabývat detailněji v kapitole č. 3 této práce.
- Strukturované dokumenty. Stromová struktura dokumentu XML je ideální pro práci se strukturovanými dokumenty, jako jsou romány, knihy

populárně naučné literatury (literatura faktů) a divadelních her. Např. můžete XML využít pro označení rolí v jednotlivých výstupech, scén, dialogů, pro pokyny k řízení jeviště atd. XML můžeme rovněž použít k označení způsobu zobrazení a tisku jednotlivých částí dokumentu (řádkování, styly odstavců), ke generování obsahu, přehledů nebo výtahů a ke zpracování informací jiným způsobem.

- Ukládání vektorové grafiky.
- Popis multimediálních prezentací.
- Vytváření hlasových rozhraní pro internetové programy.
- Definování kanálů.
- Označení softwarových balíčků a jejich vzájemné závislosti.
- Komunikace mezi programem prostřednictvím webu s využitím zpráv založených na XML.
- Výměna informací ve finančnictví.
- Vytváření, řízení a používání složitých digitálních formulářů pro internetové obchodní transakce.
- Výměna pracovních postupů.
- Výměna informací mezi tiskárnami a jejich zákazníky.
- Formátování matematických vzorců a vědeckých výpočtů na webu.
- Popis molekulárních struktur.
- Formátování tezí a disertací pro elektronické podání.
- Ukládání technologických informací a klasifikace liturgických textů.

To bylo jen několik konkrétních příkladů z určitých oblastí pro představu, kde všude se dá XML využít. XML ovšem nachází uplatnění ještě v mnoha dalších oblastech.

2.2 Formát XML

XML je formát pro reprezentaci a přenos obecných dokumentů. Při návrhu XML se autoři z konsorcia W3C řídili následujícími principy [4]:

- formát XML musí být použitelný v rámci internetu,
- formát XML by měl podporovat širokou škálu aplikací,
- formát XML musí být kompatibilní s formátem SGML,
- musí být snadné vytvářet programy, které manipulují s dokumenty v XML,
- množství variant XML by mělo být minimální a
- XML dokumenty by měly být čitelné a pochopitelné i pro člověka.

Na základě těchto principů navrhli definici XML, která zahrnuje dvě části (4):

- definici, co je to XML dokument,
- definici programů, které zpracovávají XML dokumenty – XML procesorů

2.3 XML Dokument

XML dokument je určitým způsobem uspořádaná posloupnost znaků jisté abecedy. Implicitně se předpokládá Unicode (kód ISO/IEC 10646). Fyzicky se XML dokument skládá z posloupnosti prvků nazývaných entity. Fyzická entita ještě není logický element dokumentu. Z hlediska procesoru XML může každá fyzická entita obsahovat buď rozpoznatelná data, nebo nerozpoznatelná data. Nerozpoznatelná data mohou být textová, nebo binární, která se buď procesoru nepodaří interpretovat jako znaky, nebo se jedná o data pro jinou aplikaci. Rozpoznatelná data jsou sestavena ze znaků zvolené abecedy a představují buď znaková data nebo značky (markups). Značky vyznačují logickou strukturu dokumentu a tím i jeho rozložení (layout). Dokument začíná entitou nazývanou kořen (root entity). [4]

Logicky se XML dokument skládá z prologu, deklarací, elementů, komentářů a instrukcí pro zpracování jinými aplikacemi. Logické elementy jsou v dokumentu vyznačeny značkami. Sada značek je obvykle libovolná, může být ale předepsána deklaracemi. Deklarace nejsou povinné, pokud ale chceme strukturu dokumentu kontrolovat, musím ji deklaracemi předepsat. XML dokument je dobře vytvořen, pokud všechny rozpoznatelné entity v dokumentu jsou správně vytvořeny a navíc, všechny rozpoznatelné entity, na které

existují v dokumentu odkazy, jsou rovněž dobře vytvořeny. Každá dvojice závorek musí být korektně spárována v rámci elementu a tyto dvojce musí být dobře vnořeny do sebe – závorky se nesmí křížit. Z toho plyne, že dobře vytvořený XML dokument musí mít stromovou strukturu. [2,4]

2.3.1 Syntaxe XML dokumentu

Každý dobře vytvořený XML dokument musí začínat prologem, za kterým následuje kořenový element dokumentu. Prologem se stanoví verze dokumentu XML, kódování dokumentu a případné požadavky na strukturu dokumentu. Každý XML dokument obsahuje jeden či více elementů. Vždy ovšem obsahuje právě jeden kořenový element, jehož žádná část není obsažena v žádném jiném elementu. Součástí dokumentu může být i sada komentářů a instancí pro jiné aplikace. Prolog dokumentu začíná deklarací verze XML. [4]

XML dokument je text, vždy Unicode, v Česku obvykle kódovaný jako UTF-8, ale jsou přípustná i jiná kódování. Na rozdíl od např. HTML, efektivita XML je silně závislá na struktuře, obsahu a integritě. Aby byl dokument považován za správně strukturovaný („well-formed“), musí mít nejméně následující vlastnosti [9]:

- Musí mít právě jeden kořenový (root) element.
- Neprázdné elementy musí být ohraničeny startovací a ukončovací značkou. Prázdné elementy mohou být označeny tagem „prázdný element“.
- Všechny hodnoty atributů musí být uzavřeny v uvozovkách – jednoduchých (') nebo dvojitých ("), ale jednoduchá uvozovka musí být uzavřena jednoduchou a dvojitá dvojitou. Opačný pár uvozovek může být použit uvnitř hodnot.
- Elementy mohou být vnořeny, ale nemohou se překrývat; to znamená, že každý (ne kořenový) element musí být celý obsažen v jiném elementu.
- Jména elementů v XML rozlišují malá a velká písmena

Jednoduchý recept v XML jako příklad by mohl vypadat takto [9]:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Poznamka je nutné přidat více receptů. -->
<recept jméno="chleba" čas_přípravy="5 minut" čas_vaření="3 hodiny">
  <titulek>Jednoduchý chleba</titulek>
  <přísada množství="3" jednotka="šálky">Mouka</přísada>
  <přísada množství="0,25" jednotka="unce">Kvasnice</přísada>
  <přísada množství="1,5" jednotka="šálku">Horká voda</přísada>
  <přísada množství="1" jednotka="kávová lžička">Sůl</přísada>
  <instrukce>
    <krok>Smíchejte všechny přísady dohromady a dobře prohnětte.</krok>
    <krok>Zakryjte tkaninou a nechejte hodinu v teplé místnosti.</krok>
    <krok>Znovu prohnětte, umístěte na plech a pečte v troubě.</krok>
  </instrukce>
</recept>
```

2.4 Definice aplikace XML

Aplikaci XML můžeme většinou definovat vytvořením tzv. DTD (Document Type Definion – definice typu dokumentu), který je většinou součástí dokumentu XML, nebo ji můžeme definovat pomocí tzv. XML schématu, které je umístěno v samostatném souboru. DTD i schéma XML definují a vyjmenovávají elementy, které mohou být použity v dokumentu, stanovují pořadí elementů, použití atributů a další rysy dokumentu. Použití konkrétní aplikace XML znamená většinou buď vložit DTD do dokumentu XML nebo použít schéma XML při zpracování dokumentu. Pokud chceme, aby náš dokument vyhovoval standardní aplikaci XML, musíme použít DTD, nebo schéma XML, vymezující elementy, atributy a strukturu dokumentu. [3]

Důležitou předností použití standardní aplikace XML při tvorbě dokumentů je, že aplikace podporuje konzistenci, jak uvnitř jednoho dokumentu, tak i mezi různými dokumenty stejného typu. Pro skupinu dokumentů, které se zpracovávají a zobrazují s použitím specifických softwarových nástrojů, které jsou navrženy pro daný typ dokumentu, je konzistence elementů, atributů a struktury dokumentu nezbytná. [3]

2.5 XML Schéma

Hlavní síla XML jazyka spočívá především v možnosti specifikovat sady přípustných značek i jejich konkrétní strukturu a ty přiřazovat jednotlivým XML dokumentům. Pokud daný dokument obsahuje pouze značky z konkrétní definované sady s odpovídající strukturou, je vůči této sadě validní, nebo říkáme, že se jedná o její instanci. Popis této sady nazýváme XML Schématem daného XML dokumentu. [4]

XML schéma je dokument, který definuje obsah a strukturu třídy dokumentů XML. Schéma XML může např. definovat obsah a strukturu dokumentů XML, které slouží pro uchovávání katalogu knih. Konkrétně schéma XML popisuje elementy a atributy, které mohou být obsaženy v konkrétním dokumentu a způsob, jakým mají být elementy rozvrženy v hierarchické struktuře dokumentu. [3]

XML schéma tedy obsahuje popis přípustné struktury XML dokumentů. Pro účely definice XML schématu je přímo součástí specifikace jazyka XML jazyk DTD. Tento jazyk má obvykle postačující vyjadřovací sílu a je oblíben i pro svoji jednoduchost. Pro složitější aplikace, nebo při potřebě vyjádřit přípustnou strukturu XML dokumentů přesněji, je již ale jazyk DTD nedostačující. Z toho důvodu vytvořilo konsorcium W3C specifikaci speciálního jazyka nazvaného XML Schema. [4]

2.5.1 DTD

DTD (Document Type Definition, česky Definice typu dokumentu) je jazyk pro popis struktury XML případně SGML dokumentu. Omezuje množinu přípustných dokumentů spadajících do daného typu nebo třídy. Struktura třídy nebo typu dokumentu je v DTD popsána pomocí popisu jednotlivých značek (nebo též elementů) a atributů. Popisuje, jak mohou být značky navzájem uspořádány a vnořeny. Vymezuje atributy pro každou značku a typ těchto atributů. DTD je poměrně starý a málo expresivní jazyk. Jeho další nevýhoda je, že DTD samotný není XML soubor [9].

V definici dokumentu zapsaného pomocí DTD se mohou vyskytovat [4]:

- deklarace typu elementu,
- deklarace seznamu atributů,
- deklarace entity,
- deklarace notace.

Ukázka velmi jednoduchého DTD dokumentu (priklad.dtd):

```
<!ELEMENT clovek (jmeno, adresa*)>
  <!ELEMENT jmeno (#PCDATA)>
  <!ELEMENT adresa (ulice?, cislo?, mesto)>
    <!ELEMENT ulice (#PCDATA)>
    <!ELEMENT cislo (#PCDATA)>
    <!ELEMENT mesto (#PCDATA)>
```

Tento DTD dokument definuje třídu XML dokumentů, které mají kořenovou značku <člověk>, obsahující právě jednu značku <jméno> a žádnou nebo více značek <adresa> a tak dále. Proto například tento XML dokument patří do takto vymezené třídy XML dokumentů (říkáme také, že je validní proti výše uvedenému DTD):

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE clovek SYSTEM "priklad.dtd">
<clovek>
  <jmeno>Ondřej Chmelař</jmeno>
  <adresa>
    <ulice>Nad Stráněmi</ulice>
    <cislo>45</cislo>
    <mesto>Zlín</mesto>
  </adresa>
</clovek>
```

Zde vložená ukázka jednoduchého DTD dokumentu je jen na úvod pro představu jak takový DTD dokument může vypadat, bližším a přesnějším popisem, co které části kódu znamenají se zabývám v příložené příloze (příručce pro studenty).

2.5.2 XML Schema

Speciální jazyk XML Schema byl specifikován skupinou W3C a obsahuje velké množství nových konstruktů a jeho vyjadřovací síla je podstatně větší než u DTD. Objektově orientované rysy u tohoto jazyka (jako je např. dědičnost, substituovatelnost, oddělení typu od jeho výskytu atd.) navíc přináší mnohá zjednodušení při modelování reality, pro něž je objektově orientovaný přístup přirozený. Díky tomu, že jazyk XML Schema vychází z jazyka XML (tj. každé XML schéma popsané v jazyce XML Schema je současně XML dokumentem), je možné při zpracování XML schémat využít všech nástrojů vytvořených pro práci s XML dokumenty. (Abychom nemuseli používat poměrně krkolomný výraz

„XML schéma popsané v jazyce XHM Schema“, vžilo se později označení XSD, tj. zkratka anglického termínu „XML Schema definition“). [4]

Ukázka dokumentu XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Zamestnanec">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="jmeno" type="xs:string"/>
        <xs:element name="prijmeni" type="xs:string"/>
        <xs:element name="pocetEditaci"
type="xs:integer"/>
      </xs:sequence>
      <xs:attribute name="uid" type="xs:string"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Ukázka připojení XSD k XML dokumentu:

```
<?xml version="1.0" encoding="UTF-8"?>
<Zamestnanec xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="Schema.xsd"
  uid="Novak">
  <jmeno>Jirka</jmeno>
  <prijmeni>Novák</prijmeni>
  <pocetEditaci>152</pocetEditaci>
</Zamestnanec>
```

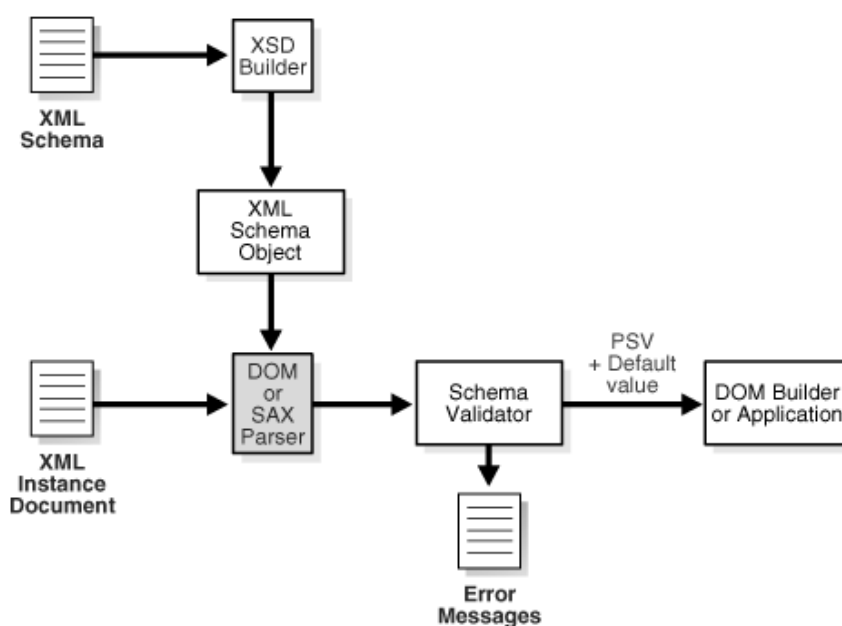
Přínosy jazyka XML Schema

Stručný přehled nejdůležitějších a nejzajímavějších výhod jazyka XML Schema [4]:

- Na rozdíl od jazyka DTD nevyžaduje speciální syntaxi – XML schémata v jazyce XML Schema jsou opět XML dokumenty s pevně danou strukturou. Uživatel tedy není nucen umět dva různé jazyky a pro zpracování XML schémat může využít všech nástrojů, které již existují pro práci s XML dokumenty.
- Má silnou podporu datových typů. Obsahuje rozsáhlou sadu vestavěných typů (např. string, boolean, date) a současně umožňuje specifikovat vlastní uživatelsky

definované typy. Užitečným prvkem jsou bezesporu také vícehodnotové datové typy, v podstatě jakási obdoba polí hodnot.

- Umožňuje přesně, tj. v rámci daného rozmezí, vyjádřit přípustné počty výskytu elementů a rámci rodičovského elementu.
- Při modelování je možné opakovaně využívat již definované prvky (datové typy, množiny elementů, množiny atributů atd.), čímž se specifikace schématu značně zjednodušuje a urychluje.
- Striktně rozlišuje mezi posloupností a množinou elementů, tj. umožňuje snadno vyjádřit libovolné pořadí podelementů v rámci rodičovského elementu.
- Při modelování reality je využíváno mnoho objektově orientovaných prvků, které jsou pro tyto účely praktické a ověřené.
- Umožňuje definovat elementy se stejným názvem, ale s různým obsahem.
- Umožňuje definovat nejen elementy s prázdným obsahem, ale i elementy, které se mohou vyskytovat bez definovaného obsahu.
- Umožňuje specifikovat unikátnost obsahu elementu, hodnoty atributu, nebo jejich kombinace v rámci požadované části XML dokumentu.
- Zachovává většinu prvků z jazyka DTD.
- Tutéž věc je možné definovat několika různými způsoby, tj. uživatel si může zvolit nástroj jemu blízký.



Obrázek 1 - Procesor XML Schema [5]

2.6 XPATH

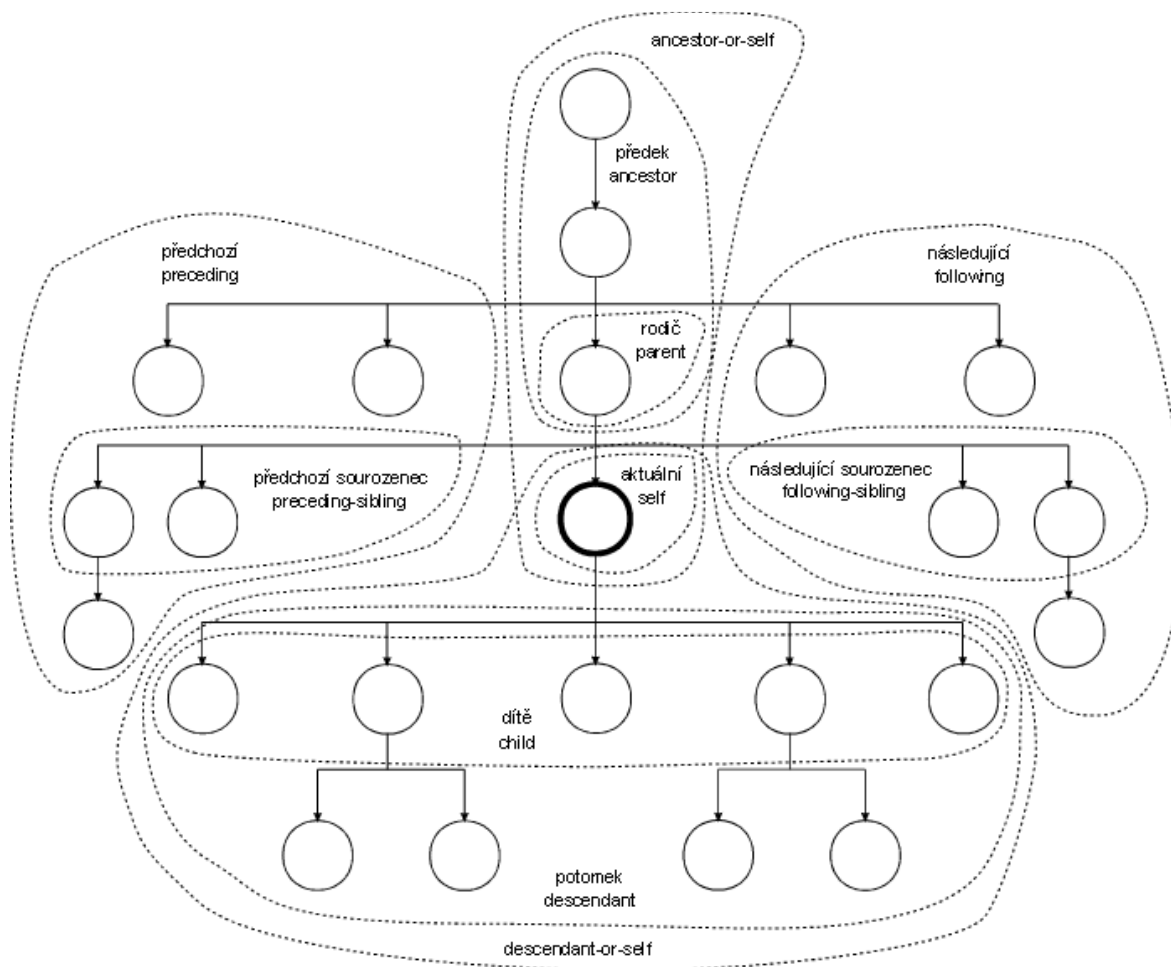
Jazyk XPath 1.0 je prostředek, který poskytuje možnost vybírat uzly ze stromové reprezentace XML dat. Je charakterizován následujícími rysy [4]:

- je jazykem použitelným pro výběr částí XML dokumentu (např. pros vstup do XSLT),
- způsob jakým popisuje cesty k elementům, je podobný způsobu, jakým operační systém počítače popisuje cesty k souborům,
- notace jazyka není založena na notaci jazyka XML,
- na XPath jsou založeny jazyky XSLT, XPointer a XQuery.

2.6.1 Datový model XPath

Pro jakýkoliv dotazovací jazyk je typické, že pracuje s nějakým modelem dat. Datový model XPath je stromově orientovaný. Na rozdíl od modelu Infoset v něm existuje sedm uzlů [4]:

- uzel označující kořen dokumentu,
- uzel elementu,
- textový uzel,
- uzel atributu,
- uzel komentáře,
- uzel instrukce pro zpracování,
- uzel jmenného prostoru.



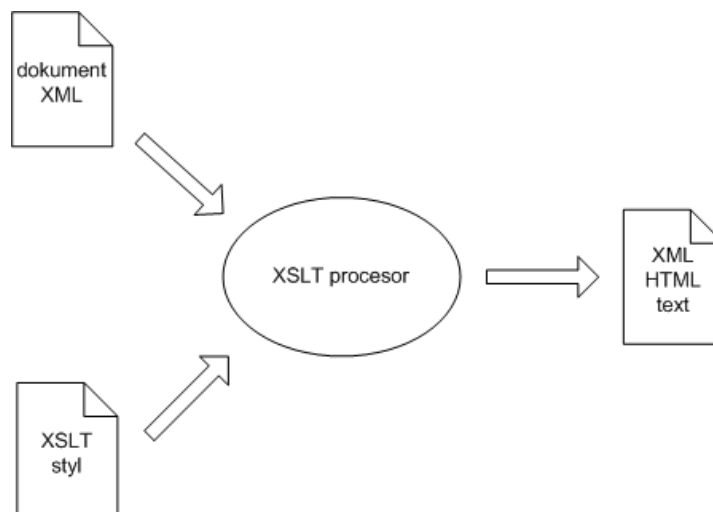
Obrázek 2 – Vztahy mezi uzly ve stromové reprezentaci XML dokumentu. [8]

2.7 XSLT

Konroscium W3C vytvořilo návrh specializovaného nástroje nazývaného XSL (eXtensible Stylesheet Language), kterým je možné měnit strukturu dokumentů XML. Jak je patrné z názvu, XSL byl v (v původním návrhu) určen pro formátování XML dokumentů (Stylesheet = formátovací sada), tj. pro jejich vizualizaci. Dnes už je však XSL nástrojem pro popis obecných transformací XML dokumentů do libovolného formátu. Lze jej tedy použít nejen pro prezentaci, ale i k nejrůznějším převodům XML dokumentů, či k extrakci informací z nich. Můžeme jej ale považovat i za dotazovací jazyk, který pro daný transformační předpis a vstupní dokument vrátí výstupní dokument – výsledek dotazu. [4]

Transformace XML dokumentů v XSL je definována pomocí sady pravidel v jazyce XSLT (eXtensible Stylesheet Language Transformations). Aplikací těchto pravidel se XML dokument transformuje na dokument používající jiný slovník, který si můžeme představit jako slovník sémantický. XSL dokument je v podstatě program v jazyce XLST. XLST není

ovšem klasický programovací jazyk, program v XLST totiž není pevně stanovená posloupnost příkazů, které by měl procesor provést, ale řídí se především strukturou zpracovávaného dokumentu. [4]



Obrázek 3 - Princip zpracování XML dokumentů pomocí XSLT stylu. [8]

2.8 XQUERY

S jazykem XQuery ve své Bakalářské práci moc nepracuji, nicméně jsem považoval za nutné se o tomto jazyce zmínit, alespoň v teoretické části mé práce.

Jazyk XQuery má opravdu velké ambice vycházející z potřeb současného světa SŘBD. Větší část syntaxe i sémantiky jazyka XPath zde nachází své využití, neméně důležité se ukázalo také koncepty a vlastnosti jazyka známé z SQL. Důležitá je i určitá vizuální podobnost s jazykem SQL, která umožňuje snazší uplatnění znalostí získaných při práci s tradičními relačními či objektově relačními SŘBD. Podobně jako jiné populární jazyky ani jazyk XQuery nevznikl sám od sebe, ale byl utvářen postupným vývojem a s využitím znalostí o jeho předchůdcích. XQuery je v podstatě přímým potomkem jazyka Quilt, který sloužil jako prototyp pro návrh uživatelské syntaxe, datového modelu a příkladů využití dotazování pro práci s XML daty. Ten shromažďoval poznatky z vývoje nejzajímavějších XML dotazovacích jazyků konce 90. let. [4]

XQuery je velmi komplexní XML dotazovací jazyk. Některé základní výrazy XQuery, jako jsou cesty, množina porovnávání, aritmetické výrazy, nebo filtrující predikáty známe již z jazyka XPath, další, pokročilejší konstrukce, jako jsou FLWOR výrazy, třídění, podmíněné výrazy nebo konstruktory, jsou dostupné až v XQuery. [4]

2.9 Aplikační rozhraní

Aby mohli aplikace s XML dokumentem jednoduše pracovat, využívají standardizovaná aplikační rozhraní XML procesorů. Existují čtyři „genetická“ API k procesorům, která zároveň poskytují jistý datový model XML, ve kterém jsou XML data reprezentována jako strom [4]:

- SAX – Simple API for XML (založené na událostech),
- DOM – Dokument Object Model (založené na stromech a objektech),
- JDOM – Java Document Object Model (založené na stromech a objektech),
- StAX – Streaming API for XML (založené na proudech dat a událostech)

Veškeré algoritmy konstruuující XML strom nebo vracející jeho uzly jsou založeny na sekvenčním čtení XML textu. Zmíněná rozhraní jsou založena na modelu Infoset. Protože sekvenční čtení XML dokumentu odpovídá průchodu příslušného stromu do hloubky ve variantě preorder, dostáváme uzly v pořadí odpovídajícím standardnímu uspořádání.

2.9.1 Infoset

Při uvažování o zpracování XML dokumentů je třeba studovat řadu otázek. Patří sem mimo jiné otázka, jakou informaci vlastně XML dokumenty obsahují, abychom věděli, jakou informaci mají XML procesory aplikacím vracet. Konsorcium W3C pro tyto účely definovalo abstraktní model obsahu XML dokumentů nazývaný významová množina XML dokumentu, zkráceně Infoset (XML Information Set), která specifikuje množinu informací obsaženou v XML dokumentu. Významová množina je specifikována pro každý dobře vytvořený XML dokument, který splňuje požadavky na správné použití jmenných prostorů. Nesprávně vytvořené XML dokumenty, nebo dokumenty využívající relativní URI, nemají významovou množinu definovanou. Existence významové množiny není vázána na validitu dokumentu. [4]

Významová množina XML dokumentu se skládá z informačních položek (information items) jedenácti typů. Každá položka má své vlastnosti (properties). Významová množina nezávisí na způsobu zpracování dokumentu, může být získána nejrůznějšími způsoby. Informační položky významové množiny mohou být [4]:

- popis dokumentu (Document Information Item),
- popis elementů (Element Information Items),
- popis atributů (Attribute Information Items),
- popis instrukcí (Processing Instruction Information Items),
- neexpandované entity (Unexpanded Entity Reference Information Items),
- znakové položky (Character Information Items),
- komentáře (Comment Information Items),
- popisy DTD (The Document Type Declaration Information Items),
- neanalyzované položky (Unparsed Entity Information Items),
- popisy notací (Notation Information Items),
- popisy jmenných prostorů (Namespace Information Items).

3 VYUŽITÍ XML V DATABÁZÍCH

Stejně tak jako patentované databázové formáty může být XML použit k označení jednotlivých polí uvnitř databázového záznamu (např. označení jména, adresy, telefonního čísla v záznamu v databázi adres). Označení každé části informace umožňuje zobrazení dat různým způsobem, jejich řazení, třídění, filtrování i jakékoli jiné zpracování. [3]

S rostoucím využitím XML, jakožto vhodného formátu pro reprezentaci dat, roste i potřeba efektivní správy XML dokumentů a dotazování XML dat. Na jedné straně sice máme velmi silnou sadu nejrůznějších standardů pro definici struktury XML dat, pro jejich zpracování, dotazování a transformaci, ovšem na straně druhé stále leží problém jejich efektivní a pokud možno úplné implementace. Obecně tuto problematiku nazýváme problém perzistentního ukládání XML dat. V množství různých návrhů a projektů, které se tímto tématem zabývají, můžeme vyzorovat několik základních strategií ukládání XML dat: uložení s využitím systému souborů, uložení v relačním databázovém systému a nativní (přímé) uložení XML dat. [4]

3.1 Uložení XML dat v relační databázi

Zde se zaměříme na správu XML dokumentů prostřednictvím relačních systémů řízení báze dat (RSŘBD), které jsou v literaturách obvykle označovány jako databáze s XML rozšířením (XML-enabled database). Důvody pro snahu o propojení těchto dvou technologií jsou poměrně zřejmé a plynou především z jejich vzájemné podobnosti, která může být velmi výhodná. Již samotný princip popisu struktury XML dokumentů má mnoho (objektově) relačních rysů (jako např. datové typy, identifikátory, dědičnost, substituovatelnost apod.) a podobně i ostatní XML technologie (jako např. dotazovací jazyky, schémata apod.) jsou databázovým technologiím velmi podobné. Faktem také je, že „klasické“ relační databáze jsou z praktického hlediska stále jedním z nejpoužívanějších způsobů jak ukládat a zpracovávat data a jejich dlouhá teoretická i implementační historie zajišťuje rozumnou úroveň spolehlivosti a efektivity. V neposlední řadě pak spojení těchto dvou technologií umožňuje rozšířit XML technologie o typicky databázové mechanismy, jako jsou indexy, transakční zpracování apod., které mohou být využity bez větších změn. [4]

3.2 SQL/XML

SQL/XML představuje rozšíření jazyka SQL umožňující pracovat s XML daty. Motivací pro vývoj této funkčnosti v SQL bylo nejen transportovat relační data do XML, ale rovněž zacházet s XML daty v rámci (O)RSŘBD podobným způsobem jako s relačními daty a kombinovat relační a XML data v dotazech. Technicky taková integrace dat znamená vyřešit obousměrný přechod od jedné dat k druhým a umožnit programátorovi v QLS zpracovávat XML data tak, jak nabízejí současné XML jazyky. Jinými slovy řečeno, z pohledu programátora jde o nativní zpracování XML v rámci objektově relační databáze. SQL/XML je vyvíjen skupinou H2.3 Task Group, v rámci číslovaných standardů SQL v části 14 (XML-Related Specifications). Ve skupině participují firmy Oracle, IBM, Microsoft, DataDirect Technologies a Sybase. Výsledek práce této skupiny ve verzi SQL 2003 obsahuje [4]:

- funkce pro vytváření XML dat,
- datový typ XML a
- mapovací pravidla.

Funkce pro vytváření XML dat se používají v příkazu SELECT přímo ve složce SELECT, datový typ XML umožňuje ukládat XML data do tabulek a přiřadit typ výsledku dotazu, který obsahuje XML data. Mapovací funkce určují jak reprezentovat SQL data, jako XML data a zobrazit relační metadata do konstruktů jazyka SQL/XML a naopak. Metadata pro XML data jsou vyjádřena v jazyce XML Schema. V dalším vývojovém stupni jazyka známého jako SQL/XML:2006 se objevují možnosti dotazování XML dat uložených v tabulkách jako data typu XML. Dotazování je realizováno pomocí funkce XMLQUERY a XMLTABLE a je založeno na jazyku XQuery. [4]

Je zřejmé, že klíčovým místem návrhu je sémantika nového XML typu. Hodnota tohoto typu se nazývá XML hodnota, která intuitivně řečeno, je buď XML elementem, nebo lesem XML elementů. Sémantika XML hodnoty vychází v SQL/XML:2003 z abstraktního modelu Infoset. V dalším vývoji SQL/XML je však již založena na datovém modelu XQuery. [4]

3.3 XML v Oracle Database

Oracle Database dokáže nejen vygenerovat výstup z databázové tabulky do formátu XML, umožňuje také definovat datový typ XML. To znamená, že můžeme vytvořit přímo

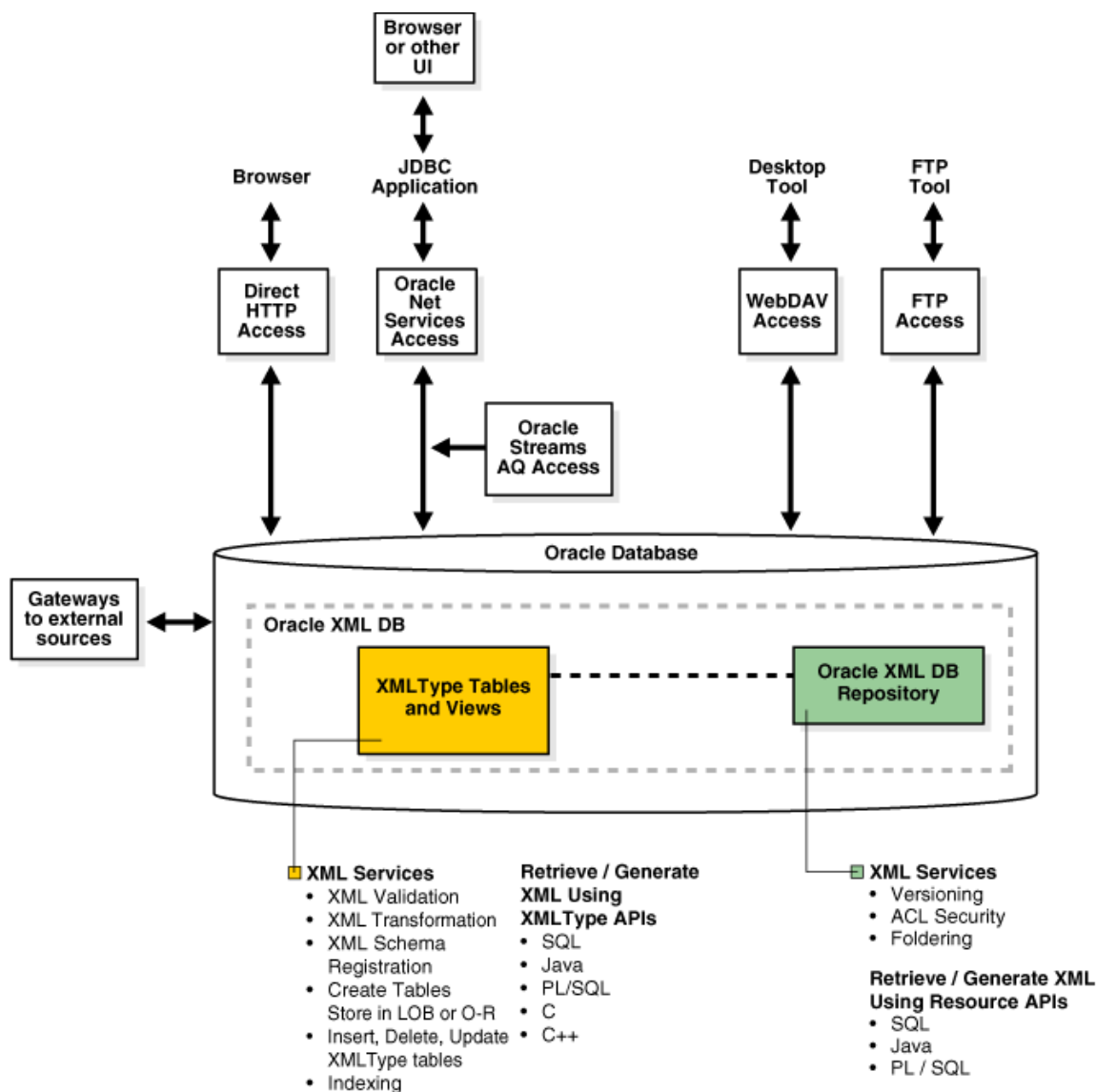
sloupce, které budou obsahovat data ve formátu XML. Technologie XML je tedy otevřená a nezávislá na platformě. XML soubor obsahuje kromě samotných dat i jejich strukturu. Zjednodušeně by se dalo říci, že tento formát pro výměnu dat je stejně dobře čitelný pro počítače i pro lidi. Tato data můžeme lehce načíst a identifikovat pomocí poměrně jednoduché rutiny v libovolném programovacím jazyku. [2]

3.3.1 Implementace XML na platformě Oracle Database 10g

Podpora technologie XML je na platformě 10g implementovaná ve více rovinách důsledně vyhovující standardům konsorcia W3C. Je přitom značný rozdíl mezi implementací ve verzích Release 1 a Release 2. Ve druhé verzi databáze 10g byla podpora XML výrazně vylepšena – při současném zjednodušení přístupu, používání i administrace. Blok podpory XML dokumentů označujeme jako XML DB. Jde o množinu výkonných ukládacích a vyhledávacích technologií zaměřených na podporu práce s XML dokumenty. Tyto technologie jsou implementovány přímo do databázového serveru, jsou tedy spojeny výhody technologií relačních databází a XML. Takové řešení umožňuje vzájemnou obousměrnou dualitu. Můžeme tedy provádět XML operace nad daty v relačních databázích a současně SQL operace nad daty uloženými v XML dokumentech. Je možné ukládat a spravovat strukturovaná i nestrukturovaná data. XML dokumenty jsou uloženy v XML repositářích (XML repository), které umožňují přístup k těmto dokumentům prostřednictvím protokolů WebDAV, HTTP a FTP. XML repositář můžeme chápat jako sklad pro správu XML dokumentů a pro přístup k těmto dokumentům. Jeho součástí je správa přístupových oprávnění, správa adresářů, protokoly WebDAV a ftp pro přístup k obsahu složek a aplikační rozhraní pro práci s XML. Skládá se z následujících částí. [2]

- Seznam a správa přístupových oprávnění (ACL) – i vlastní oprávnění
- Správa složek
- WebDAV a FTP protokol pro přístup ke složkám
- SQL vyhledávání v XML repositáři

- API pro práci s XML Repositáři
- Přístup pomocí Java servletu – manipulace s objekty



Obrázek 4 - Architektura Oracle XML DB: XMLType a Repositář [5]

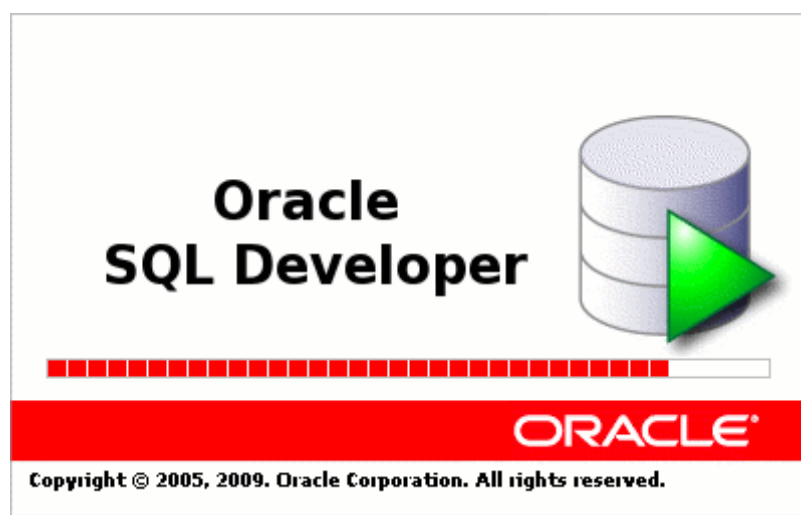
3.3.2 Administrace XML na platformě Oracle Database 10g

XML dokumenty mohou být uloženy přímo v databázových tabulkách nebo v databázi uložených XML dokumentů. Nejprve se pro administraci přihlásíme k databázovému serveru prostřednictvím webové administrátorské aplikace Oracle Enterprise Manager. Po přihlášení přejdeme do záložky Administration a všimneme si skupiny odkazů XML Database. Tato skupina je určena pro nastavení a správu podpory XML v databázích Oracle Database 10g Release 2. [2]

II. PRAKTICKÁ ČÁST

4 VÝVOJOVÉ PROSTŘEDÍ PRO TVORBU PŘÍKLADŮ

Pro odzkoušení příkladů pro práci s XML v databázi Oracle jsem využil software Oracle SQL Developer, který firma Oracle poskytuje jako vývojové prostředí ke svým databázím. Tento software je možné použít i pro práci s jinými databázemi, jako jsou např. Microsoft SQL Server, MySQL a spousta dalších, přednostně byl ovšem vyvinut pro Oracle databáze. Tento software je volně ke stažení na oficiálních stránkách této firmy, kde je možné si stáhnout i anglické manuály pro jeho obsluhu. Oracle SQL Developer je ve formě uživatelsky příjemného grafického prostředí, ve kterém lze přehledně spravovat databázi.



Obrázek 5 – Logo softwaru Oracle SQL Developer

Instalace:

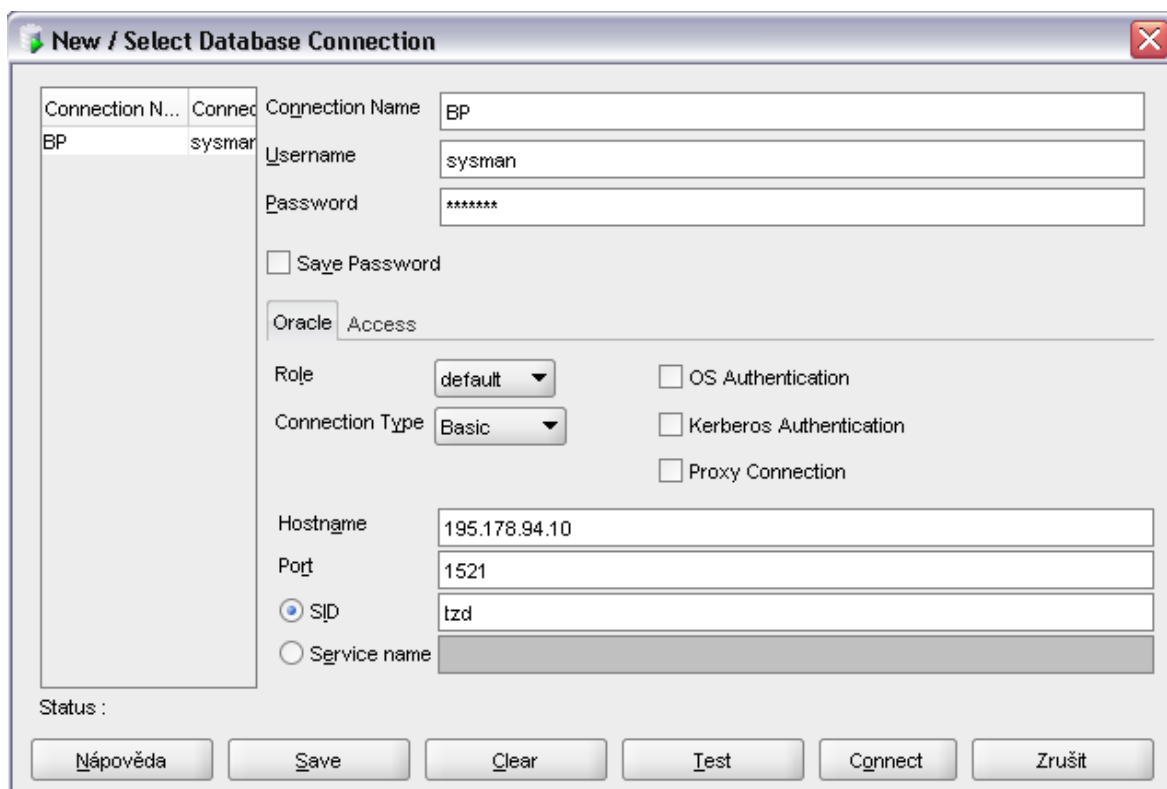
Po stažení Oracle SQL Developeru z oficiálních stránek uživatele čeká příjemné zjištění, že není třeba tento program instalovat, ale stačí jen rozbalit zip archiv, který jsme si stáhli. Celý program jen napsán v jazyce Java a je dostupný pouze v anglickém jazyce. Při vytváření a testování příkladů v příručce jsem využíval verzi Oracle SQL Developer 1.5.5. Proto, abychom mohli využívat aplikace, které jsou napsány v jazyce Java, je nutné mít nainstalováno JRE (Java Runtime Environment) což je prostředí, které umožňuje spouštět aplikace napsané v tomto jazyce.

Připojení:

Pro vytváření a následné testování příkladů obsažených v příručce bylo nutné se připojit na školní server, na kterém je nainstalována Oracle databáze verze 10g Release 2. Komunikaci s databází zajišťuje přímo Oracle SQL Developer a není k ní potřeba žádné další aplikace.

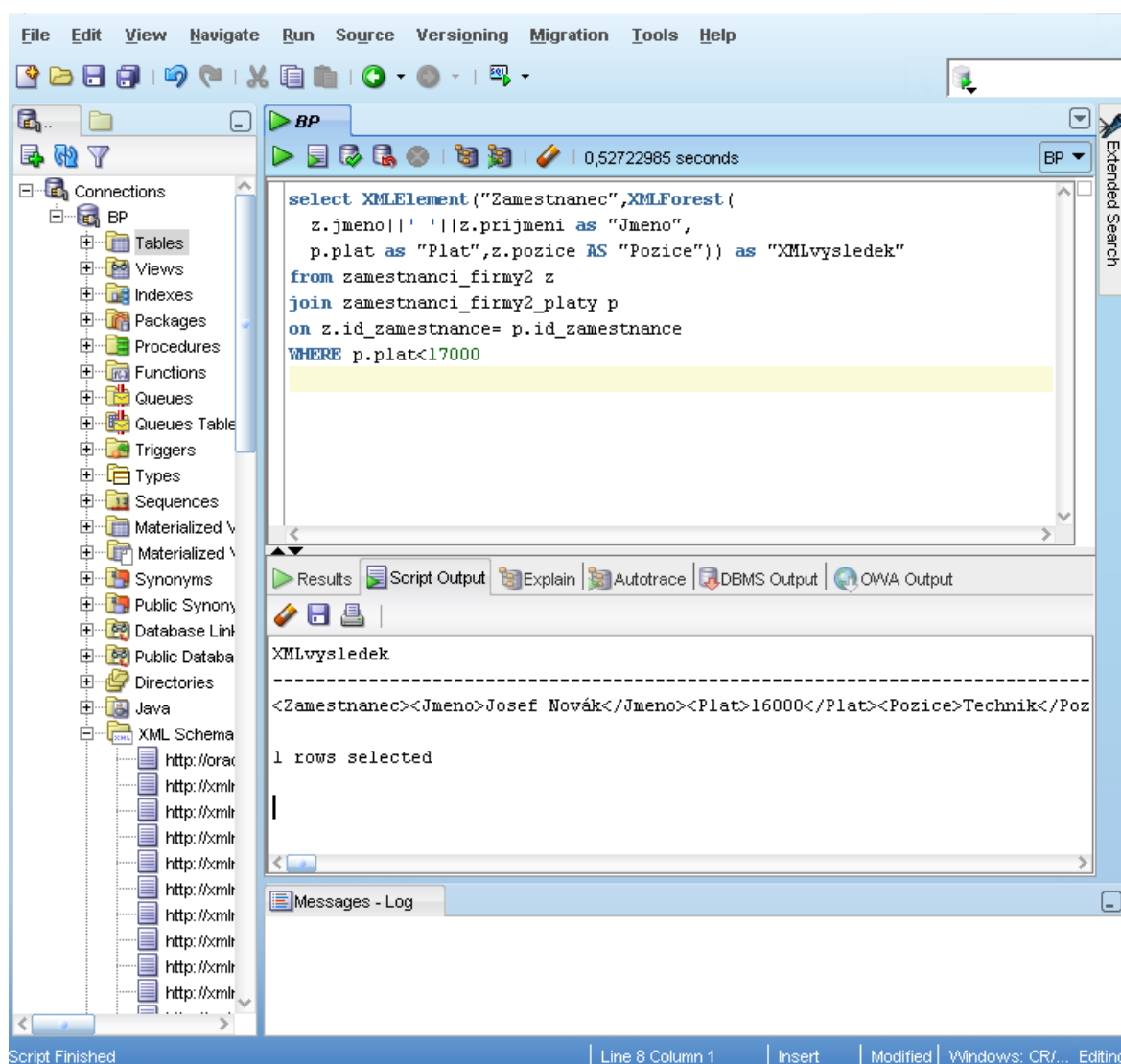
Po spuštění programu Oracle SQL Developer vyhledáme v levém sloupci programu Connections a pro vytvoření nového připojení po kliknutí pravým tlačítkem, zvolíme New Connection. Otevře se okno, do kterého se zadávají údaje potřebné pro připojení k databázi:

- Connection Name – jméno našeho připojení
- Username – uživatelské jméno
- Password - heslo
- Hostname – adresa serveru pro připojení k databázi
- Port – číslo portu
- SID – jméno instance pro dané připojení



Obrázek 6 – Okno pro vytvoření nového připojení v Oracle SQL Developer

Pro kontrolu správně nastavených připojovacích údajů můžeme využít tlačítko Test. Po ověření potvrdíme tlačítkem Connect a připojíme se k databázi. Po úspěšném připojení k databázi se nám v levém sloupci pod vytvořeným připojením rozbalí stromová struktura Oracle databáze, která obsahuje tabulky, indexy, adresáře, schémata atd. Po připojení se nám také zobrazí záložka s názvem našeho vytvořeného připojení, do které je možné psát SQL dotazy, které následně spustíme tlačítkem F9 nebo kliknutím na tlačítko Execute Statement (pod zelenou šipkou). Výsledek našeho dotazu se zobrazí pod tímto SQL dotazem v záložce Results případně Script Output.



Obrázek 7 – Screen z vývojového prostředí Oracle SQL Developer

5 UŽIVATELSKÁ PŘÍRUČKA A PŘÍKLADY

V této kapitole je zahrnuto a popsáno jak byly vytvářeny jednotlivé typy příkladů v uživatelské příručce a základní informace o uživatelské příručce samotné. Při tvorbě uživatelské příručky bylo dbáno na to, aby všechno co se týkalo následné aplikace na příkladech, bylo předem vysvětleno a nedocházelo tak k chybným syntaktickým zápisům jednotlivých příkladů. Při tvorbě uživatelské příručky byl kladen velký důraz na jednoduchost a srozumitelnost tak, aby bylo jasné, co který příklad znamená a proč je zapsán takovým způsobem.

5.1 Uživatelská příručka

Uživatelská příručka, která tvoří přílohu k této bakalářské práci, je v podstatě praktickým výstupem této práce. Příručka je rozdělena na kapitoly, ve kterých se postupně student seznamuje s XML a od úplného začátku se základními principy tvorby syntakticky správně strukturovaných dokumentů až po konkrétní využití XML v databázi Oracle na vysvětlených a předvedených příkladech.

Hlavní kapitoly příručky jsou:

- Základní principy XML
 - XML Dokument
 - Základní pravidla XML
- Definice schématu XML
 - Definice dokumentu DTD
 - XML Schema
- XPath
 - Úvod do jazyka XPath
- XML v Oracle databázi
 - XML v Oracle
 - Příklady v Oracle databázi
 - Příklady dotazů nad XML daty
 - Příklady vytváření XML z dat tabulky

Tyto kapitoly se v příručce dělí ještě na velké množství podkapitol, které se zabývají vysvětlením konkrétních pravidel, metod, funkcí, syntaxe atd.

Stručný popis jednotlivých hlavních kapitol:

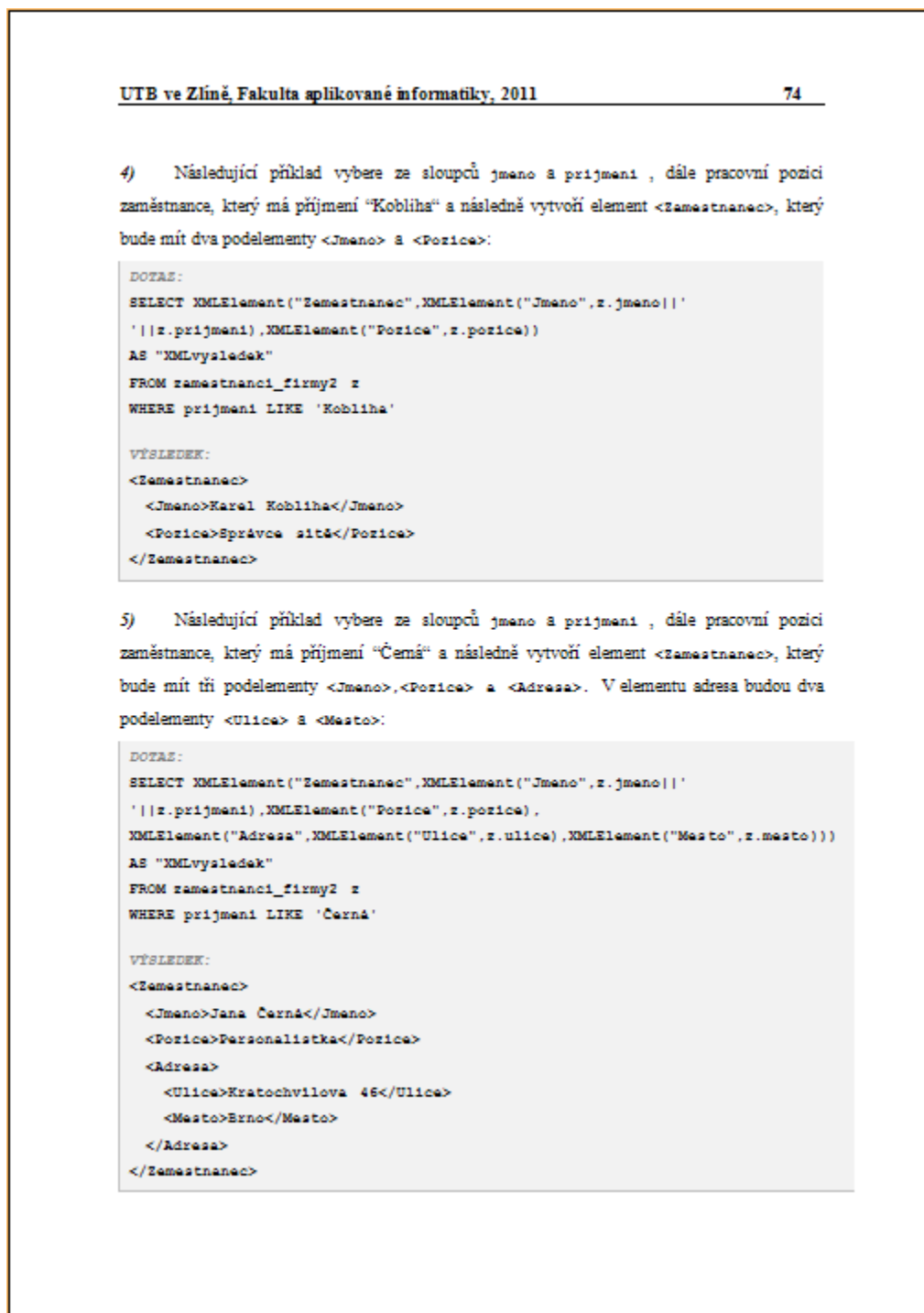
V kapitole *Základní principy XML* je od nejmenšího základu vysvětleno co je to XML a jak správně XML utvářet. V podkapitole *XML Dokument* je vysvětleno, co všechno by měl obsahovat správně strukturovaný XML dokument. V každé jednotlivé části je uveden příklad kódu, jak by tento kód měl správně vypadat a jakých chyb se vyvarovat při tvorbě XML dokumentů. Podkapitola *Základní pravidla XML* již obsahuje konkrétní základní pravidla co, a jak se v XML dokumentech může nebo naopak nemůže zapsat. U všeho jsou uvedeny příklady, jak by měl vypadat správný zápis a jak většinou vypadá nesprávný.

Kapitola *Definice schématu XML* se již věnuje tomu, jak vytvořit správné schéma, které definuje strukturu XML dokumentu, který chceme vytvářet. Nesmírnou výhodou je specifikace značek, které mohou být v dokumentu použity. Této kapitole je věnováno poměrně dost stránek, protože vytvoření XML schéma za pomoci jazyka XML Schema je možné různými přístupy. Vytvoření XML schématu je uvedeno i v příkladové části příručky přímo v databázi Oracle.

Kapitola *XPath* obsahuje od základu vysvětlení jak za pomoci tohoto dotazovacího jazyka tvořit správně strukturované dotazy nad XML daty a je zde na několika stranách názorně vysvětleno, jak celý jazyk XPath funguje. V této kapitole je uveden a vysvětlen datový model tohoto jazyka, tvorba jednotlivých výrazů, vysvětlení funkce `os` v jazyce XPath, je zde uveden správný zápis jazyka, jak vybrat elementy a atributy atd.

Poslední a ve své podstatě i nejdůležitější hlavní kapitolou této příručky je *XML v Oracle databázi*. V této kapitole jsou uvedeny jednotlivé příklady ke všem probraným tématům z předchozích kapitol. V této kapitole se využívají znalosti ze všech předchozích kapitol dohromady a bylo velmi důležité, aby všechny pravidla a funkce pro tvorbu XML byly kvalitně vysvětleny. Více již o příkladové části uživatelské příručky je zmíněno v kapitole *5.2 Příklady v Oracle*.

Následující obrázek je ukázkou jedné stránky u vytvořené uživatelské příručky:



Obrázek 8 – Ukázka stránky z uživatelské příručky

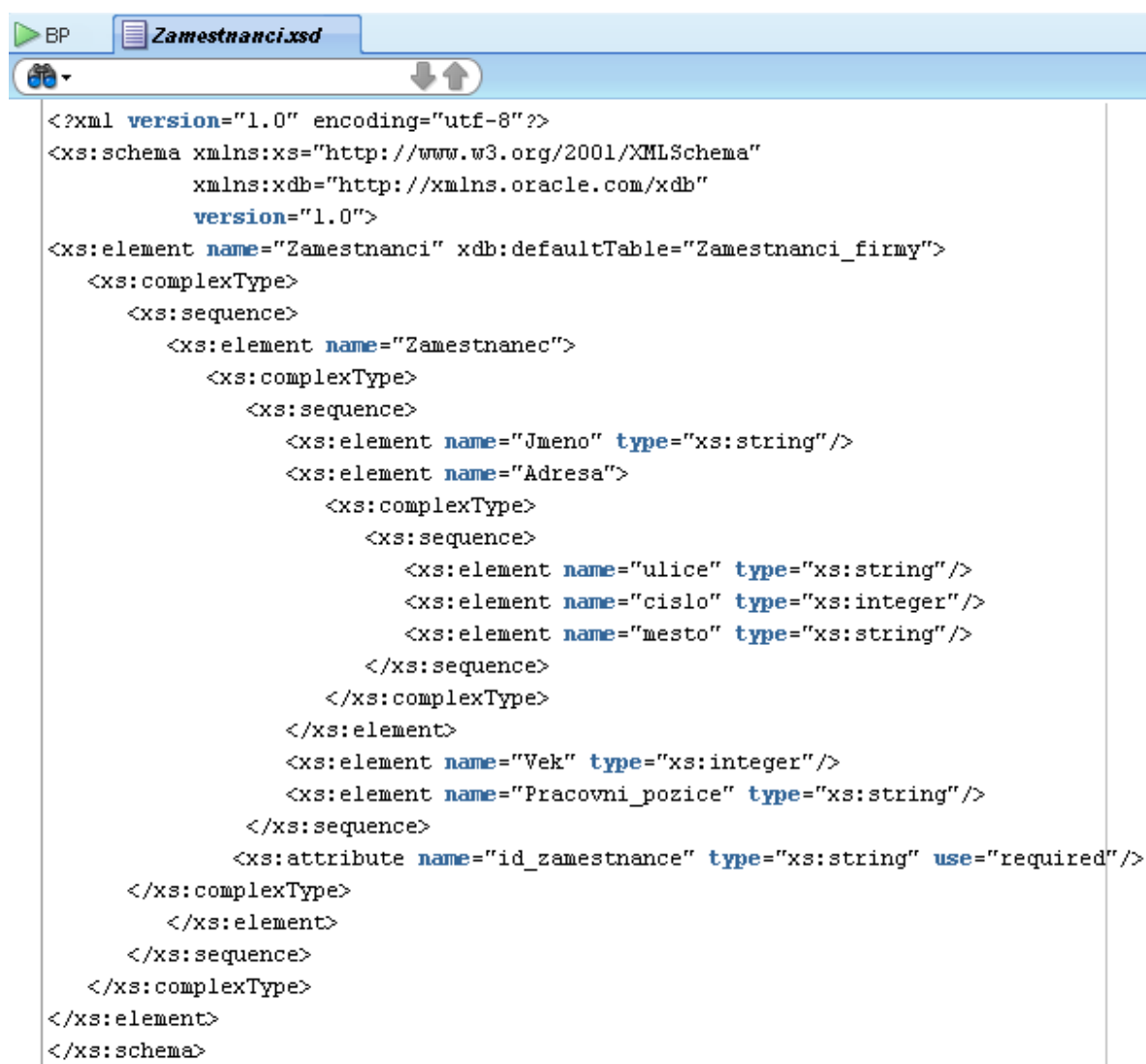
5.2 Příklady v Oracle

Příklady jsou jednou z hlavních částí uživatelské příručky a v kapitole příručky *XML v Oracle databázi* je názorně vysvětleno jak přenést znalosti z předchozích kapitol příručky přímo do Oracle databáze. Všechny uvedené příklady jsou okomentovány a jednotlivé

kroky, které byly při tvorbě příkladů učiněny, jsou zde popsány. V příručce jsou uvedeny příklady na tvorbu XML schémat, registrace a mazání schématu, dále příklady na naplnění tabulky XML daty tak, že jsou data přímo obsažena v dotazu nebo že jsou data nahraná do tabulky z XML souboru umístěného v předem specifikovaném adresáři. Tato část příručky dále obsahuje příklady jak za pomoci jazyka XPath manipulovat s XML daty, která máme již vloženy v tabulce typu XMLType. Jako jedny z posledních příkladů jsou v příručce uvedeny příklady na tvorku XML z dat, která jsou uložena v databázové tabulce.

Příkladů je v příručce poměrně velké množství a byla snaha je srozumitelně popsat, proto v následujících kapitolách budou více přiblíženy jen hlavní příkladové části z příručky:

5.2.1 Příklady na tvorbu XML schéma



Obrázek 9 – Ukázka vytvořeného schématu v Oracle

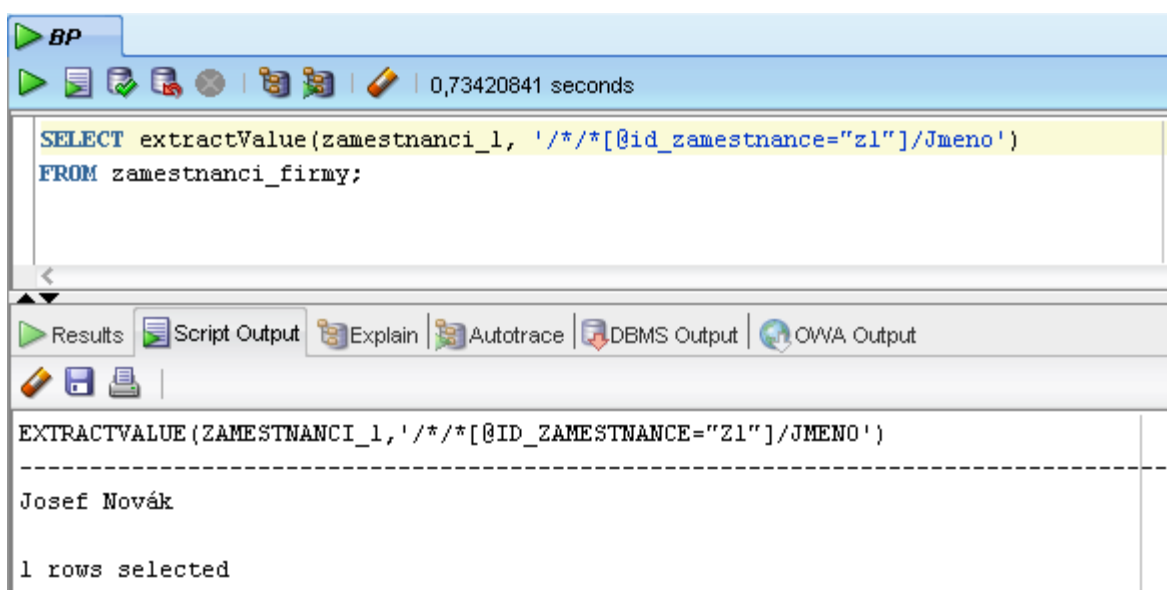
Na předchozím *Obrázku 9* můžeme vidět ukázkou vytvořeného XML schématu v databázi Oracle, které popisuje strukturu XML dokumentu zaměstnanců firmy. V případě tohoto schématu byla nastavena výchozí tabulka, to znamená, že se při registraci schématu vytvoří tabulka `zamestnanci_firmy`, která bude mít jeden sloupec typu `XMLType`. V případě, že výchozí tabulku nastavenou nemáme, tak si pak tabulku, ke které bude připojeno XML schéma, musíme vytvořit sami.

XML Schéma v databázi Oracle po uložení nevypadá tak jako na *Obrázku 9*, ale databáze si ho přizpůsobí a přidá si do něj např. informace o tom, na jaký databázový datový typ se má převést daný XML datový typ.

Databáze Oracle nedovolí uložit schéma, které je nějak nesprávně strukturované. Stačí, aby ve schématu chyběla jedna ukončovací značka, a Oracle jej již nedovolí uložit s výstražnou hláškou, že XML schéma není syntakticky správně napsáno.

5.2.2 Příklady výběru dat pomocí XPath

Jazyk XPath je jedním z nejdůležitějších a nejefektivnějších nástrojů v případě, že se chceme dotazovat nad XML daty. Umožňuje nám ve velmi krátkém zápise si vytáhnout z XML dokumentu tu informaci, kterou právě potřebujeme. V uživatelské příručce bylo věnováno dotazování se nad daty za pomoci XPath poměrně dost příkladů. Další příklady by byly analogické k již vytvořeným příkladům a za pomoci předchozí kapitoly XPath je možné vytvořit již prakticky jakýkoliv dotaz.



Obrázek 10 – Ukázka Xpath dotazu nad XML daty v Oracle

Na předchozím *Obrázku 10* je možné vidět malou ukázkou dotazu, který nám vypíše z XML dokumentu o zaměstnancích firmy jméno jednoho zaměstnance, jehož `id_zamestnance="z1"` a element, ve kterém je tato id uložena má právě dva předky. Výsledné jméno nalezeného zaměstnance s hledanou `id_zamestnance` je možné vidět níže pod dotazem.

U XPath dotazů je dobré si dávat pozor na to, jestli daný dotaz nelze zapsat ještě jednodušší formou, protože někdy není nutné složitě vypisovat cestu přes několik elementů a stačí nám zapsat pár značek ve správné posloupnosti a dokonce u mnohonásobně kratšího dotazu můžeme mít stejný výsledek jako u složitě vykonstruovaného dotazu.

V příručce jsou příklady řešeny tak, že je pro ukázkou předem známá struktura dokumentu, pak je názorně uvedený dotaz a hned za dotazem výsledek, který tento dotaz po provedení v databázi vrátí. Jeden z příkladů je pro ukázkou uveden zde:

DOTAZ:

```
SELECT extract(zamestnanci_1,'//Zamestnanec[2]/Adresa/*')  
FROM zamestnanci_firmy
```

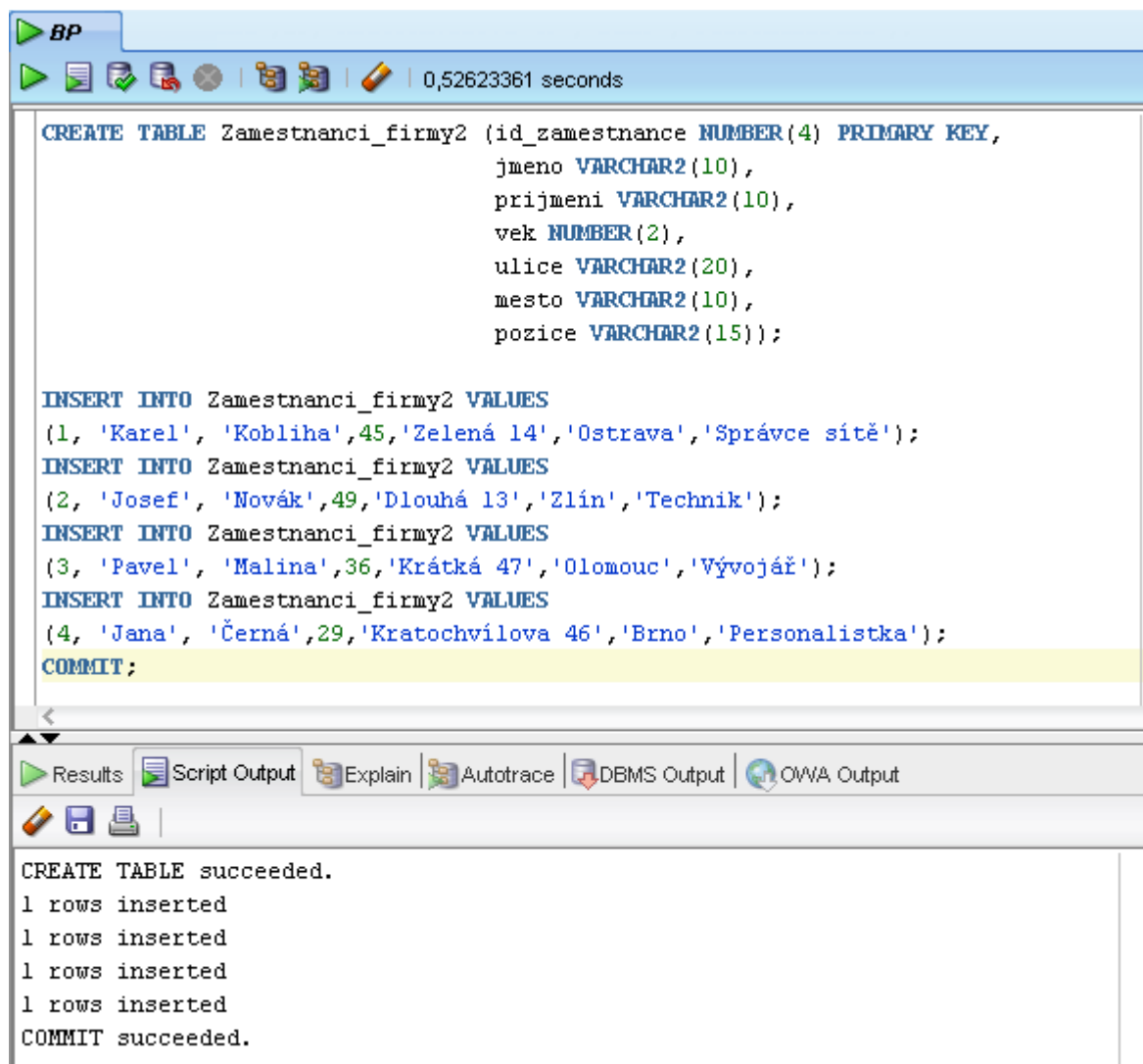
VÝSLEDEK:

```
<ulice>Krátká</ulice>  
<cislo>47</cislo>  
<mesto>Olomouc</mesto>
```

V tomto příkladě dotaz vrátí adresu v pořadí druhého zaměstnance firmy. Vyberou se všichni přímí potomci uzlu `Adresa`, což je `Ulice`, `Číslo`, `Město` a následně se vypíší. Příkaz vybírá všechny elementy `Zamestnanec` v celém dokumentu, to se provede za pomoci `"//"` dvou lomítek před názvem elementu. Vybírá se z tabulky `zamestnanci_firmy` a sloupce tabulky XMLType s názvem `zamestnanci_1`.

5.2.3 Příklady na tvorbu XML z dat tabulky

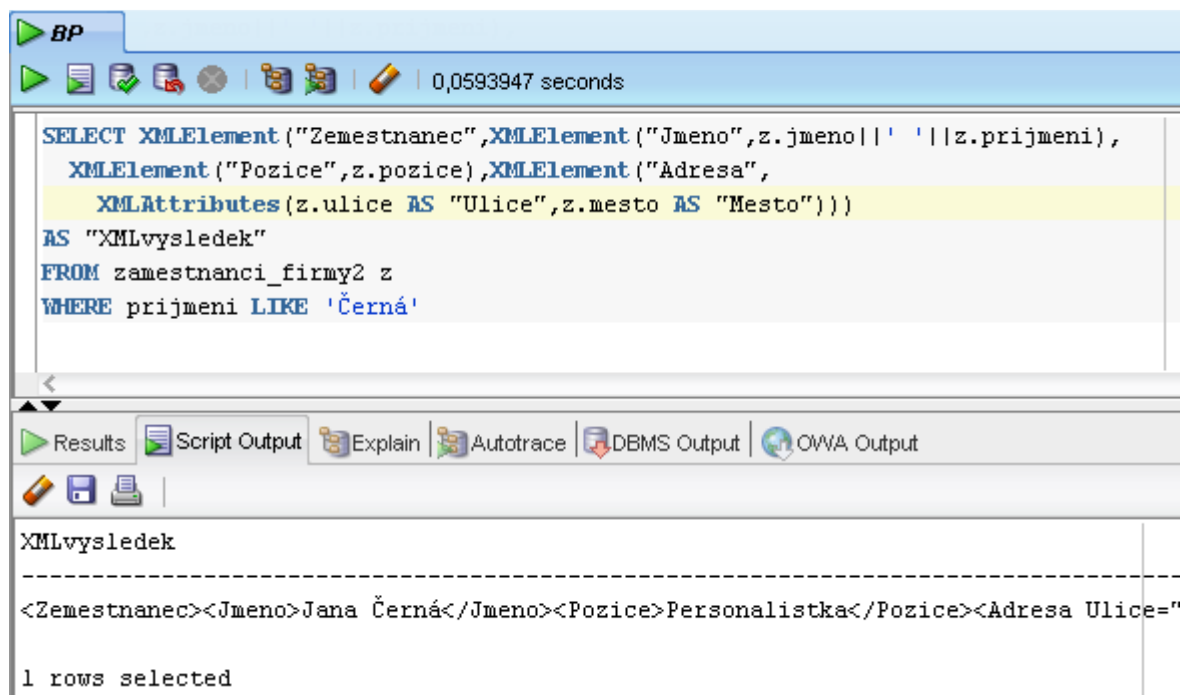
V této příkladové části bylo nejprve nutné si vytvořit nějakou tabulku, nad kterou bychom dotazy prováděly a následně ji naplnit daty. Kdybychom chtěly vytvářet XML z již nějaké existující tabulky, tak by to problém nebyl, ale aby byla ukázkou opravdu názorná, jsou příklady uvedeny i s vytvořením této tabulky. V příručce je příklad na vytvoření tabulky pro evidenci zaměstnanců firmy, která je naplněna čtyřmi zaměstnanci. Názornou ukázkou vytváření této tabulky je možné vidět na *Obrázku 11*. V případě, že máme SQL správně zapsáno tak nám databáze potvrdí vytvoření tabulky hláškou `CREATE TABLE succeeded`.



Obrázek 11 – Ukázka vytvoření tabulky z příručky v Oracle

Po vytvoření tabulky bylo možné začít tvořit dotazy. Jednotlivé dotazy jsou opět tvořeny tak, že je nejprve v příručce napsaný dotaz a hned za ním výsledek co dotaz vrátí, to proto, aby měl ten, kdo bude podle příručky zkoušet sám tvořit dotazy obrázek o tom, jaký výsledek dostane. V této příkladové části bylo dbáno hlavně na vysvětlení funkcí XMLElement, XMLAttributes, XMLForest, XMLConcat, které jsou pro tvoření dotazů nezbytné.

Na následujícím Obrázku 12 je možné vidět vytvořený dotaz nad tabulkou s daty o zaměstnancích, který vrací údaje o zaměstnanci s příjmením “Černá”. Výsledné XML obsahuje jméno zaměstnance, pracovní pozici a adresu, ve které jsou formou atributu uloženy údaje o ulici a městě. Hned za obrázkem je uveden tento příklad ve formě, v jaké je uveden v uživatelské příručce.



Obrázek 12 – Ukázka tvorby XML z dat uložených v tabulce Oracle

DOTAZ:

```
SELECT XMLElement("Zemestnanec",XMLElement("Jmeno",z.jmeno||' '
' ||z.prijmeni),XMLElement("Pozice",z.pozice),
XMLElement("Adresa",XMLAttributes(z.ulice AS "Ulice",z.mesto AS
"Mesto")))
AS "XMLvysledek"
FROM zamestnanci_firmy2 z
WHERE prijmeni LIKE 'Černá'
```

VÝSLEDEK:

```
<Zemestnanec>
  <Jmeno>Jana Černá</Jmeno>
  <Pozice>Personalistka</Pozice>
  <Adresa Ulice="Kratochvílova 46" Mesto="Brno"></Adresa>
</Zemestnanec>
```

5.2.4 Ostatní příklady v databázi Oracle

V předchozích kapitolách byly uvedeny a více přiblíženy nejhlavnější příkladové kapitoly z uživatelské příručky. Tyto kapitoly ovšem nejsou jediné kapitoly s příklady, které se v příručce vyskytují a ostatní příklady jsou dostatečně popsány v příručce v kapitole *Příklady v Oracle databázi*. Jako další příklady v příručce nalezneme příklady na registraci XML schématu, kde je popsáno jak vytvořené schéma zaregistrovat do databáze, také jak

zaregistrované schéma případně smazat. V příručce se také nachází příklady na tvorbu tabulek pro XML data a to jak s jedním sloupcem datového typu XMLType, tak s více sloupci. Je zde podrobně popsáno jak naplnit tabulku XML daty obsaženými přímo v dotazu, nebo jak naplnit tabulku XML daty, která jsou obsažena v souboru na serveru, dále také, jak si XML data z tabulky vytáhnout.

ZÁVĚR

Tato bakalářská práce se podle zadání zabývá využitím jazyka XML (Rozšiřitelný značkovací jazyk) v databázovém systému firmy Oracle. Za pomoci tohoto jazyka je možné si označit jednotlivá pole uvnitř databázového systému a označením každé části informace se nám tak nabízí možnost data uložená v databázi zobrazit různými způsoby a je možné tato data efektivně řadit, filtrovat, třídit a provádět s nimi jakékoliv jiné operace.

V teoretické části této práce jsou nastíněny teoretické aspekty potřebné pro utvoření si určité obecné představy toho co je to XML a jak se v databázi využívá. V první kapitole jsou uvedeny úvodní informace o databázovém systému Oracle, obecné informace o databázových systémech a nakonec popis konkrétního databázového systému Oracle 10g Release 2, který je nainstalovaný na školním serveru, a ve kterém byly testovány praktické příklady. Druhá kapitola teoretické části této práce je již věnována samotnému jazyku XML a tomu, jak se využívá v praxi. Spolu se základním uvedením do XML jsou zde také popsány základní termíny, se kterými se v této práci setkáváme, jako je XML dokument a jeho syntaxe, definice XML schéma, dotazovací jazyk nad XML daty XPath atd. Ve třetí a také poslední teoretické kapitole je obecně popsáno jakou funkci má XML v databázích a jak je XML zaimplementováno do oblasti databáze Oracle, dále jak je všeobecně XML využíváno v databázích.

Praktickým výstupem této bakalářské práce je uživatelská příručka, která by měla sloužit jako studijní podpora pro studenty předmětu Teorie zpracování dat. Příručka je pojata hlavně prakticky a v její první kapitole je nejprve popsáno, jak od základu vytvářet syntakticky správně strukturované dokumenty XML podle základních pravidel. U všech vysvětlených pojmů je přiložena část kódu, jak by měla být správně tato část zapsána. V některých případech je v příručce uveden pro názornou ukázkou i nesprávný zápis daného kódu, který je však viditelně označen jako nesprávný. Ve druhé kapitole příručky je na příkladech od základu popsáno jak vytvářet XML schémata v jazyce XML Schema, která určují, jakou strukturu by měl mít XML dokument a co všechno by měl tento dokument obsahovat. Třetí kapitola příručky se zabývá dotazovacím jazykem XPath, který se využívá v dotazování se nad XML daty a umožňuje pomocí jednoduše zapsaného dotazu získat požadovanou informaci z XML. V příručce je přesně popsáno, jak za pomoci tohoto jazyka můžeme vybírat konkrétní elementy a pracovat s jejich hodnotami a atributy. Vše v části XPath je opět názorně ukázáno na jednotlivých částech kódu. V poslední čtvrté závěrečné části příručky jsou již uvedeny jen příklady jak využívat XML v databázi Oracle. Vše je

popsáno na konkrétních příkladech, v případě dotazu je vždy uveden výsledek, jaký daný dotaz vrátí atd.

V závěru je v praktické části bakalářské práce popsáno vývojové prostředí, ve kterém byly vytvořeny příklady v databázi Oracle a jak byla uživatelská příručka postupně vytvářena. Ke každé části typových příkladů v příručce je zde popsáno, jak byly tyto příklady vytvářeny a k čemu slouží.

CONCLUSION

This bachelor thesis is, according to the instructions, concerned with the use of XML (Extensible Markup Language) in the Oracle database system. Using this language, it is possible to mark individual fields within a database and give a name to each part of the information, which brings the opportunity to display the data stored in the database in various ways. Moreover, it is possible to sort this data efficiently, and filter it, as well as many other operations.

The theoretical part of this thesis is focused on theoretical aspects necessary for formation of a general idea of what XML is and how to use it in the database. The first chapter provides introductory information about the Oracle database, information about database systems in general, and finally a description of a particular database system; this being the Oracle 10g Release 2, which is installed on the school server, and in which the practical examples were tested. The second chapter of the theoretical part is devoted to XML itself and to the way it is used in practice. Along with a simple introduction to XML, it also describes the basic terms used in this thesis, such as XML document and its syntax, the XML schema definition, querying the XML data using XPath, etc. Within the third and the last chapter of the theoretical part, the function of XML in databases, the way XML is implemented within the Oracle database, and the way XML is used in databases generally are all described.

The practical outcome of this thesis is a user guide to a subject called "Data Processing Theory" which should serve as a study support for students. The user guide is created to be primarily practical. In its first chapter, how to create a syntactically well-formed XML documents according to the basic rules is described as a basis for other work. All the terms are explained. Codes are attached to them in their correct forms. In some cases, the guide explains and demonstrates incorrectly written codes; those are clearly marked as incorrect. The second chapter of the guide describes how to create XML schemas in "XML Schema" language, while using appropriate examples. XML schema specifies the structure an XML document is to have and what it is to contain. The third chapter of this guide deals with the XPath query language, which is used for querying XML data. Using it, it is possible to obtain the desired information from XML with an easily written query. Selecting of single elements and work with their values and attributes are also defined. In this part of the guide, dealing with the XPath, everything is exemplified with various parts of the code. In the fourth and the last section of the guide, examples of usage of XML in the Oracle database

are formed. Everything is explained using specific examples. For each query there is the result which the query returns.

In the end of the practical part of the thesis, the development environment in which examples were created in Oracle database is described, as well as the way the user guide was gradually developed. For each type of the examples in the guide, both the way the examples were created and their functions are described.

SEZNAM POUŽITÉ LITERATURY

- [1] Procházka, David. *Oracle průvodce správou, využitím a programováním nad databázovým systémem*. 1. vyd. Praha: Grada Publishing, a.s., 2009. 168 s. ISBN 978-80-247-2762-2.
- [2] Lacko, Luboslav. *Oracle – Správa, programování a použití databázového systému*. 2. vyd. Brno: Computer Press, a.s., 2007. 576 s. ISBN 978-80-251-1490-2
- [3] Young, Michael J. *XML – krok za krokem*. 2. vyd. Brno: Computer Press, a.s., 2006. 471 s. ISBN 80-251-1070-2
- [4] Mlýnková, Irena, Pokorný, Jaroslav. *XML Technologie*. 1. vyd. Praha: Garda Publishing, a.s., 2008. 272 s. ISBN 978-80-247-2725-7
- [5] Oracle® XML DB. *Developer's Guide 10g Release 2 (10.2)*. [Online] 26.03.2011. [Citace: 26. 03 2011.] http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14259.pdf
- [6] Skřivánek, František. *Oracle Database 10g Release 2*. [Online] 19.7.2005. [Citace: 26. 03 2011.] <http://www.dbsvet.cz/view.php?cisloclanku=2005071902>
- [7] Loney, Kevin, Brila, Bob. *Mistrovství v Oracle database 10g*. 1. vyd. Brno: Computer Press, a.s., 2007. 700 s. ISBN 80-251-1277-2
- [8] Kosek, Jiří. *Seriál o XML pro Softwarové noviny*. [Online] 2001. [Citace: 29.3.2011] <http://www.kosek.cz/clanky/swn-xml/index.html>
- [9] Wikipedia. *Extensible Markup Language*. [Online] 23.2.2011, [Citace: 9.4.2011] http://cs.wikipedia.org/wiki/Extensible_Markup_Language

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

API	Application Programming Interface (Rozhraní pro programování aplikací)
DBMS	Database management system (Systém řízení báze dat)
DOM	Dokument Object Model (Objektový model dokumentu),
DTD	Document Type Definition (Definice typu dokumentu)
FTP	File Transfer Protocol (Protokol pro přenos souborů)
HTML	HyperText Markup Language (Značkovací jazyk pro hypertext)
HTTP	Hypertext Transfer Protocol (Internetový protokol pro výměnu dokumentů)
JDOM	Java Document Object Model (Java objektový model dokumentu),
ORSŘBD	Objektově Relační Systémy Řízení Báze Dat
PC	Personal computer (Personální počítač)
RDBS	Relational Database Management System (Relační systém řízení báze dat)
RSX	Real-time (operační systém reálného času)
SAX	Simple API for XML (Umožňuje sériový přístup k XML)
SGML	Standard Generalized Markup Language (značkovací metajazyk, který umožňuje definovat značkovací jazyky)
SQL	Structured Query Language (Strukturovaný dotazovací jazyk)
SŘBD	(Systém řízení báze dat)
URI	Uniform Resource Identifier (Jednotný identifikátor zdroje)
W3C	World Wide Web Consortium (Konsorcium pro vývoj webových standardů)
WebDAV	Web-based Distributed Authoring and Versioning (Množina rozšíření HTTP)
XML	Extensible Markup Language (Rozšiřitelný značkovací jazyk)
XPath	XML Path Language (Počítačový jazyk, pomocí kterého lze adresovat části XML dokumentu)
XQuery	Dotazovací jazyk pro XML
XSL	eXtensible Stylesheet Language (Rozšiřitelný stylový jazyk)

SEZNAM OBRÁZKŮ

Obrázek 1 - Procesor XML Schema [5]	23
Obrázek 2 – Vztahy mezi uzly ve stromové reprezentaci XML dokumentu. [8]	25
Obrázek 3 - Princip zpracování XML dokumentů pomocí XSLT stylu. [8]	26
Obrázek 4 - Architektura Oracle XML DB: XMLType a Repositář [5]	32
Obrázek 5 – Logo softwaru Oracle SQL Developer.....	34
Obrázek 6 – Okno pro vytvoření nového připojení v Oracle SQL Developer	35
Obrázek 7 – Screen z vývojového prostředí Oracle SQL Developer	36
Obrázek 8 – Ukázka stránky z uživatelské příručky.....	39
Obrázek 9 – Ukázka vytvořeného schématu v Oracle.....	40
Obrázek 10 – Ukázka Xpath dotazu nad XML daty v Oracle	41
Obrázek 11 – Ukázka vytvoření tabulky z příručky v Oracle	43
Obrázek 12 – Ukázka tvorby XML z dat uložených v tabulce Oracle	44

SEZNAM PŘÍLOH

Příloha P I: Na přiloženém CD disku se nachází tato práce včetně přílohy v elektronické podobě

Příloha P II: Uživatelská příručka – XML v databázi Oracle (84 stran)