

Sprawozdanie - Zaawansowane technologie bazodanowe

Laboratorium 7 - 17.01.2020

Bazy danych i XML

Dominik Wróbel

Zadanie 1 – Tworzenie tabeli i wypełnianie danymi

Utworzenie tabeli:

```
1. CREATE TABLE kompozycje (  
2.     idkompozycji CHAR(5) NOT NULL,  
3.     nazwa VARCHAR(40) NOT NULL,  
4.     opis VARCHAR(100),  
5.     cena NUMERIC(7, 2),  
6.     minimum INTEGER,  
7.     stan INTEGER  
8. );
```

Wypełnienie tabeli danymi:

```
1. \i kwiaciarnia-kompozycje.sql
```

Zadanie 2 – Tworzenie dokumentu XML na podstawie danych z tabeli kompozycje

Docelowy format:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<flowers>
```

```
    <bouquet id="ko2" quant="12" price="120">
```

```
        <name>Kosz rozyczek</name>
```

```
        <description>tuzin czerwonych rozyczek, molucella, gips, sizal,  
        koszyk czerwony z palakiem
```

```
    </description> </bouquet>
```

```
    ...
```

```
</flowers>
```

Zapytanie (funkcja xmlelement tworzy nowy element xml, a xmlattributes dodaje atrybuty do elementu):

```
1. SELECT xmlelement (
2.     NAME flowers,
3.     xmlelement(NAME bouquet, xmlattributes (k.idkompozycji AS id, k.stan AS
        quant, k.cena AS price),
4.         xmlelement(NAME NAME, k.nazwa),
5.         xmlelement(NAME description, k.opis)
6.     )
7. ) FROM kompozycje k
8. WHERE stan > 4;
```

Wynik zapytania:

```
<flowers>
<bouquet id="ko2 " quant="12" price="120.00">
<name>Kosz rozyczek</name>
<description>tuzin czerwonych rozyczek, molucella, gips, sizal, koszyk
czerwony z palakiem
</description>
</bouquet>
</flowers>

<flowers><bouquet id="ko3 " quant="5" price="250.00"><name>Kosz
mix</name><description>gladiole, gerbery, sloneczniki mini, leuki, kolorowe
liscie, kosz z palakiem</description></bouquet></flowers>
<flowers><bouquet id="don2 " quant="5" price="120.00"><name>Rozowa
azalia</name><description>rozowa azalia z dekoracja w
koszyku</description></bouquet></flowers>
<flowers><bouquet id="buk4 " quant="9" price="90.00"><name>Wiazanka
czerwona</name><description>11 roz Amor i
przybranie</description></bouquet></flowers>
<flowers><bouquet id="buk5 " quant="16" price="75.00"><name>Bukiecik
serc</name><description>3 czerwone rozyczki, serduszka, zielen, podklad
sizalowy</description></bouquet></flowers>
<flowers><bouquet id="kw2 " quant="7" price="50.00"><name>Ikebana z
rozami</name><description>czerwone rozyczki midi, kwiaty sezonowe,
ikebana</description></bouquet></flowers>
<flowers><bouquet id="kw4 " quant="5"
price="240.00"><name>Strelacje</name><description>5 strelacji, mandzuria,
wazon</description></bouquet></flowers>
<flowers><bouquet id="kw5 " quant="5" price="145.00"><name>Anturium
latem</name><description>3 anturia, kapusta, knofia, roza herbaciana,
margerytka, pteris</description></bouquet></flowers>
```

Zadanie 3 – Utworzenie tabeli printers z informacjami w formie xml

Utworzenie tabeli:

```
1. CREATE TABLE printers (  
2.     id SERIAL PRIMARY KEY,  
3.     NAME VARCHAR(50) NOT NULL,  
4.     description XML  
5. );
```

Dodanie informacji do tabeli:

```
1. #!/bin/bash  
2. database="wrobdom1"  
3.  
4. for i in *.xml  
5. do  
6. cat << EOT  
7. INSERT INTO printers (name, description) VALUES('${i%.*}', '`cat $i`');  
8. EOT  
9. # echo "${i%.*}"  
10. done
```

```
1. ./script >> commands.sql
```

Następnie sprawdzono poprawność zapisu danych:

```
SELECT * FROM printers;
```

Zadanie 4 – Tworzenie zapytań korzystając z xpath

```
SELECT NAME, Xpath('/printer/mechanism/resolution/dpi/x/text()', description) FROM printers;
```

name	xpath
-----+-----	
Brother-4550	{600}
Brother-DCP-1200	{600}
Brother-DCP-7025	{1200}
Brother-DCP-8020	{600}
Brother-DCP-8025D	{600}
Brother-DCP-8040	{600}
Brother-HL-4070CDW	{2400}
Brother-DCP-8045D	{2400}
Brother-HJ-400	{360}
Brother-HL-1020	{600}
Brother-HL-1030	{600}
Brother-HL-1040	{600}
Brother-HL-1650_70N	{1200}
Brother-HL-1050	{1200}
Brother-HL-1060	{1200}

Brother-HL-1070	{1200}
-----------------	--------

Brother-HL-10h	{600}
----------------	-------

Brother-HL-10V	{300}
----------------	-------

Brother-HL-1230	{600}
-----------------	-------

Brother-HL-1850_70N	{1200}
---------------------	--------

Brother-HL-2060	{1200}
-----------------	--------

Brother-HL-2170W	{2400}
------------------	--------

Brother-HL-2400CeN	{2400}
--------------------	--------

Brother-HL-3400CN	{2400}
-------------------	--------

Brother-HL-4040CN	{2400}
-------------------	--------

Brother-HL-4050CDN	{2400}
--------------------	--------

Brother-HL-4Ve	{300}
----------------	-------

Brother-HL-5030	{2400}
-----------------	--------

Brother-HL-5040	{2400}
-----------------	--------

Brother-HL-5140	{2400}
-----------------	--------

...

```
SELECT NAME, Coalesce(Nullif(Btrim(Xpath('/printer/mechanism/resolution/dpi/x/text()'),description) :: text, '{}'), ''), '0') AS result FROM printers ORDER BY result ;
```

name	result
Canon-LBP-5960	0
Canon-LBP-5360	0
Canon-LBP-5975	0
Canon-LBP-8A1	0
Canon-MultiPASS_C2500	0
Canon-LBP-3460	0
Canon-BJ-300	0
Canon-BJC-70	0
Canon-MultiPASS_C3000	0
Canon-LBP-4plus	0
Canon-LBP-600	0
Canon-LBP-5970	0
Brother-MP-21C	0
Canon-iP4000	0
Canon-GP_405	1200
Canon-BJC-7000	1200
Canon-BJC-7004	1200
Canon-LBP-470	1200
Brother-DCP-7025	1200
Canon-LIPS-IV	1200
Canon-LIPS-IVv	1200
Brother-HL-5250DN	1200
Canon-LBP-1760	1200
Brother-HL-6050D_DN	1200
Brother-HL-6050	1200
Canon-LBP-1000	1200
Brother-HL-2060	1200
Brother-HL-760	1200
Brother-HL-8050N	1200
Canon-GP_335	1200
Brother-MFC-7820N	1200
...	

```
SELECT NAME, Coalesce(Nullif(Btrim(Xpath('/printer/mechanism/resolution/dpi/x/text
()'),description) :: text, '{}'), ''), '0') :: INT AS result FROM printers ORDER BY
result;
```

name	result
Canon-iP4000	0
Canon-LBP-600	0
Brother-MP-21C	0
Canon-BJ-300	0
Canon-LBP-3460	0
Canon-MultiPASS_C3000	0
Canon-MultiPASS_C2500	0
Canon-LBP-4plus	0
Canon-LBP-5360	0
Canon-LBP-5960	0
Canon-BJC-70	0
Canon-LBP-5970	0
Canon-LBP-8A1	0
Canon-LBP-5975	0
Brother-PT-1950	180
Brother-PT-1950VP	180
Brother-PT-18R	180
Brother-PT-1500PC	180
Brother-PT-PC	180
Brother-PT-550A	180
Brother-PT-2610	180
Brother-PT-2600	180
Brother-PT-2450DX	180
Brother-PT-2500PC	180
Brother-PT-2420PC	180
Brother-PT-1960	180
Canon-LIPS-IIplus	240
Canon-CP-220	300
Brother-QL-500	300
Brother-QL-550	300

```
SELECT name FROM printers WHERE Coalesce(Nullif(Btrim(Xpath('/printer/mechanism/resolution/dpi/x/text()', description) :: text, '{}'), ''), '0') :: INT > 1200 AND Xpath_exists('/printer/mechanism/color', description);
```

name

Brother-HL-4070CDW
Brother-HL-2400CeN
Brother-HL-3400CN
Brother-HL-4040CN
Brother-HL-4050CDN
Brother-MFC_7150C
Brother-MFC-9100c
Canon-BJC-3000
Canon-BJC-6500
Canon-BJC-5100
Canon-BJC-6000
Canon-BJC-6100
Canon-BJC-6200
Canon-S200
Canon-S300
Canon-S400
Canon-S4500
Canon-S450
Canon-S500
Canon-S600
Canon-S630
Canon-S800
(22 rows)

Zadanie 5 – Indeksowanie

Zapytania przed utworzeniem indeksu:

```
SELECT NAME, Xpath('/printer/mechanism/resolution/dpi/x/text()', description) FROM printers;
```

QUERY PLAN

Seq Scan on printers (cost=0.00..31.43 rows=194 width=47)

(1 row)


```
SELECT NAME, Coalesce(Nullif(Btrim(Xpath('/printer/mechanism/resolution/dpi/x/text()'),description) :: text, '{}'), ''), '0') AS result FROM printers ORDER BY result;
```

QUERY PLAN

Sort (cost=40.74..41.22 rows=194 width=47)

Sort Key: (COALESCE(NULLIF(btrim((xpath('/printer/mechanism/resolution/dpi/x/text()')::text, description, '{}':text[]))::text, '{}':text), ''::text), '0':text))

-> Seq Scan on printers (cost=0.00..33.37 rows=194 width=47)

(3 rows)

```
SELECT NAME, Coalesce(Nullif(Btrim(Xpath('/printer/mechanism/resolution/dpi/x/text()'),description) :: text, '{}'), ''), '0') :: INT AS result FROM printers ORDER BY result;
```

QUERY PLAN

Sort (cost=41.71..42.19 rows=194 width=19)

Sort Key: ((COALESCE(NULLIF(btrim((xpath('/printer/mechanism/resolution/dpi/x/text()')::text, description, '{}':text[]))::text, '{}':text), ''::text), '0':text))::integer)

-> Seq Scan on printers (cost=0.00..34.34 rows=194 width=19)

(3 rows)

```
SELECT name FROM printers WHERE Coalesce(Nullif(Btrim(Xpath('/printer/mechanism/resolution/dpi/x/text()', description) :: text, '{}'), ''), '0') :: INT > 1200 AND Xpath_exists('/printer/mechanism/color', description);
```

QUERY PLAN

Seq Scan on printers (cost=0.00..35.30 rows=22 width=15)

Filter: (xpath_exists('/printer/mechanism/color'::text, description, '{}':text[]) AND ((COALESCE(NULLIF(btrim((xpath('/printer/mechanism/resolution/dpi/x/text()')::text, description, '{}':text[]))::text, '{}':text), ''::text), '0':text))::integer > 1200))

Utworzenie indeksu:

1. wrobdom1=> **CREATE INDEX** xml_ind **ON** printers USING btree (((xpath('/printer/mechanism/resolution/dpi/x/text()', description))[1])::text));
2. **CREATE INDEX**
3. wrobdom1=> **SET** ENABLE_SEQSCAN **TO OFF**;
4. **SET**

Zapytania po utworzeniu indeksu:

```
SELECT NAME, Xpath('/printer/mechanism/resolution/dpi/x/text()', description) FROM printers;
```

QUERY PLAN

Seq Scan on printers (cost=10000000000.00..10000000031.43 rows=194 width=47)

(1 row)

```
SELECT NAME, Coalesce(Nullif(Btrim(xpath('/printer/mechanism/resolution/dpi/x/text()',description) :: text, '{}'), ''), '0') AS result FROM printers ORDER BY result;
```

QUERY PLAN

Sort (cost=10000000040.74..10000000041.22 rows=194 width=47)

Sort Key: (COALESCE(NULLIF(btrim((xpath('/printer/mechanism/resolution/dpi/x/text()':text, description, '{}':text[]))::text, '{}':text), ''::text), '0':text))

-> Seq Scan on printers (cost=10000000000.00..10000000033.36 rows=194 width=47)

(3 rows)

```
SELECT NAME, Coalesce(Nullif(Btrim(Xpath('/printer/mechanism/resolution/dpi/x/text()'),description) :: text, '{}'), ''), '0') :: INT AS result FROM printers ORDER BY result;
```

QUERY PLAN

Sort (cost=100000000041.71..100000000042.19 rows=194 width=19)

Sort Key: ((COALESCE(NULLIF(btrim((xpath('/printer/mechanism/resolution/dpi/x/text()')::text, description, '{}')::text[]))::text, '{}')::text), ''::text), '0'::text)::integer)

-> Seq Scan on printers (cost=10000000000.00..100000000034.34 rows=194 width=19)

(3 rows)

```
SELECT name FROM printers WHERE Coalesce(Nullif(Btrim(Xpath('/printer/mechanism/resolution/dpi/x/text()'), description) :: text, '{}'), ''), '0') :: INT > 1200 AND Xpath_exists('/printer/mechanism/color', description);
```

QUERY PLAN

Seq Scan on printers (cost=10000000000.00..100000000035.31 rows=22 width=15)

Filter: (xpath_exists('/printer/mechanism/color'::text, description, '{}')::text[] AND ((COALESCE(NULLIF(btrim((xpath('/printer/mechanism/resolution/dpi/x/text()')::text, description, '{}')::text[]))::text, '{}')::text), ''::text), '0'::text)::integer > 1200))

(2 rows)