

XML databázy

Jana Dvořáková

3.12.2010

Pokročilé databázové technológie, FIIT STU



IT řešení, které přináší ovoce

Obsah

XML a XML databáza

Design XML databázy

Dotazovanie nad XML databázou

Typy XML databáz

Zhrnutie a diskusia



XML a XML databáza

<?xml?>

XML

Štruktúrovaný formát pre dáta

- W3C Recommendation (2008, Fifth edition)
- XML dokument = strom

Element

- Logická časť XML dokumentu
- Ohraničený **tagmi**
- Sémantika elementu je popísaná v **názve tagu**

XML databáza

Dáta sú sprístupnené v XML formáte

Logický dátový model: XML dátový model

Dotazovanie: XML dotazovacie jazyky (XPath, XQuery)

„*Natívne XML databázy vs.
Relačné databázy s podporou XML*“

Rozdiely sa
strácajú

Zoznam produktov



NEW
FRONTIER
SLOVAKIA

<ROOT>

<PRODUCT_record>

<PRODUCT_ID>190061</PRODUCT_ID>

<PRODUCT_CODE>BU100</PRODUCT_CODE>

<PRODUCT_NAME>Bežný účet</PRODUCT_NAME>

</PRODUCT_record>

<PRODUCT_record>

<PRODUCT_ID>190062</PRODUCT_ID>

<PRODUCT_CODE>CARD091</PRODUCT_CODE>

<PRODUCT_NAME>VISA</PRODUCT_NAME>

</PRODUCT_record>

<PRODUCT_record>

<PRODUCT_ID>190063</PRODUCT_ID>

<PRODUCT_CODE>IB121</PRODUCT_CODE>

<PRODUCT_NAME>InternetBanking</PRODUCT_NAME>

</PRODUCT_record>

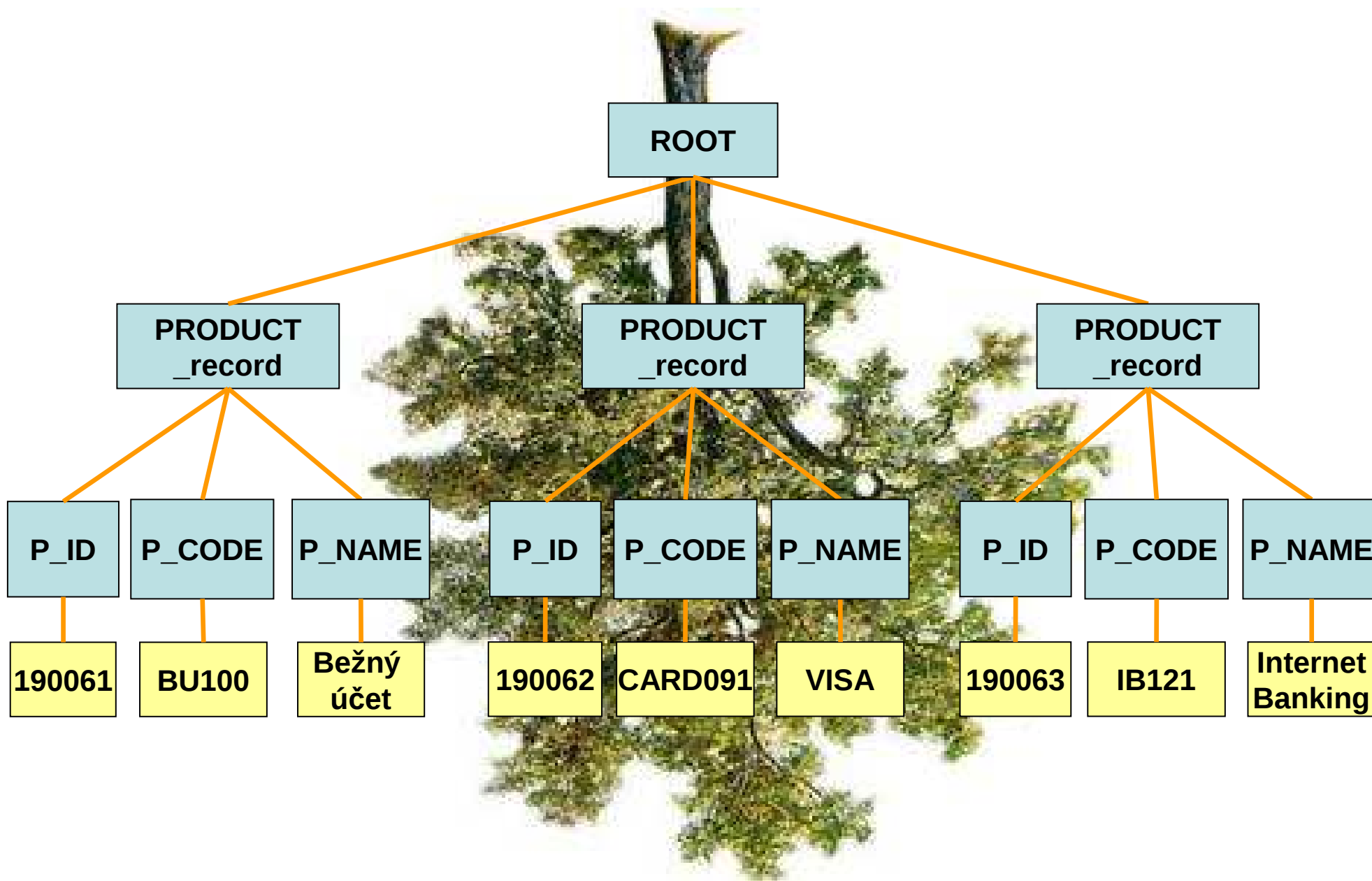
</ROOT>

Počiatočný
tag

Koncový
tag

XML
Element

Zoznam produktov – XML strom



Obsah

XML a XML databáza

Design XML databázy

Dotazovanie nad XML databázou

Typy XML databáz

Zhrnutie a diskusia



XML schéma

- Predpis štruktúry XML dokumentu
- Štruktúra XML dokumentu zodpovedá predpisu ⇒ XML dokument je **validný** voči schéme
- Rôzne jazyky
 - DTD – súčasť špecifikácia XML
 - XSD (XML Schema) – samostatný W3C štandard
 - Relax NG, ..

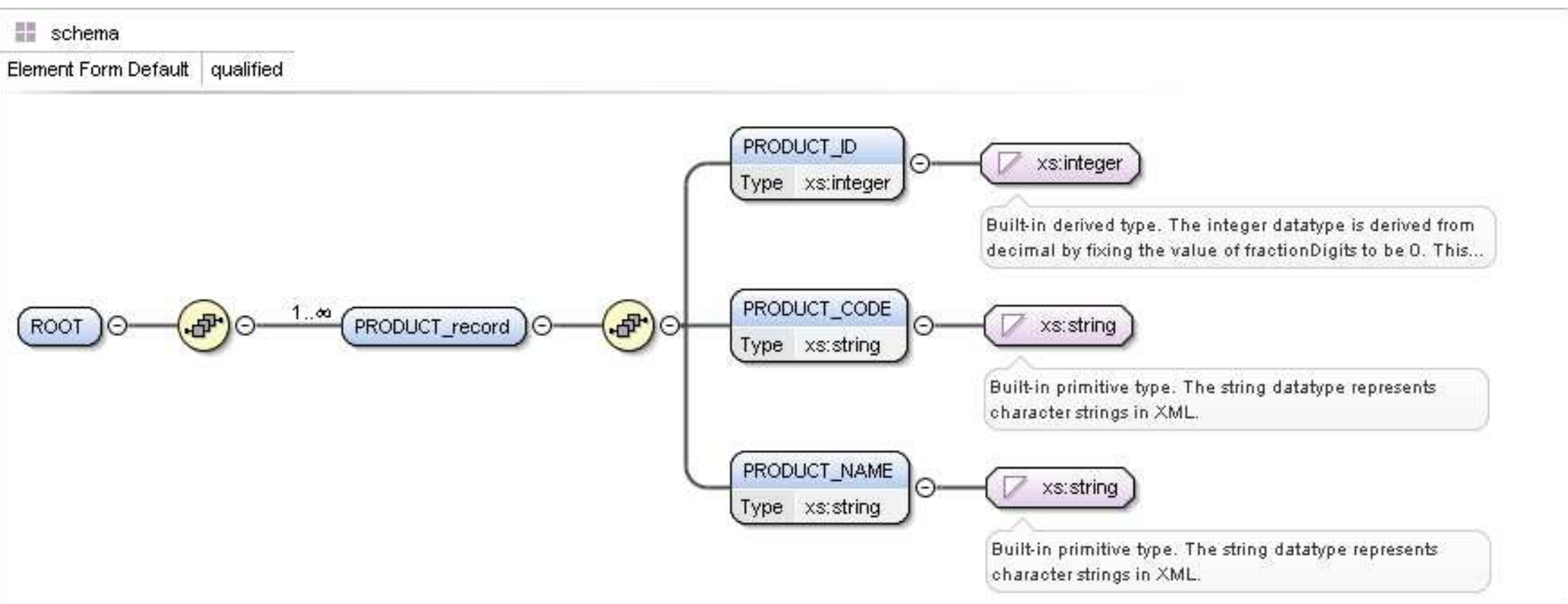
XML Schema - príklad

```
<xs:schema>
  <xs:element name="ROOT">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PRODUCT_record"
          minOccurs="1"
          maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="PRODUCT_ID" type="xs:integer"/>
              <xs:element name="PRODUCT_CODE" type="xs:string"/>
              <xs:element name="PRODUCT_NAME" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Obsah
elementu
ROOT

Obsah elementu
PRODUCT_record

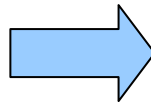
XML Schema - príklad



XML vs. relačné tabuľky

- Je jednoduché previesť relačnú tabuľku do XML:

ATTRIBUTE1	ATTRIBUTE2
value1	value2
...	...



```
<TABLE>
  <ROW>
    <ATT1>value1</ATT1>
    <ATT2>value2</ATT2>
  </ROW>
  ...
</TABLE>
```

- Ale nie vždy naopak! XML umožňuje navyše:
 - Vnorení typov 1:N
 - Priame vyjadrenie relácií N:M (špeciálny prípad vnorenia typov)

Vnorené typy

XML: Umožňuje vytvárať viac úrovní hierarchie

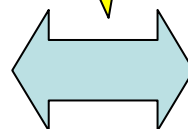
Relačný model: Iba Flat dáta, vnorené typy sú samostatné tabuľky

```

<TABLE>
  <ROW>
    <ATT1>value1</ATT1>
    <NESTED_TYPE>
      <ATT2>value21</ATT2>
      <ATT3>value31</ATT3>
    </NESTED_TYPE>
    <NESTED_TYPE>
      <ATT2>value22</ATT2>
      <ATT3>value32</ATT3>
    </NESTED_TYPE>
  </ROW>
  ...
</TABLE>

```

„Shredding“

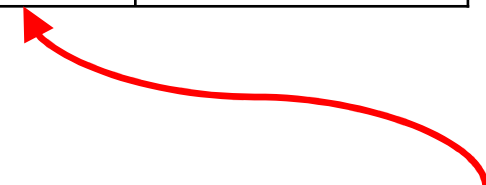


TABLE

ROW_ID	ATTRIBUTE1
100	value1
	...

NESTED_TYPE

NESTED_TYPE_ID	ATTRIBUTE2	ATTRIBUTE3	ROW_ID
1000	value21	value31	100
1000	value22	value32	100



Relácie N:M

Relačný model: relácie N:M cez vzťahové tabuľky

R_TABLE1_TABLE2

TABLE1_TABLE2_ID	TABLE1_ID	TABLE2_ID
10000	100	1000
20000	100	2000

1:N - záznam z jednej tabuľky je vo vzťahu s N záznamami z druhej tabuľky
N:M – viacnásobný vzťah platí obojstranne

TABLE1

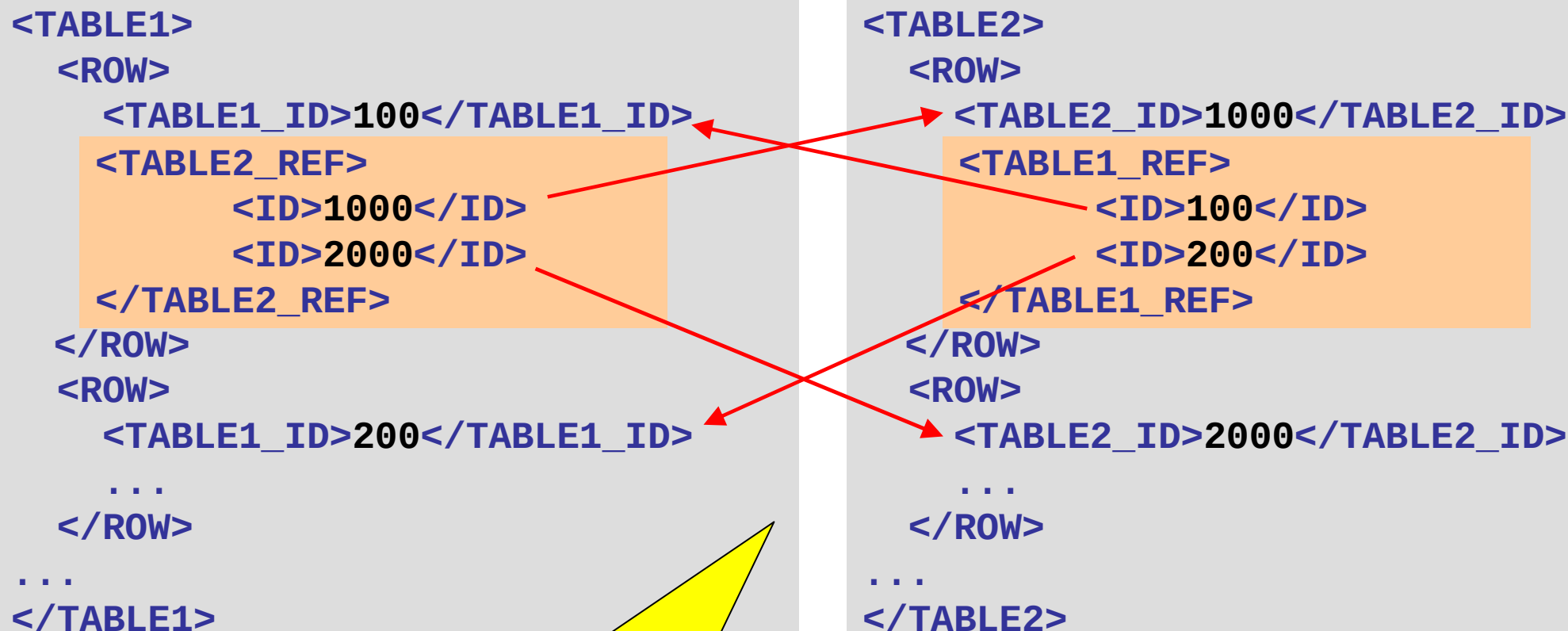
TABLE1_ID
100
200

TABLE2

TABLE2_ID
1000
2000

Relácie 1:N, N:M

XML: Relácie 1:N, N:M je možné modelovať priamo



Duplikované dáta, reálne
zvoliť jeden spôsob

Zoznam produktov



NEW
FRONTIER
SLOVAKIA

<ROOT>

```
<PRODUCT_record>
  <PRODUCT_ID>190061</PRODUCT_ID>
  <PRODUCT_CODE>BU100</PRODUCT_CODE>
  <PRODUCT_NAME>Bežný účet</PRODUCT_NAME>
  <PRODUCT_RELATED>
    <PRODUCT_RELATED_ID>190062</PRODUCT_RELATED_ID>
    <PRODUCT_RELATED_ID>190063</PRODUCT_RELATED_ID>
  </PRODUCT_RELATED>
</PRODUCT_record>

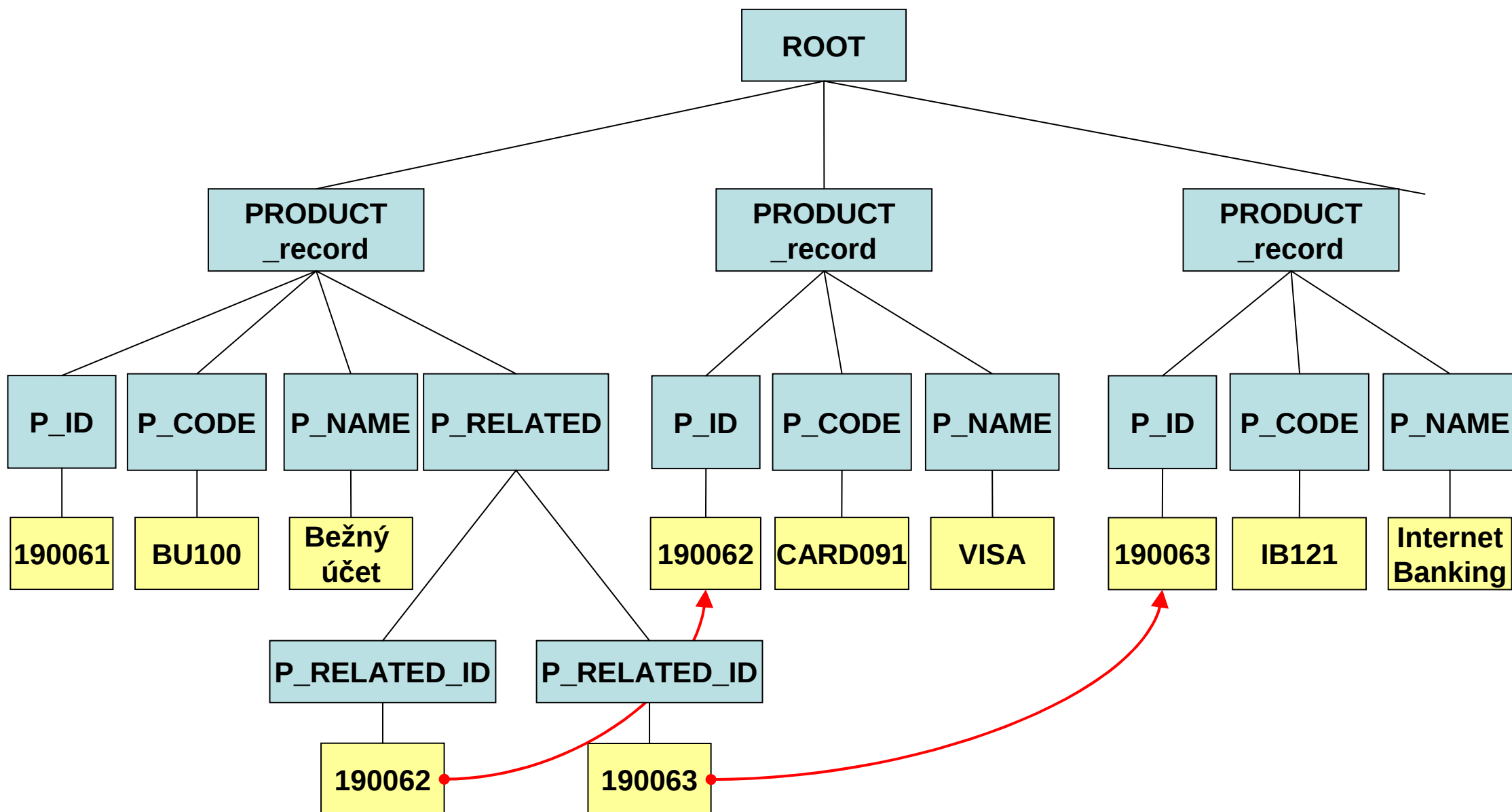
<PRODUCT_record>
  <PRODUCT_ID>190062</PRODUCT_ID>
  <PRODUCT_CODE>CARD091</PRODUCT_CODE>
  <PRODUCT_NAME>VISA</PRODUCT_NAME>
</PRODUCT_record>

<PRODUCT_record>
  <PRODUCT_ID>190063</PRODUCT_ID>
  <PRODUCT_CODE>IB121</PRODUCT_CODE>
  <PRODUCT_NAME>InternetBanking</PRODUCT_NAME>
</PRODUCT_record>
```

Doplnkové
produkty, ktoré
sa ponúkajú k
danému produktu

</ROOT>

Zoznam produktov – XML strom



Zoznam produktov – relačná DB

PRODUCT

PRODUCT_ID	PRODUCT_CODE	PRODUCT_NAME
190061	BU100	Bežný účet
190062	CARD091	VISA
190063	IB121	InternetBanking
...	...	...

V relačnej DB musíme vytvoriť dve tabuľky:

1. PRODUCT
2. R_PRODUCT_RELATED_PRODUCT

PRODUCT_RELATED_PRODUCT_ID	PRODUCT_ID	PRODUCT_RELATED_ID
100000	190061	190062
100001	190061	190063
...	...	...

R_PRODUCT_RELATED_PRODUCT

Mapovania REL $\leftrightarrow$ XML

- Mapovanie **REL** $\Rightarrow$ **XML**
 - Je možné automaticky, ale je potrebné mať definovaný výstupný formát
- Mapovanie **XML** $\Rightarrow$ **REL**
 - Je možné automaticky – pomocou „shredding“-u
 - Pokrýva prípad vnorenia typov aj mapovaní 1:M, N:M
 - Často menej efektívne ako manuálne mapovanie
- Mapovanie sa definuje na úrovni schémy (často anotovaná XML Schema)

XML a XML databáza

Design XML databázy

Dotazovanie nad XML databázou

Typy XML databáz

Zhrnutie a diskusia



XQuery

Dotazovací jazyk nad XML

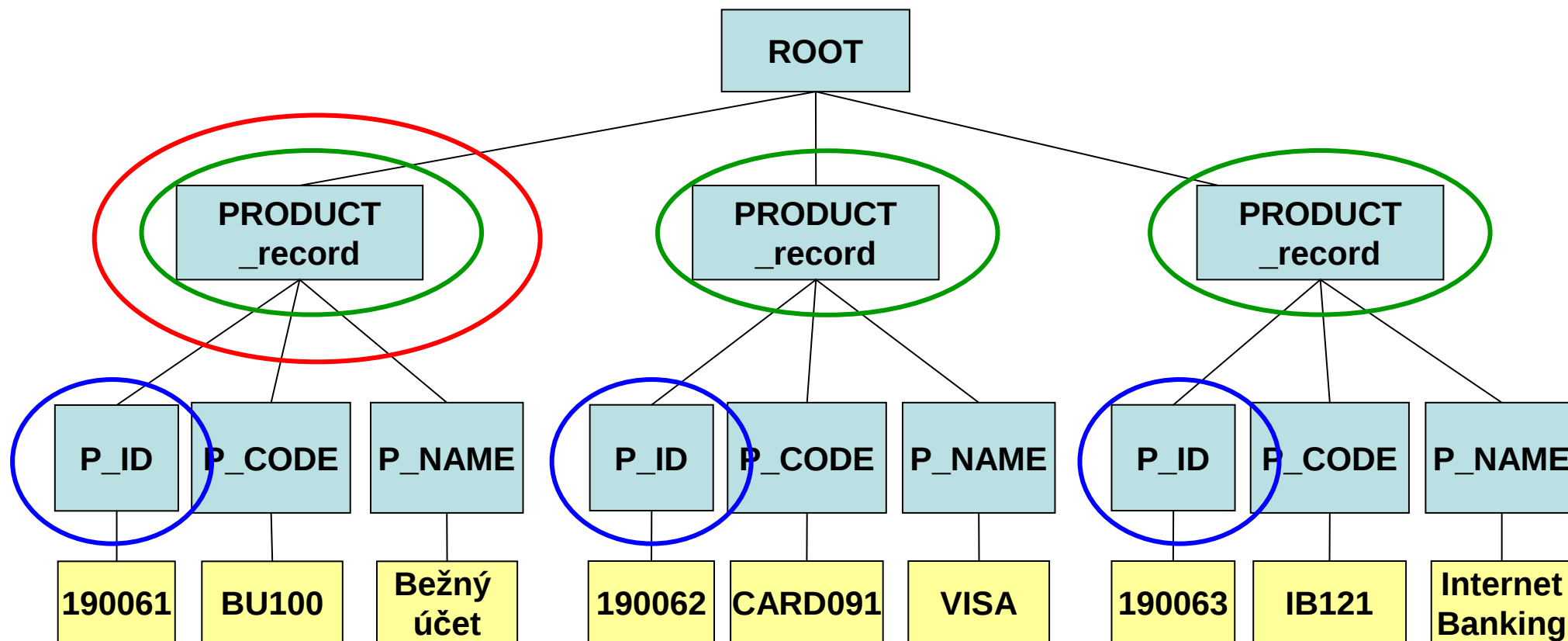
- W3C Recommendation (2007)
- Umožňuje transformácie
 - XML-to-XML
 - XML-to-(CSV,TXT, ..)
- XQuery je pre XML to, čo je SQL pre relačné tabuľky
- **Výnimka:** XQuery samotné umožňuje iba DML operácie (nie DDL operácie)
- Alternatíva: Transformačný jazyk XSLT

Základom XQuery je XPath

- Navigačný jazyk
- Pri práci s XML dokumentom je potrebné v ňom „cestovať“
- Vyberá časti XML dokumentu (typicky množinu elementov)
- Spôsob výberu je podobný navigácií v adresárovej štruktúre operačného systému



Ako funguje XPath



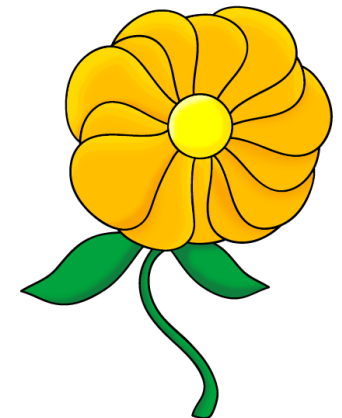
/ROOT/PRODUCT_record/PRODUCT_ID

/ROOT/PRODUCT_record[1]

//PRODUCT_record

... a mnoho ďalších operátorov. **V SQL stačí názov stĺpca.**

- **Dotazovací** jazyk, „nadstavba“ nad XPath
- Funcionalita naviac
 - Konštruovanie výstupu, užívateľské funkcie, ...
- ➔ Plnohodnotný (i keď pre SQL používateľov neintuitívny) dotazovací jazyk
- Základom sú „**FLWOR**“ výrazy
 - **F**OR, **L**ET, **W**HERE. **O**RDER BY, **R**ETURN
 - V XQuery znamená to, čo v SQL „select“



XQuery FLWOR vs. SQL select - zadanie

- **Vstup:** Tabuľky (XML dáta) PRODUCT a PRODUCT_TYPE
- **Výstup:** Zoznam mien všetkých produktov, ktorých typ je „Vkladový účet“

Tabuľka **PRODUCT**

PRODUCT_ID	PRODUCT_NAME	PRODUCT_TYPE_ID
10002	Bežný účet	100
...	...	...

Tabuľka **PRODUCT_TYPE**

PRODUCT_TYPE_ID	PRODUCT_TYPE_NAME
100	Vkladový produkt
...	...

Dotaz musí byť všeobecný, t.j. nesmie sa odkazovať sa na konkrétne ID

XML vstup

```
<ROOT>
  <PRODUCT_record>
    <PRODUCT_ID>10002</PRODUCT_ID>
    <PRODUCT_NAME>Bežný účet</PRODUCT_NAME>
    <PRODUCT_TYPE_ID>100</PRODUCT_TYPE_ID>
  </PRODUCT_record>
  ...
</ROOT>
```

```
<ROOT>
  <PRODUCT_TYPE_record>
    <PRODUCT_TYPE_ID>100</PRODUCT_TYPE_ID>
    <PRODUCT_TYPE_NAME>Vkladový produkt</PRODUCT_TYPE_NAME>
  </PRODUCT_TYPE_record>
  ...
</ROOT>
```

XQuery FLWOR vs. SQL select - porovnanie



NEW
FRONTIER
SLOVAKIA

```
select p.product_name
```

```
from product p, product_type pt
```

```
where p.product_type_id = pt.product_type_id and  
       pt.product_type_name = "Vkladový produkt"
```

Premenné **\$product**
a **\$product_type**
obsahujú naviazané
XML dáta

```
for $p in $product/ROOT/PRODUCT_record  
for $pt in $product_type/ROOT/PRODUCT_TYPE_record
```

```
where $p/PRODUCT_TYPE_ID/text() =  
       $pt/PRODUCT_TYPE_ID/text()  
and $pt/PRODUCT_TYPE_NAME/text() = "Vkladový produkt"
```

```
return $p/PRODUCT_NAME
```

Obsah

XML a XML databáza

Design XML databázy

Dotazovanie nad XML databázou

Typy XML databáz

Zhrnutie a diskusia



Natívne XML DB - vlastnosti

- Definujú logický model pre XML dokument
- Ukladajú a poskytujú dáta na základe tohto modelu
- XML dokument = základná jednotka
- Nie sú žiadne požiadavky na fyzický model (môže to byť aj relačná DB)
 - Ukladanie dát na úrovni dokumentu
 - Reflektuje presnú formu dokumentu
 - Pomalšia manipulácia
 - Ukladanie dát na úrovni uzlov
 - Reflektuje iba štruktúru dokumentu na úrovni modelu
 - Rýchlejšia manipulácia

Štandardné API

- **XML:DB API** – API pre XML databázy
- **XQJ** – XQuery API for Java

Natívne XML DB - príklady

eXist

- Open Source
- Proprietárne uloženie dát (B+ stromy, paged files)
- XQuery, XSLT, XPath, XInclude, XPointer, Xupdate, XQuery Update
- XML:DB
- Indexovanie

Sedna

- Open Source
- Proprietárne uloženie dát (schema-based clustering)
- XQuery, XPath, vlastný update mechanizmus
- XML:DB, XQJ + vlastné API pre rôzne jazyky
- Indexovanie (založené na B-stromoch), trigger

DB s podporou XML - vlastnosti



NEW
FRONTIER
SLOVAKIA

“XML-enabled” DB

- Tradičné relačné DB s podporou XML
- Rozdiel oproti „natívnymi XML DB“ sa postupne stráca

SQL/XML,

= rozšírenie SQL o podporu práce s XML

- Štandard ISO/IEC 9075-14:2003, definuje
 - XML dátový typ
 - Funkcie na konštrukciu XML
 - XMLELEMENT, XMLATTRIBUTES, XMLFOREST, XMLCONCAT,...
 - Funkcie na použitie XQuery v SQL
 - XMLQUERY
 - XMLTABLE

DB s podporou XML - príklady



NEW
FRONTIER
SLOVAKIA

	Oracle 11g	DB2	MS SQL Server 2008 R2	PotgreSQL 9.0	MySQL 5.1
XML dátový typ	ÁNO	ÁNO	ÁNO	ÁNO	NIE
SQL/XML	ÁNO	ÁNO	ÁNO	čiastočne	NIE
XML Schema	ÁNO	ÁNO	ÁNO	NIE	NIE
XQuery	ÁNO	ÁNO	ÁNO	NIE	NIE
XPath	ÁNO	ÁNO	ÁNO	čiastočne	čiastočne
XML update	ÁNO	ÁNO	ÁNO	čiastočne	čiastočne
XML generovanie	ÁNO	ÁNO	ÁNO	ÁNO	NIE
REL->XML mapovanie	ÁNO	ÁNO	ÁNO	ÁNO	NIE
XML->REL mapovanie	ÁNO	ÁNO	ÁNO	NIE	NIE
XML indexovanie	ÁNO	ÁNO	ÁNO	NIE	NIE

XML a XML databáza

Design XML databázy

Dotazovanie nad XML databázou

Typy XML databáz

Zhrnutie a diskusia



XML databáza – pre a proti

PRE



- Širšie možnosti pri modelovaní
 - Jednoduchšie modelovanie zložitých vzťahov
 - Jednoduchší dátový model (menej „tabuliek“)
- Jednoduchá lokálna manipulácia s dátami
 - Stačí mať k dispozícii XQuery procesor
- GUI
 - Možnosť automaticky generovať z modelu užívateľsky príjemné GUI

PROTI



- Pomalosť
 - Nedosahuje rýchlosť transakčného systému relačných DB
 - Problémy s väčším počtom záznamov
- Nová technológia
 - Neustálená, štandardné postupy iba vo vývoji
 - Pre väčšinu vývojárov nie je dôverne známa
 - Nutnosť naučiť sa myslieť v XML doméne
- DDL operácie
 - Bez podpory pre DDL operácie („XQuery update“?)

**Rozhodnutie závislé
na type dát a
spôsobe ich použitia**

Diskusia

