

Model badania wydajności zapytań wykorzystujących dostęp do dokumentów XML

Promotor:
dr inż. Paweł FIGAT



Dziedzina problemu

- W bazach danych do poprawy wydajności zapytań stosuje się różnego rodzaju indeksy
- Wprowadzenie do baz danych typu danych XML ułatwiło przetwarzanie danych tego typu
- Następnie wprowadzono różne mechanizmy służące do poprawy wydajności przetwarzania dokumentów XML



Cel i teza

- **Cel pracy:** Zaprojektowanie modelu badania wydajności zapytań, oraz zbadanie wydajność mechanizmów mających zwiększać wydajność przetwarzania dokumentów XML w SQL Server 2008.
- **Teza:** Jakie mechanizmy służące do poprawy wydajności przetwarzania dokumentów XML stosować, żeby zyskać poprawę wydajności?



XML i XSD

- XML

- Dokument tekstowy
- Elastyczność
- Uniwersalność

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <Kontakty>
3   <Kontakt>
4     <Rodzaj>Telefon stacjonarny</Rodzaj>
5     <Numer>22 5433454</Numer>
6   </Kontakt>
7   <Kontakt>
8     <Rodzaj>Telefon komórkowy</Rodzaj>
9     <Numer>505 324 324</Numer>
10    <Uwaga>Nie wysyłać smsów</Uwaga>
11  </Kontakt>
12 </Kontakty>
```

- XSD – XML Schema

- Dokument XML
- Nakładanie ograniczeń na:
 - Strukturę
 - Typy danych



Obsługa XML w SQL Server

- Value

```
1 SELECT @dane.value(' (//Kontakt/Numer) [1]', 'varchar(20)') as Numer;
```

- Exists

```
1 SELECT @dane.exist(' (//Kontakt/Uwaga)') as czyUwaga;
```

- Query

```
1 SELECT @dane.query(' (//Kontakt[1])') as pierwszyKontakt;
```

- Nodes

```
1 SELECT
2     nazwisko + ' ' + imie as klient,
3     dane.value(' ./Rodzaj[1]', 'varchar(50)') as rodzajKontaktu,
4     dane.value(' ./Numer[1]', 'varchar(30)') as Numer
5 FROM Klient CROSS APPLY
6     kontakty.nodes(' (//Kontakty/Kontakt)') as tmp(dane);
```

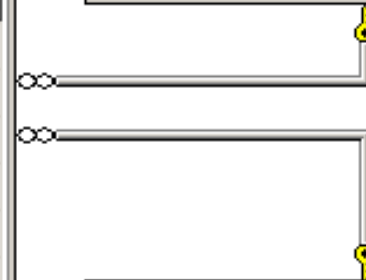


Środowisko testowe – struktura bazy

kontrahenci		
	Column Name	Data Type
🔑	idKontrahenta	int
	idRodzajKontrahenta	int
	idMiasta	int
	nazwa	varchar(150)
	nip	varchar(10)
	faktury	xml

rodzajeKontrahenta		
	Column Name	Data Type
🔑	idRodzajKontrahenta	int
	rodzajKontrahenta	varchar(50)

miasta		
	Column Name	Data Type
🔑	idMiasta	int
	miasto	varchar(50)



Środowisko testowe – dokument XML

```
1 <Faktury>
2   <Faktura>
3     <Numer>13431/2011</Numer>
4     <DataWystawienia>20110105</DataWystawienia>
5     <TerminPlatnosci>20110215</TerminPlatnosci>
6     <Elementy>
7       <Element>
8         <Nazwa>Papier A4</Nazwa>
9         <Ilosc>12</Ilosc>
10        <Cena>120</Cena>
11        <StawkaVat>22</StawkaVat>
12      </Element>
13      <Element>
14        <Nazwa>Tusz HP - czarny</Nazwa>
15        <Ilosc>2</Ilosc>
16        <Cena>195</Cena>
17        <StawkaVat>23</StawkaVat>
18      </Element>
19      <Element>
20        <Nazwa>Płyty CD - 10szt</Nazwa>
21        <Ilosc>6</Ilosc>
22        <Cena>75</Cena>
23        <StawkaVat>3</StawkaVat>
24      </Element>
25    </Elementy>
26  </Faktura>
27  <Faktura>
28    <Numer>5646/2011</Numer>
29    <DataWystawienia>20110215</DataWystawienia>
30    <TerminPlatnosci>20110322</TerminPlatnosci>
31    <Elementy>
32      <Element>
33        <Nazwa>Klawiatura</Nazwa>
34        <Ilosc>1</Ilosc>
35        <Cena>55</Cena>
36        <StawkaVat>11</StawkaVat>
37      </Element>
38    </Elementy>
39  </Faktura>
40  <Faktura>
41  <Faktura>
42  <Faktura>
43  <Faktura>
44  <Faktura>
45  <Faktura>
46  <Faktura>
47  <Faktura>
48  <Faktura>
49  <Faktura>
50  <Faktura>
51  <Faktura>
52  <Faktura>
53  <Faktura>
54  <Faktura>
55  <Faktura>
56  <Faktura>
57  <Faktura>
58  <Faktura>
59  <Faktura>
60  <Faktura>
61  <Faktura>
62  <Faktura>
63  <Faktura>
64  <Faktura>
65  <Faktura>
66  <Faktura>
67  <Faktura>
68  <Faktura>
69  <Faktura>
70  <Faktura>
71  <Faktura>
72  <Faktura>
73  <Faktura>
74  <Faktura>
75  <Faktura>
76  <Faktura>
77  <Faktura>
78  <Faktura>
79 </Faktury>
```

Generowanie danych

- Źródła danych
 - 903 miasta – wykaz miast Polski wikipedia.org
 - 939 różnych elementów faktur – asortyment różnych sklepów internetowych
- Tabela *kontrahenci*
 - 10 000 wierszy
 - 1-30 faktur
 - 1-20 elementów



Model badania wydajności

- Testy czterech zapytań ukierunkowanych na różne aspekty wydajności
- Dla każdego 3 testy po 12 wykonań każdy
 - Odrzucenie dwóch skrajnych wyników z każdego testu
- 12 różnych warunków testowych dla każdego zapytania
 - „czyste zapytanie”
 - Cztery rodzaje indeksów XML oraz wszystkie razem
 - Schemat XSD
 - Schemat XSD oraz testy wszystkich rodzajów indeksów



Wyniki testów



Zapytanie pierwsze

- Podsumowanie ze sprzedaży poszczególnych produktów dla kontrahentów konkretnego typu

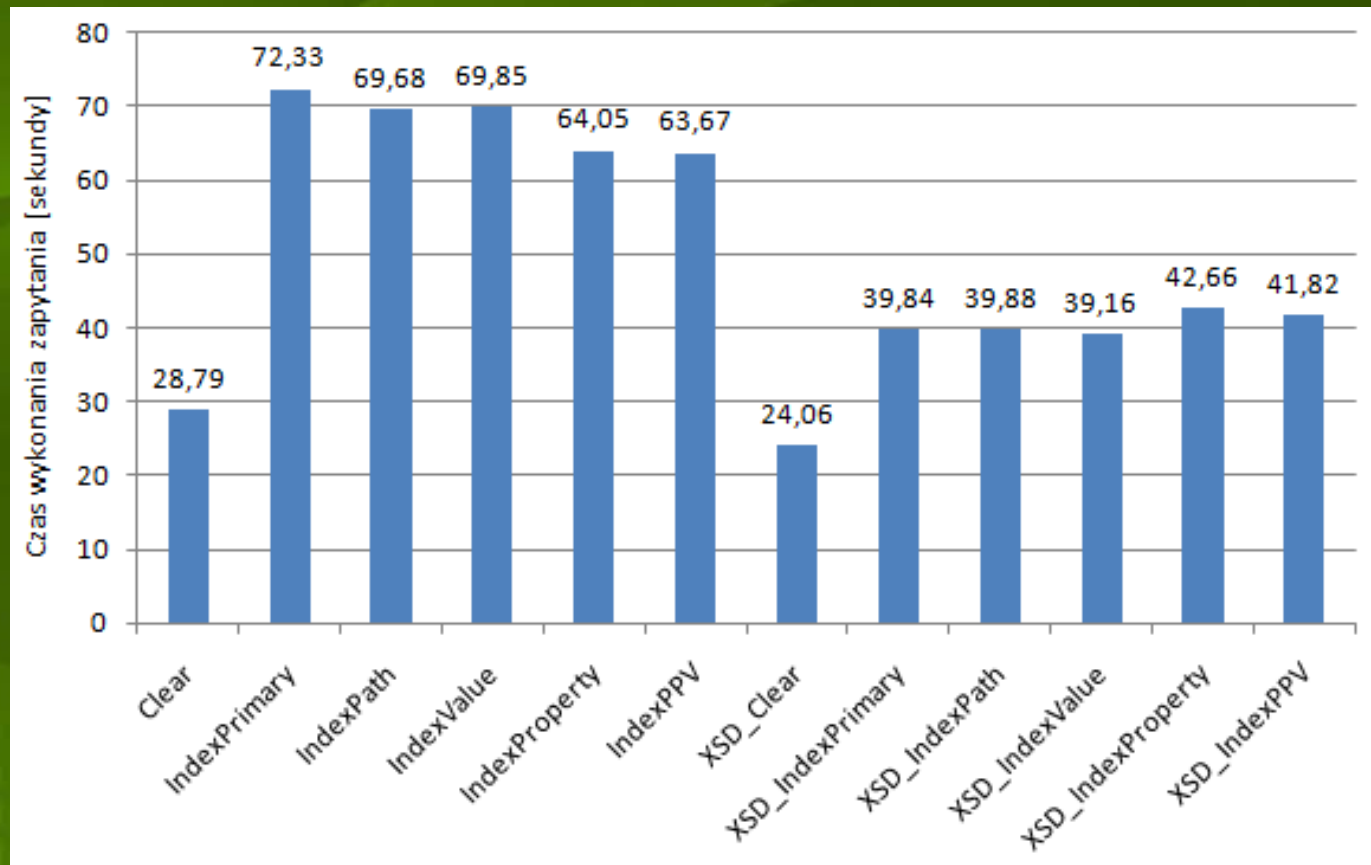
```
1 select tmp.Nazwa,SUM(wartosc) Wartosc,COUNT(*) Ilosc
2 from kontrahenci cross apply
3 (
4     select k.value('Nazwa[1]', 'varchar(128)') Nazwa,
5           k.value('Ilosc[1]', 'int') * k.value('Cena[1]', 'money')
6           as wartosc
7     from faktury.nodes('(/Faktury/Faktura/Elementy/Element)') as t(k)
8 ) as tmp
9 where idRodzajKontrahenta=4
10 group by tmp.nazwa;
```

	Nazwa	Wartosc	Ilosc
1	Pióro kulkowe Parker Duofold Morska Szachownica	1582169,00	320
2	Spinacz biurowy owalny 28mm	1530643,00	308
3	Tusz atramentowy AC-15C zastępuje tusz Canon BCI...	1617876,00	335
4	C4813A-Głowica drukująca nr 11. Kolory: błękitna, pu...	1739381,00	336
5	Folia laminacyjna A5 100 mic	1475617,00	302
6	Pióro wieczne Parker Duofold Centennial Oliwkowa S...	1478145,00	299
7	Toner AT-42 X	1641922,00	335
8	Pojemnik na czasopisma Leitz Allura ET6877/6864	1585223,00	302
9	Bloczek samoprzylepny kolorowy 20x50 mm	1729279,00	353
10	Segregator A4 Esselte Panorama 35mm	1531441,00	320



Zapytanie pierwsze

- Trzy testy 4:57min



Zapytanie drugie

- Podsumowanie ze sprzedaży kontrahentów konkretnego typu

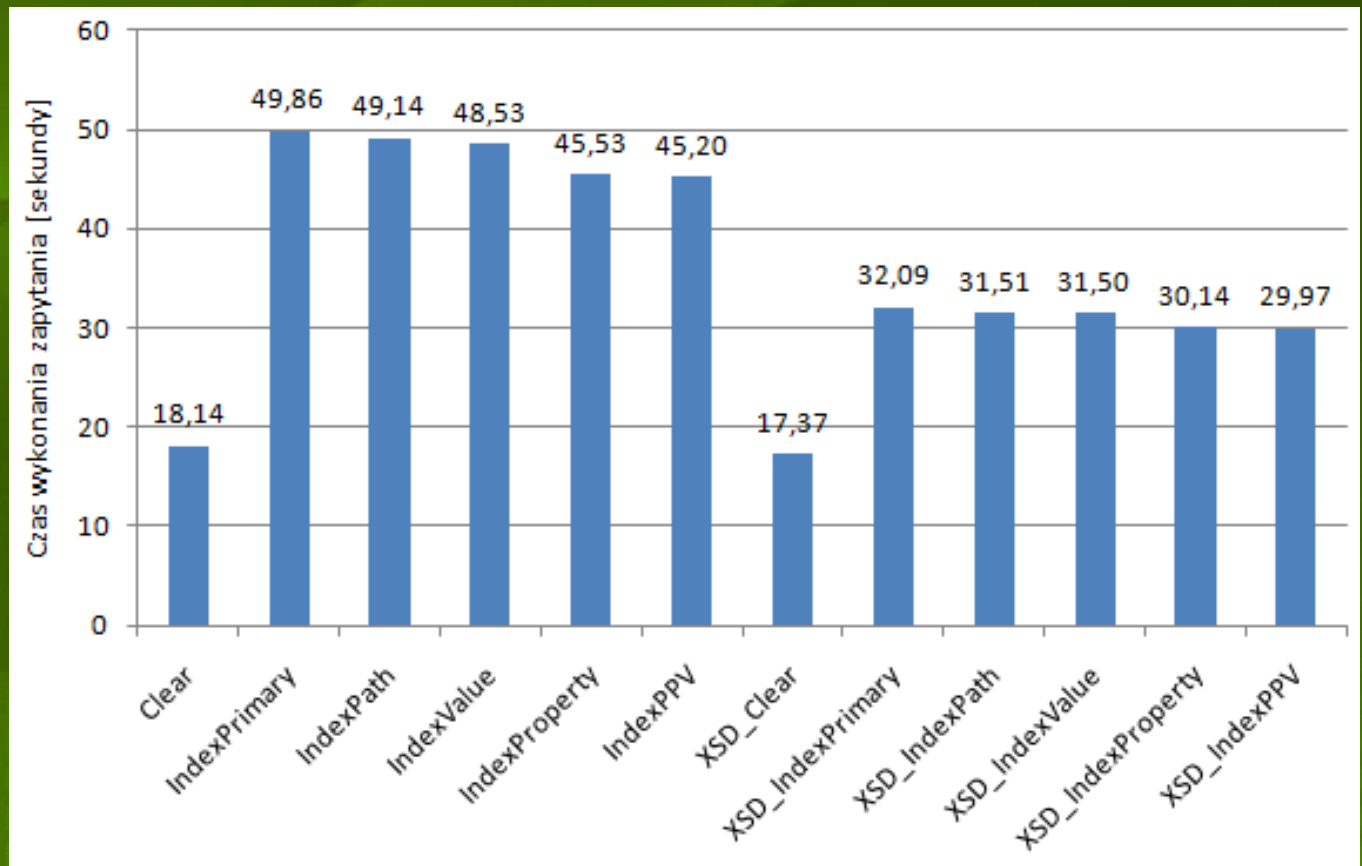
```
1 select Nazwa,Nip,
2   (
3     select SUM(
4       k.value('Ilosc[1]', 'int')*
5       k.value('Cena[1]', 'money')
6     )
7     from faktury.nodes('/Faktury/Faktura/Elementy/Element') t(k)
8   ) as Suma
9 from kontrahenci where idRodzajKontrahenta=4;
10 group by tmp.nazwa;
```

	Nazwa	Nip	Suma
1	Firma 16	5894375251	696161,00
2	Firma 19	3694909821	708495,00
3	Firma 21	1096465341	273379,00
4	Firma 23	2609435881	175278,00
5	Firma 29	6144790681	849701,00
6	Firma 32	1450923471	547966,00
7	Firma 33	8180208831	623377,00
8	Firma 34	4661961811	240630,00
9	Firma 48	4826257121	886174,00
10	Firma 60	9169762971	813397,00



Zapytanie drugie

- Trzy testy 3:34min



Zapytanie trzecie

- Podsumowanie ze sprzedaży kontrahentów konkretnego typu, tylko dla elementów faktury ze stawką vat ≥ 22

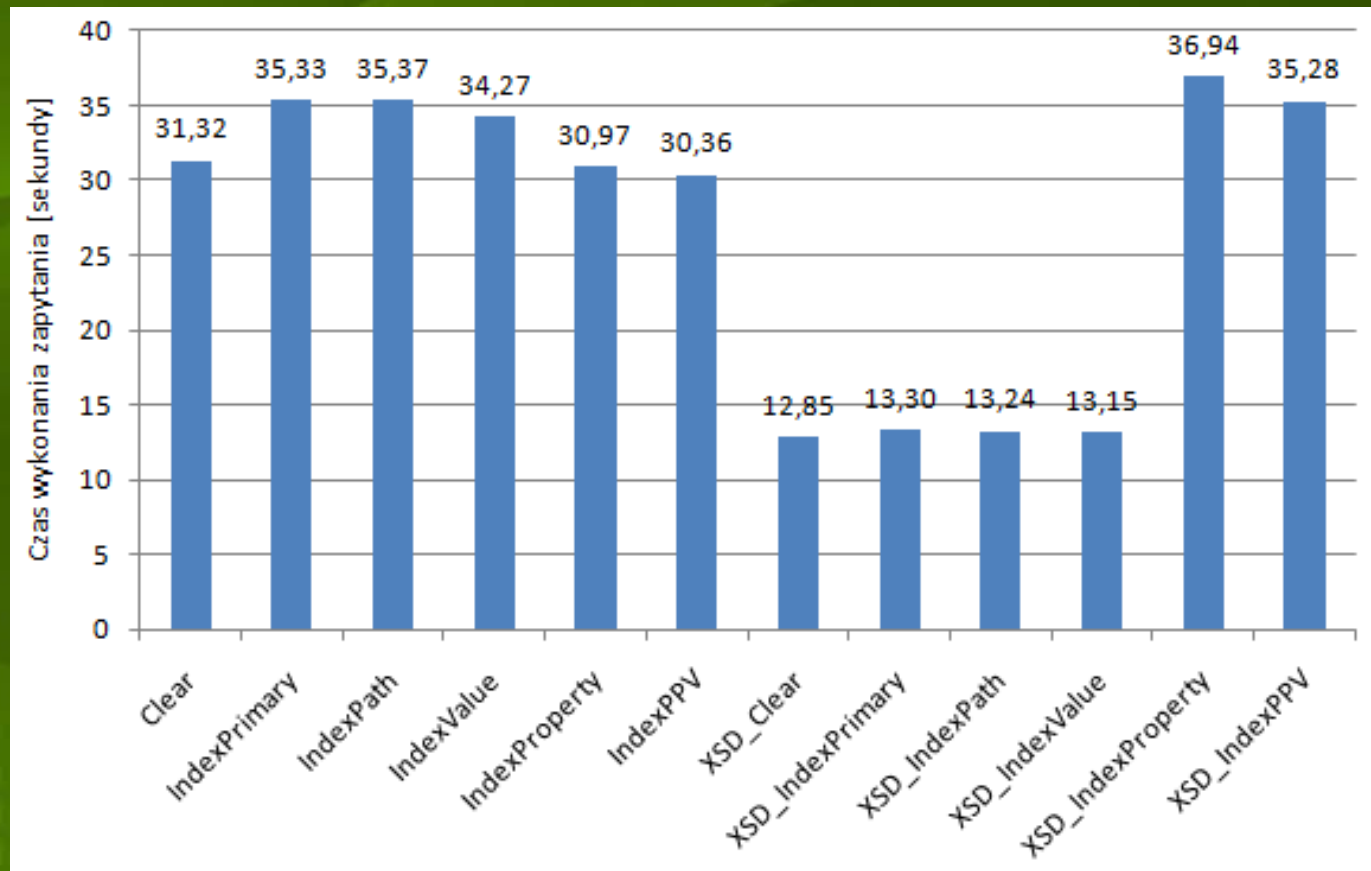
	Nazwa	Nip	Suma
1	Firma 16	5894375251	193113,00
2	Firma 19	3694909821	327552,00
3	Firma 21	1096465341	154255,00
4	Firma 23	2609435881	NULL
5	Firma 29	6144790681	400415,00
6	Firma 32	1450923471	315571,00
7	Firma 33	8180208831	345499,00
8	Firma 34	4661961811	95389,00
9	Firma 48	4826257121	398200,00
10	Firma 60	9169762971	323551,00

```
1 select Nazwa,Nip,
2   (
3     select SUM(
4       k.value('Ilosc[1]', 'int')*
5       k.value('Cena[1]', 'money')
6     )
7   from faktury.nodes('/Faktury/Faktura/Elementy/Element') t(k)
8   where k.exist('StawkaVat[.>=22]')=1
9   ) as Suma
10  from kontrahenci where idRodzajKontrahenta=4;
```



Zapytanie trzecie

- Trzy testy 2:41min



Zapytanie czwarte

- Lista klientów danego typu z fakturą w roku 2010

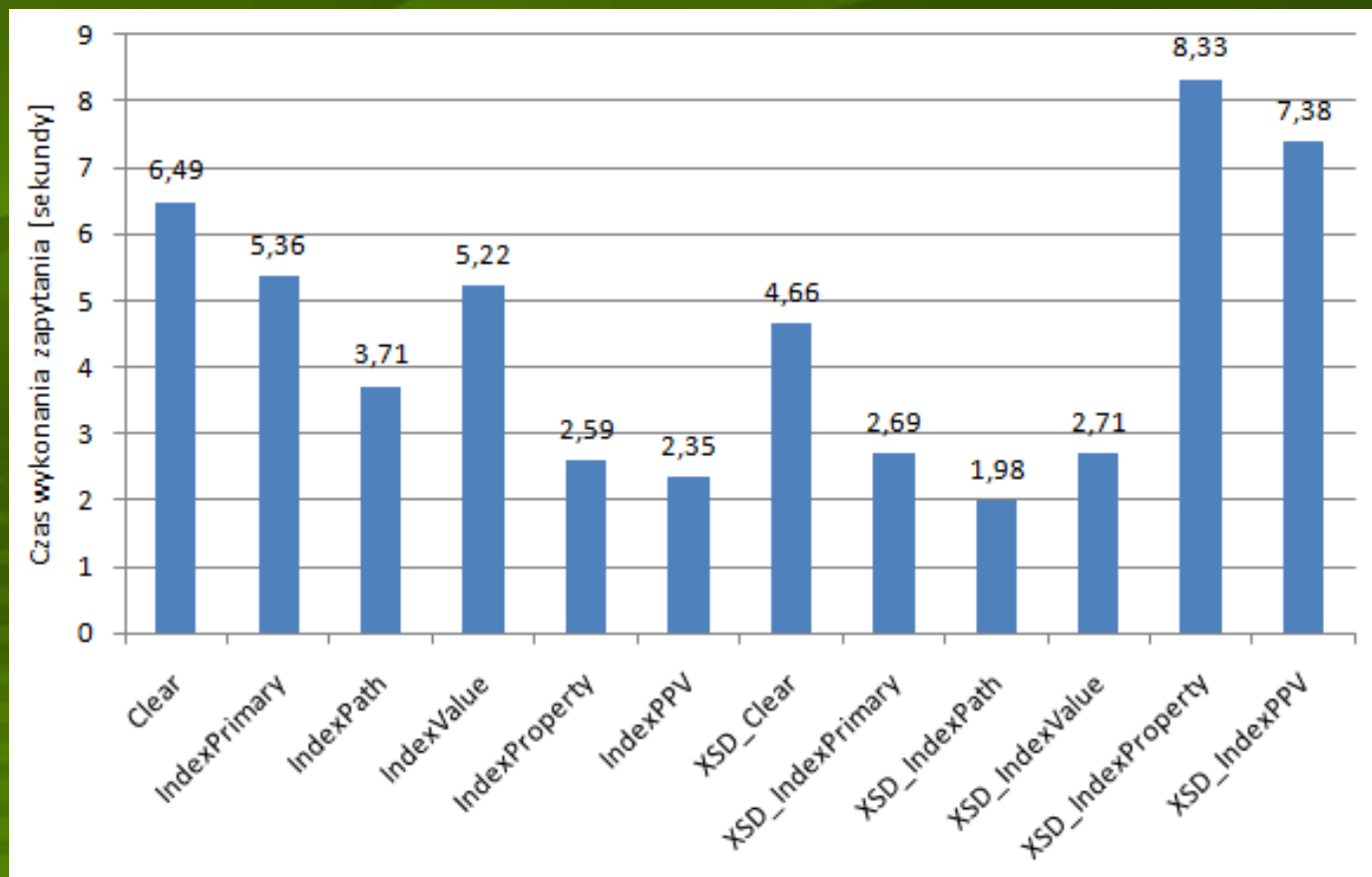
```
1 select distinct Nazwa,Nip
2 from kontrahenci cross apply faktury.nodes('/Faktury/Faktura') t(k)
3 where idRodzajKontrahenta=4
4 and k.exist('DataWystawienia[substring(.,1,4) = "2010"]')=1;
```

	nazwa	nip
1	Firma 2298	1003211901
2	Firma 3521	1011984201
3	Firma 1259	1014122631
4	Firma 3178	1018230091
5	Firma 2495	1019393681
6	Firma 4906	1021097271
7	Firma 1636	1027600701
8	Firma 4133	1044099691
9	Firma 2632	1044819321
10	Firma 6296	1054599751



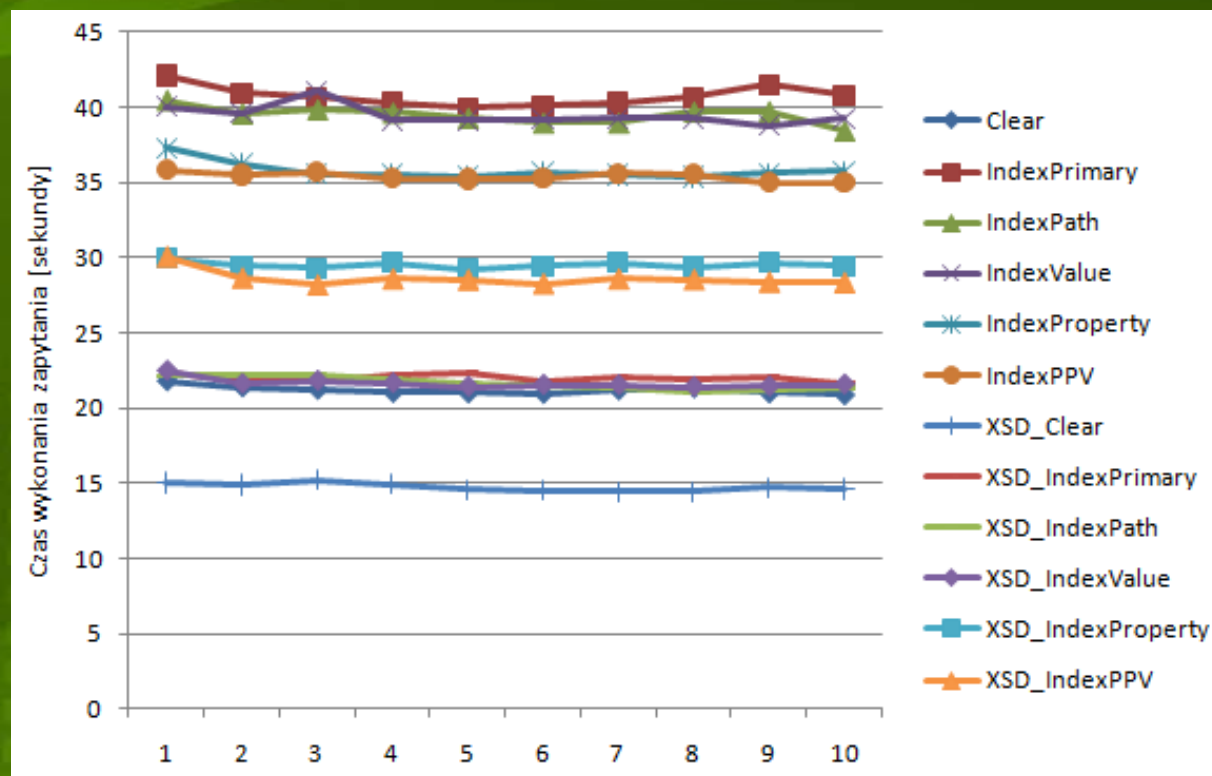
Zapytanie czwarte

- Trzy testy 0:26min



Podsumowanie 1/2

- Testy trwały ok. 27 godzin
- Zmiana warunków dla jednego testu średnio 1:12
- Suma bez zmian warunków testów 11:40



Podsumowanie 2/2

- Z przeprowadzonych testów wynika:
 - Najlepsze wyniki daje stosowanie schematów XSD
 - Stosowanie indeksów w większości przypadków spowodowało znaczne pogorszenie wydajności.
 - W przypadku indeksów XML można zakładać wzrost wydajności dla dokumentów XML o bardziej złożonej strukturze
- Wykonany model testowy jest bardzo elastyczny, można go w przyszłości z powodzeniem stosować do testowania różnych aspektów wydajności



Dziękuję za uwagę

