

IV1023 vt2022

Avancerad Datahantering med XML

SQL/XML

nikos dimitrakas
nikosd@kth.se
08-161295
Rum 2423

Läsanvisningar
Kapitel 8.3 och 15 i kursboken

SQL/XML

- **Stöd för XML i relationsdatabaser enligt SQL 2003, SQL 2006 och SQL 2008**
 - SQL/XML: del av SQL-standarden
 - SQLX: gruppen bakom SQL/XML
 - SQLXML: Microsofts XML-tillägg för SQL
- **XML-datatyp**
- **Komposition av XML (XML som resultat av SQL-SELECT-satser)**
- **Stöd för exekvering av XQuery**
- **Stöd för interoperabilitet mellan XQuery och SQL**
- **Inget speciellt (ännu) om uppdatering av XML**
 - XQuery Update Facility kom i mars 2011

XML-datatyp

- **Lagra XML, inte den serialiserade varianten (som skulle vara en vanlig CLOB)**
 - Enligt XQuery sedan SQL 2006
- **Varje cell är ett XML-dokument (eventuellt fragment)**
- **Stöd för validering**

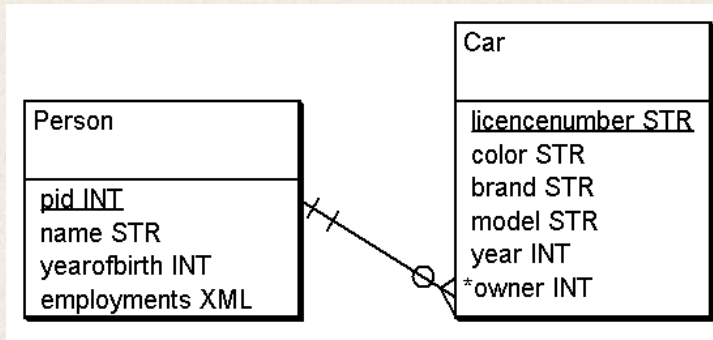
SQL-funktioner

- | | |
|------------------------|-----------------------|
| • XMLELEMENT | • XMLQUERY |
| • XMLATTRIBUTES | • XMLTABLE |
| • XMLFOREST | • XMLEXISTS |
| • XMLCONCAT | |
| • XMLCOMMENT | • XMLSERIALIZE |
| • XMLPI | • XMLPARSE |
| • XMLNAMESPACES | • XMLCAST |
| • XMLAGG | |
| • XMLTEXT | • XMLVALIDATE |
| • XMLDOCUMENT | |

Exempeldata

PERSON

pid	name	yearofbirth
1	John Higgins	1975
2	Steven Hendry	1973
3	Mathew Stevens	1982
4	Ronnie O'Sullivan	1980
5	Ken Doherty	1974
6	Steve Davis	1960
7	Paul Hunter	1983
8	Neil Robertson	1982



CAR

licencenumber	color	brand	model	year	owner
ABC123	black	NISSAN	Cherry	1995	1
CCD457	blue	FIAT	Forza	2001	2
DKL998	green	SAAB	9000C	1998	3
RSQ199	black	NISSAN	Micra	1999	4
WID387	red	FIAT	Nova	2003	5
ROO197	blue	SAAB	900i	1982	3
TYD226	black	NISSAN	Cherry	1990	1
PTF357	red	VOLVO	V70	2001	6

5

Exempeldata

- Kolumnen employments enligt följande DTD:

<!ELEMENT root (employment*)>

<!ELEMENT employment EMPTY>

<!ATTLIST employment

startdate CDATA #REQUIRED

enddate CDATA #IMPLIED

employer CDATA #REQUIRED>

6

Exempeldata

• Kolumnen employments enligt följande XML Schema:

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="root">
    <complexType>
      <sequence>
        <element name="employment" type="EmploymentType"
          minOccurs="0" maxOccurs="unbounded" />
      </sequence>
    </complexType>
  </element>
  <complexType name="EmploymentType">
    <attribute name="startdate" type="date" use="required" />
    <attribute name="enddate" type="date" use="optional" />
    <attribute name="employer" type="string" use="required" />
  </complexType>
</schema>
```

7

Exempeldata

pid	employments
1	<root><employment startdate="2001-08-20" enddate="2009-02-28" employer="ABB"/> <employment startdate="2009-04-15" employer="UPC"/></root>
2	<root><employment startdate="2002-08-20" enddate="2003-06-30" employer="ABB"/> <employment startdate="2003-08-01" employer="UPC"/> <employment startdate="2006-11-01" employer="ABB"/></root>
3	<root><employment startdate="2003-01-10" employer="UPC"/> </root>
4	<root><employment startdate="2002-03-10" enddate="2010-05-22" employer="LKP"/> <employment startdate="2010-08-15" employer="STG"/></root>
5	<root><employment startdate="2002-02-12" enddate="2003-05-11" employer="LKP"/> <employment startdate="2003-05-12" enddate="2003-12-02" employer="ABB"/> <employment startdate="2003-12-06" enddate="2005-02-17" employer="LKP"/> <employment startdate="2005-02-18" enddate="2008-05-16" employer="FFD"/> <employment startdate="2008-06-02" employer="STG"/></root>
6	<root><employment startdate="2001-01-05" enddate="2005-12-31" employer="ABB"/> <employment startdate="2006-01-15" enddate="2009-01-22" employer="LKP"/> <employment startdate="2009-02-01" employer="FFD"/></root>
7	<root><employment startdate="2004-01-10" enddate="2008-09-29" employer="FFD"/> <employment startdate="2008-10-01" enddate="2010-11-20" employer="LKP"/></root>
8	<root><employment startdate="2006-02-03" enddate="2008-10-30" employer="UPC"/> <employment startdate="2008-11-20" employer="ABB"/></root>

8

XMLELEMENT

- Skapar ett XML-element
 - Ange elementets namn efter NAME
 - Ange elementets innehåll

```
SELECT XMLELEMENT(NAME "Person", name)
FROM Person
```

```
<Person>John Higgins</Person>
<Person>Stephen Hendry</Person>
<Person>Matthew Stevens</Person>
<Person>Ronnie O'Sullivan</Person>
<Person>Ken Doherty</Person>
<Person>Steve Davis</Person>
<Person>Paul Hunter</Person>
<Person>Neil Robertson</Person>
```

XMLTEXT

- Skapar en XML-textnod

```
SELECT XMLELEMENT(NAME "Person", XMLTEXT(name))
FROM Person
```

Samma som

```
SELECT XMLELEMENT(NAME "Person", name)
FROM Person
```

XMLDOCUMENT

- Skapar en XML-dokumentnod

```
SELECT XMLDOCUMENT(XMLELEMENT(NAME "Person", name))  
FROM Person
```

Samma som

```
SELECT XMLELEMENT(NAME "Person", name)  
FROM Person
```

XMLATTRIBUTES

- Skapar XML-attribut
 - Används inuti XMLELEMENT

```
SELECT XMLELEMENT(NAME "Person", XMLATTRIBUTES(yearofbirth), name)  
FROM Person
```

```
<Person YEAROFBIRTH="1975">John Higgins</Person>  
<Person YEAROFBIRTH="1973">Stephen Hendry</Person>  
<Person YEAROFBIRTH="1982">Matthew Stevens</Person>  
<Person YEAROFBIRTH="1980">Ronnie O'Sullivan</Person>  
<Person YEAROFBIRTH="1974">Ken Doherty</Person>  
<Person YEAROFBIRTH="1960">Steve Davis</Person>  
<Person YEAROFBIRTH="1983">Paul Hunter</Person>  
<Person YEAROFBIRTH="1982">Neil Robertson</Person>
```

XMLATTRIBUTES

- Kan användas för att skapa flera attribut
- Kan bestämma attributnamnen

```
SELECT XMLELEMENT(NAME "Person",  
                  XMLATTRIBUTES(yearofbirth AS "FödelseÅr",  
                                (SELECT COUNT(*)  
                                 FROM Car  
                                 WHERE owner = pid) AS "AntalBilar"),  
                  name)  
FROM Person
```

```
<Person FödelseÅr="1975" AntalBilar="2">John Higgins</Person>  
<Person FödelseÅr="1973" AntalBilar="1">Stephen Hendry</Person>  
<Person FödelseÅr="1982" AntalBilar="2">Matthew Stevens</Person>  
<Person FödelseÅr="1980" AntalBilar="1">Ronnie O'Sullivan</Person>  
<Person FödelseÅr="1974" AntalBilar="1">Ken Doherty</Person>  
<Person FödelseÅr="1960" AntalBilar="1">Steve Davis</Person>  
<Person FödelseÅr="1983" AntalBilar="0">Paul Hunter</Person>  
<Person FödelseÅr="1982" AntalBilar="0">Neil Robertson</Person>
```

13

XMLELEMENT utan innehåll

- Behöver inte ha innehåll
- Behöver inte ha attribut

```
SELECT XMLELEMENT(NAME "Person")  
FROM Person
```

```
<Person/>  
<Person/>  
<Person/>  
<Person/>  
<Person/>  
<Person/>  
<Person/>  
<Person/>
```

```
SELECT XMLELEMENT(NAME "Person", XMLATTRIBUTES(yearofbirth, name))  
FROM Person
```

```
<Person YEAROFBIRTH="1975" NAME="John Higgins"/>  
<Person YEAROFBIRTH="1973" NAME="Stephen Hendry"/>  
<Person YEAROFBIRTH="1982" NAME="Matthew Stevens"/>  
<Person YEAROFBIRTH="1980" NAME="Ronnie O'Sullivan"/>  
<Person YEAROFBIRTH="1974" NAME="Ken Doherty"/>  
<Person YEAROFBIRTH="1960" NAME="Steve Davis"/>  
<Person YEAROFBIRTH="1983" NAME="Paul Hunter"/>  
<Person YEAROFBIRTH="1982" NAME="Neil Robertson"/>
```

14

XMLCONCAT

- Slår ihop två XML-värden (kolumner) till ett
 - Varje XMLELEMENT skulle ge en kolumn (ett värde per rad)

- Utan:

```
SELECT XMLELEMENT(NAME "Namn", name),  
       XMLELEMENT(NAME "FödelseÅr", yearofbirth)  
FROM Person
```

<Namn>John Higgins</Namn>	<FödelseÅr>1975</FödelseÅr>
<Namn>Stephen Hendry</Namn>	<FödelseÅr>1973</FödelseÅr>
<Namn>Matthew Stevens</Namn>	<FödelseÅr>1982</FödelseÅr>
<Namn>Ronnie O'Sullivan</Namn>	<FödelseÅr>1980</FödelseÅr>
<Namn>Ken Doherty</Namn>	<FödelseÅr>1974</FödelseÅr>
<Namn>Steve Davis</Namn>	<FödelseÅr>1960</FödelseÅr>
<Namn>Paul Hunter</Namn>	<FödelseÅr>1983</FödelseÅr>
<Namn>Neil Robertson</Namn>	<FödelseÅr>1982</FödelseÅr>

Obs! resultatet har två kolumner

15

XMLCONCAT

- Med:

```
SELECT XMLCONCAT(XMLELEMENT(NAME "Namn", name),  
                  XMLELEMENT(NAME "FödelseÅr", yearofbirth))  
FROM Person
```

```
<Namn>John Higgins</Namn><FödelseÅr>1975</FödelseÅr>  
<Namn>Stephen Hendry</Namn><FödelseÅr>1973</FödelseÅr>  
<Namn>Matthew Stevens</Namn><FödelseÅr>1982</FödelseÅr>  
<Namn>Ronnie O'Sullivan</Namn><FödelseÅr>1980</FödelseÅr>  
<Namn>Ken Doherty</Namn><FödelseÅr>1974</FödelseÅr>  
<Namn>Steve Davis</Namn><FödelseÅr>1960</FödelseÅr>  
<Namn>Paul Hunter</Namn><FödelseÅr>1983</FödelseÅr>  
<Namn>Neil Robertson</Namn><FödelseÅr>1982</FödelseÅr>
```

Resultatet har en kolumn

16

XMLFOREST

- Skapar flera enkla element med en gång

```
SELECT XMLFOREST(name AS "Namn",  
                  yearofbirth AS "FödelseÅr")  
FROM Person
```

```
<Namn>John Higgins</Namn><FödelseÅr>1975</FödelseÅr>  
<Namn>Stephen Hendry</Namn><FödelseÅr>1973</FödelseÅr>  
<Namn>Matthew Stevens</Namn><FödelseÅr>1982</FödelseÅr>  
<Namn>Ronnie O'Sullivan</Namn><FödelseÅr>1980</FödelseÅr>  
<Namn>Ken Doherty</Namn><FödelseÅr>1974</FödelseÅr>  
<Namn>Steve Davis</Namn><FödelseÅr>1960</FödelseÅr>  
<Namn>Paul Hunter</Namn><FödelseÅr>1983</FödelseÅr>  
<Namn>Neil Robertson</Namn><FödelseÅr>1982</FödelseÅr>
```

17

Nästla

```
SELECT XMLELEMENT(NAME "Person",  
                  XMLATTRIBUTES(pid AS "ID"),  
                  XMLELEMENT(NAME "Info",  
                              XMLATTRIBUTES(name AS "Om"),  
                              XMLFOREST((SELECT COUNT(*)  
                                           FROM Car  
                                           WHERE owner = pid) AS "Bilar",  
                                           yearofbirth AS "År")))  
FROM Person
```

```
<Person ID="1"><Info Om="John Higgins"><Bilar>2</Bilar><År>1975</År></Info></Person>  
<Person ID="2"><Info Om="Stephen Hendry"><Bilar>1</Bilar><År>1973</År></Info></Person>  
<Person ID="3"><Info Om="Matthew Stevens"><Bilar>2</Bilar><År>1982</År></Info></Person>  
<Person ID="4"><Info Om="Ronnie O'Sullivan"><Bilar>1</Bilar><År>1980</År></Info></Person>  
<Person ID="5"><Info Om="Ken Doherty"><Bilar>1</Bilar><År>1974</År></Info></Person>  
<Person ID="6"><Info Om="Steve Davis"><Bilar>1</Bilar><År>1960</År></Info></Person>  
<Person ID="7"><Info Om="Paul Hunter"><Bilar>0</Bilar><År>1983</År></Info></Person>  
<Person ID="8"><Info Om="Neil Robertson"><Bilar>0</Bilar><År>1982</År></Info></Person>
```

18

XMLAGG

- Aggregerar flera XML-värden till ett
- En aggregatfunktion (som COUNT, SUM, etc.)
 - Antingen blir alla rader en grupp
 - Eller gruppering med GROUP BY

```
SELECT XMLAGG(XMLELEMENT(NAME "Person", name))  
FROM Person
```

```
<Person>John Higgins</Person><Person>Stephen  
Hendry</Person><Person>Matthew Stevens</Person><Person>Ronnie  
O'Sullivan</Person><Person>Ken Doherty</Person><Person>Steve  
Davis</Person><Person>Paul Hunter</Person><Person>Neil  
Robertson</Person>
```

Obs! Hela resultatet är en rad och en kolumn (en cell)

19

XMLAGG

- Och ett rotelement för att få well-formed XML

```
SELECT XMLELEMENT(NAME "Folk",  
  XMLAGG(XMLELEMENT(NAME "Person", name)))  
FROM Person
```

```
<Folk><Person>John Higgins</Person><Person>Stephen  
Hendry</Person><Person>Matthew  
Stevens</Person><Person>Ronnie O'Sullivan</Person><Person>Ken  
Doherty</Person><Person>Steve Davis</Person><Person>Paul  
Hunter</Person><Person>Neil Robertson</Person></Folk>
```

20

Resultatet indenterat

```
<Folk>
  <Person>John Higgins</Person>
  <Person>Stephen Hendry</Person>
  <Person>Matthew Stevens</Person>
  <Person>Ronnie O'Sullivan</Person>
  <Person>Ken Doherty</Person>
  <Person>Steve Davis</Person>
  <Person>Paul Hunter</Person>
  <Person>Neil Robertson</Person>
</Folk>
```

21

XMLAGG med GROUP BY

- Ett resultat per grupp
 - Aggregerade vs grupperade kolumner (som vanligt)

```
SELECT XMLELEMENT(NAME "Färg",
                  XMLATTRIBUTES(color AS "Namn"),
                  XMLAGG(XMLELEMENT(NAME "Bil", licencenumber)))
FROM Car
GROUP BY color
```

```
<Färg Namn="black"><Bil>ABC123</Bil><Bil>RSQ199</Bil><Bil>TYD226</Bil></Färg>
<Färg Namn="blue"><Bil>CCD457</Bil><Bil>ROO197</Bil></Färg>
<Färg Namn="green"><Bil>DKL998</Bil></Färg>
<Färg Namn="red"><Bil>PTF357</Bil><Bil>WID387</Bil></Färg>
```

22

XMLAGG med GROUP BY

```
SELECT XMLELEMENT(NAME "Färg",
    XMLATTRIBUTES(color AS "Namn"),
    XMLAGG(XMLELEMENT(NAME "Bil",
        XMLFOREST(licencenumber AS "Regnr", name AS "Ägare"))))
FROM Car, Person
WHERE owner = pid
GROUP BY color
```

```
<Färg Namn="black"><Bil><Regnr>ABC123</Regnr><Ägare>John
Higgins</Ägare></Bil><Bil><Regnr>RSQ199</Regnr><Ägare>Ronnie
O'Sullivan</Ägare></Bil><Bil><Regnr>TYD226</Regnr><Ägare>John
Higgins</Ägare></Bil></Färg>
<Färg Namn="blue"><Bil><Regnr>CCD457</Regnr><Ägare>Stephen
Hendry</Ägare></Bil><Bil><Regnr>ROO197</Regnr><Ägare>Ken
Doherty</Ägare></Bil></Färg>
<Färg Namn="green"><Bil><Regnr>DKL998</Regnr><Ägare>Matthew
Stevens</Ägare></Bil></Färg>
<Färg Namn="red"><Bil><Regnr>PTF357</Regnr><Ägare>Steve
Davis</Ägare></Bil><Bil><Regnr>WID387</Regnr><Ägare>Matthew
Stevens</Ägare></Bil></Färg>
```

23

XMLAGG med GROUP BY

```
SELECT XMLELEMENT(NAME "Bilfärger", XMLAGG(färgxml))
FROM (SELECT XMLELEMENT(NAME "Färg",
    XMLATTRIBUTES(color AS "Namn"),
    XMLAGG(XMLELEMENT(NAME "Bil",
        XMLFOREST(licencenumber AS "Regnr",
            name AS "Ägare")))) AS färgxml
FROM Car, Person
WHERE owner = pid
GROUP BY color) AS inretabell
```

```
<Bilfärger><Färg Namn="black"><Bil><Regnr>ABC123</Regnr><Ägare>John
Higgins</Ägare></Bil><Bil><Regnr>RSQ199</Regnr><Ägare>Ronnie
O'Sullivan</Ägare></Bil><Bil><Regnr>TYD226</Regnr><Ägare>John
Higgins</Ägare></Bil></Färg><Färg
Namn="blue"><Bil><Regnr>CCD457</Regnr><Ägare>Stephen
Hendry</Ägare></Bil><Bil><Regnr>ROO197</Regnr><Ägare>Ken
Doherty</Ägare></Bil></Färg><Färg
Namn="green"><Bil><Regnr>DKL998</Regnr><Ägare>Matthew
Stevens</Ägare></Bil></Färg><Färg
Namn="red"><Bil><Regnr>PTF357</Regnr><Ägare>Steve
Davis</Ägare></Bil><Bil><Regnr>WID387</Regnr><Ägare>Matthew
Stevens</Ägare></Bil></Färg></Bilfärger>
```

24

Resultatet indenterat

<Bilfärger>

<Färg Namn="black">

<Bil>

<Regnr>ABC123</Regnr>

<Ägare>John Higgins</Ägare>

</Bil>

<Bil>

<Regnr>RSQ199</Regnr>

<Ägare>Ronnie O'Sullivan</Ägare>

</Bil>

<Bil>

<Regnr>TYD226</Regnr>

<Ägare>John Higgins</Ägare>

</Bil>

</Färg>

<Färg Namn="blue">

<Bil>

<Regnr>CCD457</Regnr>

<Ägare>Stephen Hendry</Ägare>

</Bil>

<Bil>

<Regnr>ROO197</Regnr>

<Ägare>Ken Doherty</Ägare>

</Bil>

</Färg>

><Färg Namn="green">

<Bil>

<Regnr>DKL998</Regnr>

<Ägare>Matthew Stevens</Ägare>

</Bil>

</Färg>

<Färg Namn="red">

<Bil>

<Regnr>PTF357</Regnr>

<Ägare>Steve Davis</Ägare>

</Bil>

<Bil>

<Regnr>WID387</Regnr>

<Ägare>Matthew Stevens</Ägare>

</Bil>

</Färg>

</Bilfärger>

25

XMLCOMMENT

- Skapar XML-kommentarer

**SELECT XMLCOMMENT('Vilket väder...')
FROM Person**

<!--Vilket väder...-->

<!--Vilket väder...-->

<!--Vilket väder...-->

<!--Vilket väder...-->

<!--Vilket väder...-->

<!--Vilket väder...-->

<!--Vilket väder...-->

<!--Vilket väder...-->

26

XMLCOMMENT dynamisk

```
SELECT XMLELEMENT(NAME "Folk",  
    XMLAGG(XMLCONCAT(XMLCOMMENT('Nu kommer  
elementet för person med pid=' CONCAT pid),  
        XMLFOREST(name AS "Person"))))  
FROM Person
```

```
<Folk><!--Nu kommer elementet för person med pid=1--><Person>John  
Higgins</Person><!--Nu kommer elementet för person med pid=2--  
><Person>Stephen Hendry</Person><!--Nu kommer elementet för  
person med pid=3--><Person>Matthew Stevens</Person><!--Nu kommer  
elementet för person med pid=4--><Person>Ronnie  
O'Sullivan</Person><!--Nu kommer elementet för person med pid=5--  
><Person>Ken Doherty</Person><!--Nu kommer elementet för person  
med pid=6--><Person>Steve Davis</Person><!--Nu kommer elementet  
för person med pid=7--><Person>Paul Hunter</Person><!--Nu kommer  
elementet för person med pid=8--><Person>Neil  
Robertson</Person></Folk>
```

27

Resultatet indenterat

```
<Folk>  
  <!--Nu kommer elementet för person med pid=1-->  
  <Person>John Higgins</Person>  
  <!--Nu kommer elementet för person med pid=2-->  
  <Person>Stephen Hendry</Person>  
  <!--Nu kommer elementet för person med pid=3-->  
  <Person>Matthew Stevens</Person>  
  <!--Nu kommer elementet för person med pid=4-->  
  <Person>Ronnie O'Sullivan</Person>  
  <!--Nu kommer elementet för person med pid=5-->  
  <Person>Ken Doherty</Person>  
  <!--Nu kommer elementet för person med pid=6-->  
  <Person>Steve Davis</Person>  
  <!--Nu kommer elementet för person med pid=7-->  
  <Person>Paul Hunter</Person>  
  <!--Nu kommer elementet för person med pid=8-->  
  <Person>Neil Robertson</Person>  
</Folk>
```

28

XMLPI

- Skapar XML processing instructions

```
SELECT XMLPI(NAME "hej", 'till="alla glada"')  
FROM Person
```

```
<?hej till="alla glada"?>  
<?hej till="alla glada"?>  
<?hej till="alla glada"?>  
<?hej till="alla glada"?>  
<?hej till="alla glada"?>  
<?hej till="alla glada"?>  
<?hej till="alla glada"?>  
<?hej till="alla glada"?>
```

XMLPI

- Behöver inte se ut som attribut

```
SELECT XMLPI(NAME "hej", 'alla glada')  
FROM Person
```

```
<?hej alla glada?>  
<?hej alla glada?>  
<?hej alla glada?>  
<?hej alla glada?>  
<?hej alla glada?>  
<?hej alla glada?>  
<?hej alla glada?>  
<?hej alla glada?>
```

XMLPI

- **Dynamiska värden**

```
SELECT XMLPI(NAME "hej", name)  
FROM Person
```

```
<?hej John Higgins?>  
<?hej Stephen Hendry?>  
<?hej Matthew Stevens?>  
<?hej Ronnie O'Sullivan?>  
<?hej Ken Doherty?>  
<?hej Steve Davis?>  
<?hej Paul Hunter?>  
<?hej Neil Robertson?>
```

31

XMLNAMESPACES

- **Skapar namespaces inuti ett element**
 - Anges i XMLELEMENT före XMLATTRIBUTES

```
SELECT XMLELEMENT(NAME "iv1023:Data",  
                  XMLNAMESPACES('http://ns.kth.se/iv1023' AS "iv1023"),  
                  XMLFOREST(COUNT(*) AS "iv1023:Personer",  
                            (SELECT COUNT(*) FROM Car) AS "iv1023:Bilar"))  
FROM Person
```

```
<iv1023:Data xmlns:iv1023="http://ns.kth.se/iv1023">  
  <iv1023:Personer>8</iv1023:Personer>  
  <iv1023:Bilar>8</iv1023:Bilar>  
</iv1023:Data>
```

32

XMLNAMESPACES DEFAULT

```
SELECT XMLELEMENT(NAME "Data",  
    XMLNAMESPACES(DEFAULT 'http://ns.kth.se/iv1023'),  
    XMLFOREST(COUNT(*) AS "Personer",  
        (SELECT COUNT(*) FROM Car) AS "Bilar"))  
FROM Person
```

```
<Data xmlns="http://ns.kth.se/iv1023">  
  <Personer>8</Personer>  
  <Bilar>8</Bilar>  
</Data>
```

XMLPARSE

- Omvandlar ett serialiserat XML-värde till XML
 - DOCUMENT eller CONTENT

```
XMLPARSE(DOCUMENT '<rot a="1" />')
```

Obs! XML

```
'<rot a="1" />'
```

Obs! STRING

XMLSERIALIZE

- **Serialiserar ett XML-värde enligt en angiven datatyp**
 - DOCUMENT eller CONTENT (default)
 - Kan generera XML-deklarationen

XMLSERIALIZE(xmlvärde AS CLOB)

**XMLSERIALIZE(xmlvärde AS CLOB
INCLUDING XMLDECLARATION)**

XMLCAST

- **Omvandlar värden från/till XML**

XMLCAST(värde AS datatyp)

XMLVALIDATE

- **Validerar ett XML-värde enligt ett schema**
 - Schemat kan anges i funktionsanropet
 - Schemat kan anges inuti XML-värdet
- **Returnerar det validerade XML-värdet**
 - Eller ett fel

XMLVALIDATE(xmlvärde ACCORDING TO xmlschema)

XMLVALIDATE(xmlvärde)

XMLQUERY

- **Exekverar XQuery-uttryck**
 - Returnerar XML

```
SELECT XMLQUERY('for $a in (1, 2, 3) return element Nummer {$a}')  
FROM Person  
WHERE pid = 1
```

```
<Nummer>1</Nummer><Nummer>2</Nummer><Nummer>3</Nummer>
```

XMLQUERY

– Kan skicka kolumnvärden som variabler

```
SELECT name, XMLQUERY('for $a in distinct-values($e//@employer)
                        return element Arbetsgivare {$a}'
                        PASSING employments AS "e")
FROM Person
```

John Higgins	<Arbetsgivare>ABB</Arbetsgivare><Arbetsgivare>UPC</Arbetsgivare>
Stephen Hendry	<Arbetsgivare>ABB</Arbetsgivare><Arbetsgivare>UPC</Arbetsgivare>
Matthew Stevens	<Arbetsgivare>UPC</Arbetsgivare>
Ronnie O'Sullivan	<Arbetsgivare>LKP</Arbetsgivare><Arbetsgivare>STG</Arbetsgivare>
Ken Doherty	<Arbetsgivare>LKP</Arbetsgivare><Arbetsgivare>ABB</Arbetsgivare> <Arbetsgivare>FFD</Arbetsgivare><Arbetsgivare>STG</Arbetsgivare>
Steve Davis	<Arbetsgivare>ABB</Arbetsgivare><Arbetsgivare>LKP</Arbetsgivare> <Arbetsgivare>FFD</Arbetsgivare>
Paul Hunter	<Arbetsgivare>FFD</Arbetsgivare><Arbetsgivare>LKP</Arbetsgivare>
Neil Robertson	<Arbetsgivare>UPC</Arbetsgivare><Arbetsgivare>ABB</Arbetsgivare>

39

XMLQUERY

```
SELECT name, XMLQUERY('count(distinct-values($e//@employer))'
                        PASSING employments AS "e") AS AntalArbetsgivare
FROM Person
```

John Higgins	2
Stephen Hendry	2
Matthew Stevens	1
Ronnie O'Sullivan	2
Ken Doherty	4
Steve Davis	3
Paul Hunter	2
Neil Robertson	2

40

XMLTABLE

- Skapar en tabell ur ett XQuery-resultat

```
SELECT *
```

```
FROM XMLTABLE('for $a in (2, 5, 9)  
               return element Resultat {$a}')
```

```
<Resultat>2</Resultat>
```

```
<Resultat>5</Resultat>
```

```
<Resultat>9</Resultat>
```

41

XMLTABLE implicit join

```
SELECT name, t2.*  
FROM Person, XMLTABLE('for $a in distinct-values($e//@employer)  
                       return element Arbetsgivare {$a}'  
                       PASSING employments AS "e" ) AS t2
```

John Higgins	<Arbetsgivare>ABB</Arbetsgivare>
John Higgins	<Arbetsgivare>UPC</Arbetsgivare>
Stephen Hendry	<Arbetsgivare>ABB</Arbetsgivare>
Stephen Hendry	<Arbetsgivare>UPC</Arbetsgivare>
Matthew Stevens	<Arbetsgivare>UPC</Arbetsgivare>
Ronnie O'Sullivan	<Arbetsgivare>LKP</Arbetsgivare>
Ronnie O'Sullivan	<Arbetsgivare>STG</Arbetsgivare>
Ken Doherty	<Arbetsgivare>LKP</Arbetsgivare>
Ken Doherty	<Arbetsgivare>ABB</Arbetsgivare>
Ken Doherty	<Arbetsgivare>FFD</Arbetsgivare>
Ken Doherty	<Arbetsgivare>STG</Arbetsgivare>
Steve Davis	<Arbetsgivare>ABB</Arbetsgivare>
Steve Davis	<Arbetsgivare>LKP</Arbetsgivare>
Steve Davis	<Arbetsgivare>FFD</Arbetsgivare>
Paul Hunter	<Arbetsgivare>FFD</Arbetsgivare>
Paul Hunter	<Arbetsgivare>LKP</Arbetsgivare>
Neil Robertson	<Arbetsgivare>UPC</Arbetsgivare>
Neil Robertson	<Arbetsgivare>ABB</Arbetsgivare>

42

XMLTABLE kolumnner

SELECT name, t2.*

FROM Person, XMLTABLE('\$e//employment'

PASSING employments AS "e"

COLUMNS

arbetsgivare VARCHAR(10) PATH '@employer',

startdate DATE PATH '@startdate',

enddate DATE PATH '@enddate') AS t2

NAME	ARBETSGIVARE	STARTDATE	ENDDATE
John Higgins	ABB	08/20/2001	02/28/2009
John Higgins	UPC	04/15/2009	-
Stephen Hendry	ABB	08/20/2002	06/30/2003
Stephen Hendry	UPC	08/01/2003	-
Stephen Hendry	ABB	11/01/2006	-
Matthew Stevens	UPC	01/10/2003	-
Ronnie O'Sullivan	LKP	03/10/2002	05/22/2010
Ronnie O'Sullivan	STG	08/15/2010	-
Ken Doherty	LKP	02/12/2002	05/11/2003
Ken Doherty	ABB	05/12/2003	12/02/2003
Ken Doherty	LKP	12/06/2003	02/17/2005
Ken Doherty	FFD	02/18/2005	05/16/2008
Ken Doherty	STG	06/02/2008	-
Steve Davis	ABB	01/05/2001	12/31/2005
Steve Davis	LKP	01/15/2006	01/22/2009
Steve Davis	FFD	02/01/2009	-
Paul Hunter	FFD	01/10/2004	09/29/2008
Paul Hunter	LKP	10/01/2008	11/20/2010
Neil Robertson	UPC	03/02/2006	10/30/2008
Neil Robertson	ABB	11/20/2008	-

43

XMLTABLE

SELECT name AS Namn, COUNT(*) AS AntalJobbJustNu

FROM Person, XMLTABLE('\$e//employment[not(@enddate)]'

PASSING employments AS "e") AS t2

GROUP BY name

Namn	AntalJobbJustNu
John Higgins	1
Ken Doherty	1
Matthew Stevens	1
Ronnie O'Sullivan	1
Stephen Hendry	2
Steve Davis	1
Neil Robertson	1

SELECT t2.*

FROM Person, XMLTABLE('\$e//employment[not(@enddate)]'

PASSING employments AS "e") AS t2

WHERE name = 'Stephen Hendry'

<employment startdate="2003-08-01" employer="UPC"/>

<employment startdate="2006-11-01" employer="ABB"/>

44

XMLTABLE

```
SELECT name, startdate
FROM Person, XMLTABLE('$e//employment[not(@enddate)]/@employer'
                      PASSING employments AS "e"
                      COLUMNS arbetsgivare VARCHAR(10) PATH '.',
                               startdate VARCHAR(10) PATH '../@startdate') AS t2
WHERE arbetsgivare = 'UPC'
```

NAME	STARTDATE
John Higgins	2009-04-15
Stephen Hendry	2003-08-01
Matthew Stevens	2003-01-10

XMLTABLE, OUTER JOIN

```
SELECT name, startdate
FROM Person LEFT OUTER JOIN XMLTABLE('$e//employment[@employer = "ABB"]'
                                     PASSING employments AS "e"
                                     COLUMNS startdate DATE PATH '@startdate',
                                              enddate VARCHAR(10) PATH '@enddate') AS t2
      ON 1=1
```

NAME	STARTDATE	ENDDATE
John Higgins	2001-08-20	2009-02-28
Stephen Hendry	2002-08-20	2003-06-30
Stephen Hendry	2006-11-01	NULL
Matthew Stevens	NULL	NULL
Ronnie O'Sullivan	NULL	NULL
Ken Doherty	2003-05-12	2003-12-02
Steve Davis	2001-01-05	2005-12-31
Paul Hunter	NULL	NULL
Neil Robertson	2008-11-20	NULL

XMLTABLE FOR ORDINALITY

- **FOR ORDINALITY**

- Speciell kolumn som numrerar raderna i resultatet av varje XMLTABLE-anrop

```
SELECT name, ordnr, arbetsgivare
FROM Person, XMLTABLE('distinct-values($e//employment[@enddate]/@employer)'
                      PASSING employments AS "e"
                      COLUMNS ordnr FOR ORDINALITY,
                      arbetsgivare VARCHAR(5) PATH 'string()') AS t2
```

John Higgins	1	ABB
Stephen Hendry	1	ABB
Ronnie O'Sullivan	1	LKP
Ken Doherty	1	LKP
Ken Doherty	2	ABB
Ken Doherty	3	FFD
Steve Davis	1	ABB
Steve Davis	2	LKP
Paul Hunter	1	FFD
Paul Hunter	2	LKP
Neil Robertson	1	UPC

47

XMLEXISTS

- Kollar om resultatet av ett XQuery-uttryck är tomt

```
SELECT name
FROM Person
WHERE
XMLEXISTS('$e//employment[@employer="ABB"]'
          PASSING employments AS "e")
```

John Higgins
Stephen Hendry
Ken Doherty
Steve Davis
Neil Robertson

48

Stöd i DBMS

- **Fullt stöd enligt SQL 2003 och delvis enligt SQL 2006 och SQL 2008**
 - IBM DB2 10, 11 (inte riktigt fullt stöd för XPath)
 - Oracle 11g R2, 12c, 18c, 19c (vissa begränsningar)
 - PostgreSQL 11-14 (begränsat, endast XPath, inte XQuery)
 - OpenLink Virtuoso (begränsat stöd)
 - SAP (tidigare Sybase) SQL Anywhere 17 (begränsat stöd)
- **Alternativa sätt**
 - Microsoft

Fortsättning

- **Quiz om SQL/XML**
- **Lektionsuppgifter**
- **Seminarieuppgifter (Inlupp 1)**