

Runtime of various XML structures/XPath expressions using the example a car database (Used BASEX GUI for finding runtime)

1)

XPATH Query(car_database)

```
declare namespace ns = "https://www.w3schools.com";  
for $car in /ns:car_database/ns:car[position() <= 10]  
return $car/ns:Engine
```

Total Time: 103.12 ms

XPATH Query(car_database_attr)

```
declare namespace ns = "https://www.w3schools.com";  
for $car in /ns:car_database/ns:car[position() <= 10]  
return $car/@Engine
```

Total Time: 19.5 ms

2)

XPATH Query(car_database)

```
declare namespace ns = "https://www.w3schools.com";  
for $car in /ns:car_database/ns:car[xs:decimal(ns:MSRP_in_USD) > 90.000]  
return $car/ns:Model
```

Total Time: 897.11 ms

XPATH Query(car_database_attr)

```
declare namespace ns = "https://www.w3schools.com";  
for $car in /ns:car_database/ns:car  
let $price := translate($car/@MSRP_in_USD, '$,', '')  
where $price castable as xs:decimal and xs:decimal($price) > 90.000  
return $car/@Model
```

Total Time: 393.19 ms

3)XPATH Query(car_database)

```
declare namespace ns = "https://www.w3schools.com";  
for $car in /ns:car_database/ns:car[  
    xs:decimal(ns:MSRP_in_USD) > 90.000 and xs:integer(ns:Passenger_Capacity) > 7  
]  
return $car/ns:Model
```

Total Time: 770.9 ms

XPATH Query(car_database_attr)

```
declare namespace ns = "https://www.w3schools.com";
```

```
for $car in /ns:car_database/ns:car
let $price := translate($car/@MSRP_in_USD, '$', '')
let $capacity := $car/@Passenger_Capacity
where
    $price castable as xs:decimal
    and xs:decimal($price) < 90000
    and $capacity castable as xs:integer
    and xs:integer($capacity) > 7
return $car/@Model
```

Total Time: 589.39 ms

4)

XPATH Query(car_database)

```
declare namespace ns = "https://www.w3schools.com";
for $car in /ns:car_database/ns:car[
    starts-with(ns:Model, '2018') and
    contains(ns:Model, 'Porsche') and
    xs:integer(ns:Passenger_Capacity) > 2
]
return $car/ns:Model
```

Total Time: 981.52 ms

XPATH Query(car_database_attr)

```
declare namespace ns = "https://www.w3schools.com";
for $car in /ns:car_database/ns:car[
    starts-with(ns:Model, '2018') and
    contains(ns:Model, 'Porsche') and
    xs:integer(ns:Passenger_Capacity) > 2
]
return $car/@Model
```

Total Time: 643.26 ms

Conclusion

The test results clearly indicate that XPath queries on an XML database where values are stored exclusively as attributes, rather than elements, exhibit the lower runtime and thus offer greater efficiency. Typically, this structure not only improves evaluation time but also speeds up printing results. Given these findings, one might conclude that attributes should always be preferred over elements when mapping values, especially if the data does not require nested structures.

However, I would not fully endorse this viewpoint. A significant advantage of XML is its capability to define a structure that is both machine-readable and human-readable. As demonstrated in my second model, structures predominantly composed of attributes can be more challenging to maintain and modify. Hence, while attributes can enhance performance in certain scenarios, they may not always be the optimal choice for ensuring clarity and ease of data manipulation.