

Ejercicio 3

1. Mostrar los nombres de los módulos del ciclo de "Sistemas Microinformáticos y Redes".

- `ies/modulos/modulo/ciclo[.="SMR"]/../../nombre`

```
<?xml version="1.0" encoding="UTF-8"?>
<ies>
  <nombre>IES Marcote</nombre>
  <web>http://www.colegiomarcote.com</web>
  <ciclos>
    <ciclo id="ASIR">
      <nombre>Administración de Sistemas Informáticos en Red</nombre>
      <grado>Superior</grado>
      <decretoTitulo año="2009" />
    </ciclo>
    <ciclo id="DAM">
      <nombre>Desarrollo de Aplicaciones Web</nombre>
      <grado>Superior</grado>
      <decretoTitulo año="2010" />
    </ciclo>
    <ciclo id="SMR">
      <nombre>Sistemas Microinformáticos y Redes</nombre>
      <grado>Medio</grado>
      <decretoTitulo año="2008" />
    </ciclo>
  </ciclos>
  <modulos>
    <modulo id="0228">
      <nombre>Aplicaciones web</nombre>
      <curso>2</curso>
      <horasSemanales>4</horasSemanales>
      <ciclo>SMR</ciclo>
    </modulo>
    <modulo id="0372">
      <nombre>Gestión de bases de datos</nombre>
      <curso>1</curso>
      <horasSemanales>5</horasSemanales>
      <ciclo>ASIR</ciclo>
    </modulo>
    <modulo id="0373">
      <nombre>Lenguajes de marcas y sistemas de gestión de información</nombre>
    </modulo>
  </modulos>
</ies>
```

2. Mostrar los nombres de los ciclos formativos que incluyen el módulo "Lenguajes de marcas y sistemas de gestión de información".

- `ies/modulos/modulo/nombre[.="Lenguajes de marcas y sistemas de gestión de información"]/../../ciclo`

```
<?xml version="1.0" encoding="UTF-8"?>
<ies>
  <nombre>IES Marcote</nombre>
  <web>http://www.colegiomarcote.com</web>
  <ciclos>
    <ciclo id="ASIR">
      <nombre>Administración de Sistemas Informáticos en Red</nombre>
      <grado>Superior</grado>
      <decretoTitulo año="2009" />
    </ciclo>
    <ciclo id="DAM">
      <nombre>Desarrollo de Aplicaciones Web</nombre>
      <grado>Superior</grado>
      <decretoTitulo año="2010" />
    </ciclo>
    <ciclo id="SMR">
      <nombre>Sistemas Microinformáticos y Redes</nombre>
      <grado>Medio</grado>
      <decretoTitulo año="2008" />
    </ciclo>
  </ciclos>
  <modulos>
    <modulo id="0228">
      <nombre>Aplicaciones web</nombre>
      <curso>2</curso>
      <horasSemanales>4</horasSemanales>
      <ciclo>SMR</ciclo>
    </modulo>
    <modulo id="0372">
      <nombre>Gestión de bases de datos</nombre>
      <curso>1</curso>
      <horasSemanales>5</horasSemanales>
      <ciclo>ASIR</ciclo>
    </modulo>
    <modulo id="0373">
      <nombre>Lenguajes de marcas y sistemas de gestión de información</nombre>
      <curso>1</curso>
      <horasSemanales>3</horasSemanales>
      <ciclo>ASIR</ciclo>
    </modulo>
  </modulos>
</ies>
```

-

Ejercicio 5

- Mostrar cada uno de los nombres de los bailes con la etiqueta "losbailes".

for \$baile in /bailes/baile

return <losbailes>{\$baile/nombre/text()}</losbailes>

File [ejemplo] - BaseX 9.7.1
Database Editor View Visualization Options Help

Find: ...

Editor: Context: db:open("ejemplo")

```
1 for $baile in /bailes/baile
2 return <losbailes>{$baile/nombre/text()}</losbailes>
```

Result: 6 Results, 190 b
Total Time: 1.38 ms

Info:

- Compiling:
 - rewrite context value to document-node() item. -> db:open-pre("ejemplo", 0)
 - rewrite uri root(nodes) to document-node() item. util:root(db:open-pre("ejemplo", 0)) -> db:open-pre("ejemplo", 0)
 - inline for \$baile, 0 in db:open-pre("ejemplo", 0)/bailes/baile
 - simplify FLWOR expression: db:open-pre("ejemplo", 0)/bailes/baile 1 <losbailes>{ nombre/text() } </losbailes>
- Optimized Query:
 - db:open-pre("ejemplo", 0)/bailes/baile 1 <losbailes>{ nombre/text() } </losbailes>
- Query:
 - for \$baile in /bailes/baile return <losbailes>{\$baile/nombre/text()}</losbailes>
- Result:
 - Hits: 6 Items
 - Updated: 0 Items
 - Printed: 190 b
 - Read Locking: ejemplo
 - Write Locking: (none)
- Timing:
 - Parsing: 0.28 ms
 - Compiling: 0.52 ms
 - Evaluating: 0.51 ms
 - Printing: 0.06 ms
 - Total Time: 1.38 ms

- Mostrar los nombres de los bailes seguidos con el número de plazas entre paréntesis, ambos dentro de la misma etiqueta "losbailes".

for \$baile in /bailes/baile

return <losbailes>{\$baile/nombre/text()} ({ \$baile/plazas/text() })</losbailes>

File [ejemplo] - BaseX 9.7.1
Database Editor View Visualization Options Help

Find: ...

Editor: Context: db:open("ejemplo")

```
1 for $baile in /bailes/baile
2 return <losbailes>{$baile/nombre/text()} ({ $baile/plazas/text() })</losbailes>
```

Result: 6 Results, 219 b
Total Time: 2.0 ms

Info:

- Compiling:
 - rewrite context value to document-node() item. -> db:open-pre("ejemplo", 0)
 - rewrite uri root(nodes) to document-node() item. util:root(db:open-pre("ejemplo", 0)) -> db:open-pre("ejemplo", 0)
 - inline for \$baile, 0 in db:open-pre("ejemplo", 0)/bailes/baile
 - simplify FLWOR expression: db:open-pre("ejemplo", 0)/bailes/baile 1 <losbailes>{ nombre/text(), " (" , plazas/text(), ")" } </losbailes>
- Optimized Query:
 - db:open-pre("ejemplo", 0)/bailes/baile 1 <losbailes>{ nombre/text(), " (" , plazas/text(), ")" } </losbailes>
- Query:
 - for \$baile in /bailes/baile return <losbailes>{\$baile/nombre/text()} ({ \$baile/plazas/text() })</losbailes>
- Result:
 - Hits: 6 Items
 - Updated: 0 Items
 - Printed: 219 b
 - Read Locking: ejemplo
 - Write Locking: (none)
- Timing:
 - Parsing: 0.31 ms
 - Compiling: 0.77 ms
 - Evaluating: 0.83 ms
 - Printing: 0.09 ms
 - Total Time: 2.0 ms

- Mostrar los nombres de los bailes cuyo precio sea mayor de 30.

```
for $baile in /bailes/baile
where $baile/precio > 30
return $baile/nombre
```

The screenshot shows the BaseX 9.2.1 interface. The editor window contains the following XQuery:

```
1 for $baile in /bailes/baile
2 where $baile/precio > 30
3 return $baile/nombre
```

The data table displays the results of the query, showing 4 results. The table has columns for document ID, name, price, start, and end. The results are:

| 1 | 2 | 3 | 4 | 5 | 6 |
|---------------------------|--------|----------|-----------|-----------------------|--------|
| nomb. | precio | comienzo | fin | nomb. | precio |
| Tango | 27 | 1/1/2011 | 1/12/2011 | Rock | 30 |
| profesor Roberto García | sala 1 | | | profesor Jesus Lozano | sala 2 |
| plazas 20 | | | | plazas 12 | |
| Mer... | 75 | 1/1/2011 | 1/12/2011 | profesor Jesus Lozano | sala 2 |
| pas... | 320 | | | profesor Jesus Lozano | sala 2 |
| plazas 8 | | | | profesor Jesus Lozano | sala 2 |
| Cha-cha-cha | 80 | 1/2/2011 | 31/7/2011 | Salsa | 32 |
| profesor Miriam Gutiérrez | sala 1 | | | profesor Jesus Lozano | sala 2 |
| plazas 18 | | | | profesor Jesus Lozano | sala 2 |
| | | | | profesor Jesus Lozano | sala 2 |
| | | | | profesor Jesus Lozano | sala 2 |

The results pane shows the following XML output:

```
<nombre>Cha-cha-cha/</nombre>
<nombre>Merengue/</nombre>
<nombre>Salsa/</nombre>
<nombre>Pasodoble/</nombre>
```

The bottom pane shows the query execution details, including the compiled query, the optimized query, and the timing information.

- **Mostrar los nombres de los bailes cuyo precio sea mayor de 30 y la moneda "euro".**

for \$baile in /bailes/baile

where \$baile/precio > 30 and \$baile/precio/@moneda = "euro"

return \$baile/nombre

file [ejemplo] - BaseX 9.2.1
Database Editor View Visualization Options Help

Find: Find contents...

Context: db:open("ejemplo")

Editor

```

1 for $baile in /bailes/baile
2 where $baile/precio > 30 and $baile/precio/@moneda = "euro"
3 return $baile/nombre

```

OK 3 : 21

ejemplo.xml

| 1 | 2 | 3 | 4 | 5 | 6 |
|---------------------------|--------|----------|------------|---------------------------|--------|
| nomb. | precio | comienzo | fin | nomb. | precio |
| Tango | 27 | 1/1/2011 | 1/12/2011 | Rock | 30 |
| profesor Roberto García | 1 | sala | | profesor Jesus Lozano | 2 |
| plazas | 20 | | | plazas | 12 |
| cha-cha-cha | 80 | 1/2/2011 | 31/7/2011 | Salsa | 32 |
| profesor Miriam Gutiérrez | 1 | sala | | profesor Jesus Lozano | 2 |
| plazas | 18 | | | plazas | 10 |
| Pasodoble | 320 | 1/1/2011 | 31/12/2011 | profesor Miriam Gutiérrez | 1 |
| | | | | sala | |

3 Results, 80 b

```

<nombre>Cha-cha-cha</nombre>
<nombre>Salsa</nombre>
<nombre>Pasodoble</nombre>

```

Result

Total Time: 2.37 ms

Compiling:

- rewrite context value to document-node() item: .-> db:open-pre("ejemplo", 0)
- rewrite util:root(nodes) to document-node() item: util:root(db:open-pre("ejemplo", 0)) -> db:open-pre("ejemplo", 0)
- rewrite > comparison to range comparison: (\$baile_0/precio > 30) -> \$baile_0/precio >= 30.000000000000004
- rewrite to predicate: precio >= 30.000000000000004
- apply attribute index for "euro"
- rewrite to predicate: (precio/@moneda = "euro")
- inline for \$baile_0 in db:attribute("ejemplo", "euro")/self::attribute(moneda)/parent:precio/parent:baile[precio >= 30.000000000000004]
- simplify FLWOR expression: db:attribute("ejemplo", "euro")/self::attribute(moneda)/parent:precio/parent:baile[precio >= 30.000000000000004]/nombre

Optimized Query:

```
db:attribute("ejemplo", "euro")/self::attribute(moneda)/parent:precio/parent:baile[precio >= 30.000000000000004]/nombre
```

Query:

```
for $baile in /bailes/baile where $baile/precio > 30 and $baile/precio/@moneda = "euro" return $baile/nombre
```

Result:

- Hit(s): 3 Items
- Updated: 0 Items
- Printed: 80 b
- Read Locking: ejemplo
- Write Locking: (none)

Timing:

Time required: 2.37 ms

57 KB

- **Mostrar los nombres y la fecha de comienzo de los bailes que comiencen el mes de enero (utiliza para buscarlo la cadena de texto "/1/").**

for \$baile in /bailes/baile

where contains(\$baile/comienzo, "/1/")

return

```
<baile>
  {$baile/profesor}
  {$baile/comienzo}
</baile>
```

The screenshot shows the BaseX 9.7.1 interface with the following components:

- Editor:** Contains the XQuery:


```
1 for $baile in /bailes/baile
2 where contains($baile/comienzo, "/1/")
3 return
4   <baile>
5     {$baile/profesor}
6     {$baile/comienzo}
7   </baile>
```
- Visualizer:** Displays a table with 6 columns: nomb, precio, comienzo, fin, plazas, and profesor. It shows 3 results for dances starting in January.

| nomb | precio | comienzo | fin | plazas | profesor |
|--------|--------|----------|-----------|--------|-------------------------|
| Tango | 27 | 1/1/2011 | 1/12/2011 | 20 | profesor Roberto Garcia |
| Salsa | 32 | 1/3/2011 | 1/12/2011 | 15 | profesor Laura Mendiola |
| Mer... | 75 | 1/1/2011 | 1/12/2011 | 12 | profesor Jesus Lozano |
- Result:** Shows the XML output:


```
<baile>
  <profesor>Roberto Garcia</profesor>
  <comienzo>1/1/2011</comienzo>
</baile>
<baile>
  <profesor>Jesus Lozano</profesor>
  <comienzo>1/1/2011</comienzo>
</baile>
<baile>
  <profesor>Miriam Gutierrez</profesor>
  <comienzo>1/1/2011</comienzo>
</baile>
```
- Info:** Provides details about the query execution:
 - Compiling:** Rewrite context value to document-node(), item: -> db:open-pre("ejemplo", 0), Rewrite util:root(nodes) to document-node() item: util:root(db:open-pre("ejemplo", 0)) -> db:open-pre("ejemplo", 0), Rewrite to predicate: contains(comienzo, "/1/"), Inline for \$baile, 0 in db:open-pre("ejemplo", 0)/bailes/baile[contains(comienzo, "/1/")] -> db:open-pre("ejemplo", 0)/bailes/baile[contains(comienzo, "/1/")] | <baile>{ profesor, comienzo }</baile>
 - Optimized Query:** db:open-pre("ejemplo", 0)/bailes/baile[contains(comienzo, "/1/")] | <baile>{ profesor, comienzo }</baile>
 - Query:** for \$baile in /bailes/baile where contains(\$baile/comienzo, "/1/") return <baile> { (\$baile/profesor) (\$baile/comienzo) }</baile>
 - Result:** Hits: 3 items, Updated: 0 items, Printed: 271 b, Read Locking: ejemplo, Write Locking: (none)
 - Timing:** Parsing: 0.39 ms, Compiling: 0.97 ms, Evaluating: 2.84 ms, Printing: 0.15 ms

- **Mostrar los nombres de los profesores y la sala en la que dan clase, ordénalos por sala.**

for \$baile in /bailes/baile

order by \$baile/sala

return

<baile>

{ \$baile/profesor }

{ \$baile/sala }

</baile>

The screenshot shows the BaseX 9.7.1 interface with the following components:

- Editor:** Contains the XQuery:


```
1 for $baile in /bailes/baile
2 order by $baile/sala
3 return
4   <baile>
5     { $baile/profesor }
6     { $baile/sala }
7   </baile>
```
- Visual Tree:** A graphical representation of the query execution, showing the iteration over the /bailes/baile nodes and the resulting XML structure.
- Results:** Displays 6 results in a table format. The table has columns for 'nomb.' (name), 'precio' (price), 'comienzo' (start), and 'fin' (end). The results are grouped by 'baile' and ordered by 'sala'.
- Result Panel:** Shows the compiled XQuery:


```
<baile>
  <profesor>Roberto Garcia</profesor>
  <sala>1</sala>
</baile>
<baile>
  <profesor>Miriam Gutierrez</profesor>
  <sala>1</sala>
</baile>
<baile>
  <profesor>Laura Mendiola</profesor>
  <sala>1</sala>
</baile>
<baile>
  <profesor>Miriam Gutierrez</profesor>
  <sala>1</sala>
</baile>
<baile>
  <profesor>Jesus Lozano</profesor>
  <sala>2</sala>
</baile>
<baile>
  <profesor>Jesus Lozano</profesor>
  <sala>2</sala>
```
- Info Panel:** Provides details about the query execution, including:
 - Compiling:** 0.67 ms
 - Result:** 6 items, 454 b
 - Timing:** Parsing: 0.67 ms, Compiling: 1.09 ms, Evaluating: 1.93 ms, Printing: 0.19 ms, Total Time: 3.87 ms

- **Mostrar los nombres de los profesores eliminando los repetidos y acompañar cada nombre con todas las salas en la que da clase, ordénalos por nombre.**

```
for $profesor in distinct-values(/bailes/baile/profesor)
let $salas := /bailes/baile[profesor=$profesor]/sala
order by $profesor
return
```

```
<profesores>
  <nombre>{$profesor}</nombre>
  {$salas}
</profesores>
```

The screenshot shows the BaseX 8.7.1 interface. The editor contains the following XQuery:

```
1 for $profesor in distinct-values(/bailes/baile/profesor)
2 let $salas := /bailes/baile[profesor=$profesor]/sala
3 order by $profesor
4 return
5   <profesores>
6     <nombre>{$profesor}</nombre>
7     {$salas}
8   </profesores>
```

The data table displays the following information:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------|--------|----------|-----------|-------|--------|-----------|--------|----------|-----------|
| nomb. | precio | comienzo | fin | nomb. | precio | nomb. | precio | comienzo | fin |
| Tango | 27 | 1/1/2011 | 1/12/2011 | Rock | 30 | Mer... | 75 | 1/1/2011 | 1/12/2011 |
| | | | | | | profesor | | | |
| | | | | | | Roberto | | | |
| | | | | | | García | | | |
| | | | | | | sala | | | |
| | | | | | | 1 | | | |
| | | | | | | | | | |
| | | | | | | profesor | | | |
| | | | | | | Jesus | | | |
| | | | | | | Lozano | | | |
| | | | | | | sala | | | |
| | | | | | | 2 | | | |
| | | | | | | | | | |
| | | | | | | profesor | | | |
| | | | | | | Miriam | | | |
| | | | | | | Gutierrez | | | |
| | | | | | | sala | | | |
| | | | | | | 1 | | | |
| | | | | | | | | | |
| | | | | | | profesor | | | |
| | | | | | | Jesus | | | |
| | | | | | | Lozano | | | |
| | | | | | | sala | | | |
| | | | | | | 2 | | | |
| | | | | | | | | | |
| | | | | | | profesor | | | |
| | | | | | | Miriam | | | |
| | | | | | | Gutierrez | | | |
| | | | | | | sala | | | |
| | | | | | | 1 | | | |

The results pane shows the XML output:

```
<profesores>
  <nombre>Jesus Lozano</nombre>
  <sala>2</sala>
  <sala>2</sala>
</profesores>
<profesores>
  <nombre>Laura Mendiola</nombre>
  <sala>1</sala>
</profesores>
<profesores>
  <nombre>Miriam Gutierrez</nombre>
  <sala>1</sala>
  <sala>1</sala>
</profesores>
<profesores>
  <nombre>Roberto Garcia</nombre>
  <sala>1</sala>
</profesores>
```

The bottom pane shows the query execution details, including the compiled query and the result set.

- **Mostrar la media de los precios de todos los bailes.**

```
let $baile := /bailes/baile
```

```
return <media>{avg($baile/precio)}</media>
```

File [ejemplo] - BaseX 9.7.1
Database Editor View Visualization Options Help

Find Find...

Context: db:open("ejemplo")

Editor

```
1 let $baile := /bailes/baile
2 return <media>{avg($baile/precio)}</media>
```

OK 2 : 43

1 Result, 17 b

Result

```
<media>94</media>
```

Total Time: 1.43 ms

Info

Compiling:

- rewrite context value to document-node() item: -> db:open-pre("ejemplo", 0)
- rewrite util:root(nodes) to document-node() item: util:root(db:open-pre("ejemplo", 0)) -> db:open-pre("ejemplo", 0)
- flatten nested iter path: db:open-pre("ejemplo", 0)/bailes/baile
- rewrite fn:avg(items) to xs:double item: avg(db:open-pre("ejemplo", 0)/bailes/baile/precio) -> 94
- inline let \$baile_0 := db:open-pre("ejemplo", 0)/bailes/baile
- simplify FLWOR expression: <media>94</media>

Optimized Query:

```
<media>94</media>
```

Query:

```
let $baile := /bailes/baile return <media>{avg($baile/precio)}</media>
```

Result:

- Hit(s): 1 item
- Updated: 0 items
- Printed: 17 b
- Read Locking: ejemplo
- Write Locking: (none)

Timing:

- Parsing: 0.28 ms
- Compiling: 0.89 ms
- Evaluating: 0.21 ms

Time required: 1.43 ms

65 KB

- **Mostrar la suma de los precios de los bailes de la sala 1.**

```
let $baile := /bailes/baile[sala=1]
```

```
return <total_sala_1>{sum($baile/precio)}</total_sala_1>
```

The screenshot shows the BaseX 9.7.1 interface. The top menu bar includes File, Database, Editor, View, Visualization, Options, and Help. The main window is divided into three panes:

- Left Pane (File Explorer):** Shows the project structure for 'ejemplo'. It includes folders like 'bin', 'etc', 'ico', 'lib', 'repo', 'src', and 'webapp'. It also lists files such as 'BaseX.jar (4042 KB)', 'CHANGELOG (47 KB)', 'LICENSE (1274 B)', 'readme.txt (1802 B)', and 'uninstall.exe (31 KB)'.
- Editor:** Contains the XQuery code:


```
1 let $baile := /bailes/baile[sala=1]
2 return <total_sala_1>{sum($baile/precio)}</total_sala_1>
```
- XML Viewer:** Displays the 'ejemplo.xml' file. It shows a root element 'bailes' with three child elements (1, 3, 4, 6). Each child element contains a table of dance data with columns: nomb., precio, comienzo, fin, profesor, sala, and plazas.

| nomb. | precio | comienzo | fin | profesor | sala | plazas |
|--------|--------|----------|-----------|-------------------------|------|--------|
| Tango | 27 | 1/1/2011 | 1/12/2011 | profesor Roberto García | 1 | 20 |
| Rock | 30 | | | | | 15 |
| Mer... | 75 | 1/1/2011 | 1/12/2011 | profesor Jesus Lozano | 2 | 12 |
| Pas... | 320 | | | | | 8 |

Below the XML viewer, the **Results** pane shows the output of the query:


```
<total_sala_1>457</total_sala_1>
```

 The status bar indicates '1 Result, 32 b' and 'Total Time: 1.62 ms'.

- **Mostrar cuántas plazas en total ofrece el profesor "Jesus Lozano".**

```
let $baile := /bailes/baile[profesor = "Jesus Lozano"]
return <plazas>{sum($baile/plazas)}</plazas>
```

File [ejemplo] - BaseX 9.7.1
Database Editor View Visualization Options Help

Find Find...

Context: db:open("ejemplo")

Editor

```
1 let $baile := /bailes/baile[profesor = "Jesus Lozano"]
2 return <plazas>{sum($baile/plazas)}</plazas>
```

OK 2 : 45

1 Result

ejemplo.xml

| baile | | | | 3 | | | | 4 | | | | 6 | | | |
|--------|--------|-----------|-----------|----------|-----------|----------|------|--------|--------|----------|-----------|--------|--------|-----------|------------|
| nomb. | precio | comienzo | fin | nomb. | precio | comienzo | fin | nomb. | precio | comienzo | fin | nomb. | precio | comienzo | fin |
| Tango | 27 | 1/1/2011 | 1/12/2011 | Rock | 30 | | | Mer... | 75 | 1/1/2011 | 1/12/2011 | Pas... | 320 | | |
| | | profesor | sala | | | | | | | profesor | sala | | | | |
| | | Roberto | 1 | | | | | | | Jesus | 2 | | | | |
| | | García | | | | | | | | Lozano | | | | | |
| plazas | | | | plazas | | | | plazas | | | | plazas | | | |
| 20 | | | | 15 | | | | 12 | | | | 8 | | | |
| | | comienzo | fin | | | comienzo | fin | | | comienzo | fin | | | comienzo | fin |
| Cha- | 80 | 1/2/2011 | 31/7/2011 | 1/3/2011 | 1/12/2011 | | | Salsa | 32 | 1/5/2011 | 1/12/2011 | | | 1/1/2011 | 31/12/2011 |
| cha- | | | | | | | | | | | | | | | |
| | | profesor | sala | | | profesor | sala | | | profesor | sala | | | profesor | sala |
| | | Miriam | 1 | | | Laura | 1 | | | Jesus | 2 | | | Miriam | 1 |
| | | Gutier... | | | | Mendiola | | | | Lozano | | | | Gutier... | |
| plazas | | | | | | | | plazas | | | | | | | |
| 18 | | | | | | | | 10 | | | | | | | |

Result

1 Result, 19 b

```
<plazas>22</plazas>
```

Total Time: 1.68 ms

Info

Compiling:

- rewrite context value to document-node() item: -> db:open-pre("ejemplo", 0)
- rewrite util:root(nodes) to document-node() item: util:root(db:open-pre("ejemplo", 0)) -> db:open-pre("ejemplo", 0)
- apply text index for "Jesus Lozano"
- flatten nested cached path: db:text("ejemplo", "Jesus Lozano")/parent:profesor/parent:baile
- inline let \$baile_0 := db:text("ejemplo", "Jesus Lozano")/parent:profesor/parent:baile
- simplify FLWOR expression: <plazas>{ sum(db:text("ejemplo", "Jesus Lozano")/parent:profesor/parent:baile/plazas) } </plazas>

Optimized Query:

```
<plazas>{ sum(db:text("ejemplo", "Jesus Lozano")/parent:profesor/parent:baile/plazas) } </plazas>
```

Query:

```
let $baile := /bailes/baile[profesor = "Jesus Lozano"] return <plazas>{sum($baile/plazas)}</plazas>
```

Result:

- Hit(s): 1 item
- Updated: 0 items
- Printed: 19 b
- Read Locking: ejemplo
- Write Locking: (none)

Timing:

- Parsing: 0.28 ms
- Compiling: 0.89 ms
- Evaluating: 0.37 ms

Time required: 1.68 ms

- **Mostrar el dinero que ganaría la profesora "Laura Mendiola" si se completaran todas las plazas de su baile, sabiendo que sólo tiene un baile.**

```
let $baile := /bailes/baile[profesor = "Laura Mendiola"]
return $baile/plazas * $baile/precio
```

File [ejemplo] - BaseX 9.7.1
Database Editor View Visualization Options Help

Find

Find...

C:\Program Files (x86)\BaseX\

bin
etc
ico
lib
repo
etc
webapp
BaseX.jar (4042 KB)
CHANGELOG (47 KB)
LICENSE (1254 B)
readme.txt (1802 B)
uninstall.exe (31 KB)

Find contents...

1 let \$baile := /bailes/baile[profesor = "Laura Mendiola"]
2 return \$baile/plazas * \$baile/precio

OK 2 : 37

ejemplo.xml

| | | | | | | | | | | | | | | | |
|-------------------------|--------|----------|-----------|-----------------------|--------|----------|-----------|-----------------------|--------|----------|-----------|---------------------------|--------|----------|-----------|
| bailes | | | | 3 | | | | 4 | | | | 6 | | | |
| nomb. | precio | comienzo | fin | nomb. | precio | comienzo | fin | nomb. | precio | comienzo | fin | nomb. | precio | comienzo | fin |
| Tango | 27 | 1/1/2011 | 1/12/2011 | Rock | 30 | 1/1/2011 | 1/12/2011 | Mer... | 75 | 1/1/2011 | 1/12/2011 | Pas... | 320 | 1/1/2011 | 1/12/2011 |
| profesor Roberto García | | | | profesor Jesus Locano | | | | profesor Jesus Locano | | | | profesor Miriam Gutier... | | | |
| sala 1 | | | | sala 2 | | | | sala 2 | | | | sala 1 | | | |
| plazas 20 | | | | plazas 12 | | | | plazas 8 | | | | plazas 18 | | | |

1 Result

1 Result, 3 b

450

Result

Total Time: 1.64 ms

Info

Compiling:

- rewrite context value to document-node() item: -> db:open-pre("ejemplo", 0)
- rewrite util:root(nodes) to document-node() item: util:root(db:open-pre("ejemplo", 0)) -> db:open-pre("ejemplo", 0)
- apply text index for "Laura Mendiola"

Optimized Query:

let \$baile_0 := db:text("ejemplo", "Laura Mendiola")parent:profesorparent:baile return (\$baile_0/plazas * \$baile_0/precio)

Query:

let \$baile := /bailes/baile[profesor = "Laura Mendiola"] return \$baile/plazas * \$baile/precio

Result:

- Hit(s): 1 item
- Updated: 0 items
- Printed: 0 b
- Read Locking: ejemplo
- Write Locking: (none)

Timing:

- Parsing: 0.35 ms
- Compiling: 0.62 ms
- Evaluating: 0.63 ms
- Printing: 0.04 ms
- Total Time: 1.64 ms

Query Plan

Time required: 1.64 ms

54 KB

- **Mostrar el dinero que ganaría el profesor "Jesus Lozano" si se completaran todas las plazas de su baile, pero mostrando el beneficio de cada baile por separado.**

```
for $baile in /bailes/baile
where $baile/profesor = "Jesus Lozano"
return <beneficio>{$baile/plazas * $baile/precio}</beneficio>
```

The screenshot shows the BaseX 9.2.1 interface with the following components:

- Editor:** Contains the XQuery:


```
1 for $baile in /bailes/baile
2 where $baile/profesor = "Jesus Lozano"
3 return <beneficio>{$baile/plazas * $baile/precio}</beneficio>
```
- ejemplo.xml:** An XML document representing dance classes. It has a root element 'bailes' with multiple 'baile' elements as children. Each 'baile' element contains 'nomb.' (name), 'precio' (price), 'comienzo' (start), 'fin' (end), 'plazas' (spots), and 'profesor' (teacher).

| baile | nomb. | precio | comienzo | fin | plazas | profesor |
|-------|--------|--------|----------|------------|--------|---------------------------|
| 1 | Tango | 27 | 1/1/2011 | 1/12/2011 | 1 | profesor Roberto García |
| 3 | Rock | 30 | 1/3/2011 | 1/12/2011 | 1 | profesor Laura Mendiola |
| 4 | Mer... | 75 | 1/1/2011 | 1/12/2011 | 2 | profesor Jesus Lozano |
| 5 | Salsa | 32 | 1/5/2011 | 1/12/2011 | 2 | profesor Jesus Lozano |
| 6 | Pas... | 320 | 1/1/2011 | 31/12/2011 | 1 | profesor Miriam Gutiérrez |
- Results:** Shows 2 results for the query:


```
<beneficio>900</beneficio>
<beneficio>320</beneficio>
```
- Info:** Provides details about the query execution:
 - Compiling:** Lists various optimizations like rewriting context values and predicates.
 - Optimized Query:** Shows the final query after optimizations:


```
db:text("ejemplo", "Jesus Lozano")/parent:profesor/parent:baile 1 <beneficio>{(plazas * precio)}</beneficio>
```
 - Query:** Shows the original query being executed.
 - Result:** Shows the output of the query.
 - Timing:** Shows execution times: Parsing: 0.36 ms, Compiling: 1.02 ms, Evaluating: 0.34 ms.

- Mostrar el dinero que ganaría la profesora "Laura" (no conocemos su apellido) si se completaran todas las plazas de su baile.

```
for $baile in /bailes/baile[starts-with(profesor, "Laura")]
return <beneficio>{$baile/plazas * $baile/precio}</beneficio>
```

The screenshot shows the BaseX 9.7.1 interface. The editor contains the following XQuery:

```
1 for $baile in /bailes/baile[starts-with(profesor, "Laura")]
2 return <beneficio>{$baile/plazas * $baile/precio}</beneficio>
```

The results pane displays a table with 6 columns: nomb., precio, comienzo, fin, plazas, and profesor. The data is as follows:

| nomb. | precio | comienzo | fin | plazas | profesor |
|-------------|--------|----------|-----------|--------|----------------------------|
| Tango | 27 | 1/1/2011 | 1/12/2011 | 1 | profesor Roberto Garcia |
| Rock | 30 | | | 15 | |
| Mer... | 75 | 1/1/2011 | 1/12/2011 | 2 | profesor Jesus Locano |
| Cha-cha-cha | 80 | 1/2/2011 | 31/7/2011 | 1 | profesor Pluriam Gutier... |
| Salsa | 32 | 1/3/2011 | 1/12/2011 | 1 | profesor Laura Mendiola |
| | | | | 10 | |

The result pane shows the output: <beneficio>450</beneficio>. The bottom status bar indicates "Time required: 1.56 ms".

- **Mostrar el nombre del baile, su precio y el precio con un descuento del 15% para familias numerosas. Ordenar por el nombre del baile**

```
for $baile in /bailes/baile
order by $baile/nombre
return
```

```
<baile>
  {$baile/nombre}
  <precio>{$baile/precio/text()}</precio>
  <fam_numerosa>{$baile/precio * 0.85}</fam_numerosa>
</baile>
```

6 Results

| 1 | 2 | 3 | 4 | 5 | 6 |
|----------|----------|-------------|----------|----------|----------|
| nombre | pla | nombre | pla | nombre | pla |
| Tango | 20 | Cha-cha-cha | 18 | Merengue | 12 |
| precio | | precio | | precio | |
| 27 | 80 | 30 | 75 | 32 | 320 |
| comienzo | profesor | comienzo | profesor | comienzo | profesor |
| 1/1/2011 | 1/2/2011 | 1/3/2011 | 1/1/2011 | 1/5/2011 | 1/1/2011 |
| fin | sala | fin | sala | fin | sala |
| 1 | 1 | 1 | 2 | 2 | 1 |

6 Results, 640 b

```
<baile>
  <nombre>Cha-cha-cha</nombre>
  <precio>80</precio>
  <fam_numerosa>68</fam_numerosa>
</baile>
<baile>
  <nombre>Merengue</nombre>
  <precio>75</precio>
  <fam_numerosa>63.75</fam_numerosa>
</baile>
<baile>
  <nombre>Pasodoble</nombre>
  <precio>320</precio>
  <fam_numerosa>272</fam_numerosa>
</baile>
<baile>
  <nombre>Rock</nombre>
  <precio>30</precio>
  <fam_numerosa>25.5</fam_numerosa>
</baile>
<baile>
  <nombre>Salsa</nombre>
  <precio>32</precio>
  <fam_numerosa>27.2</fam_numerosa>
</baile>
<baile>
  <nombre>Tango</nombre>
  <precio>27</precio>
  <fam_numerosa>22.95</fam_numerosa>
</baile>
```

Result

Compiling:

- rewrite context value to document-node() item: -> db-open-pre("ejemplo", 0)
- rewrite util:root(nodes) to document-node() item: util:root(db-open-pre("ejemplo", 0)) -> db-open-pre("ejemplo", 0)

Optimized Query:

(for \$baile_0 in db-open-pre("ejemplo", 0)/bailes/baile order by \$baile_0/nombre empty least return <baile>{ \$baile_0/nombre, <precio>{ \$baile_0/precio/text() }</precio>, <fam_numerosa>{ (\$baile_0/precio * 0.85) }</fam_numerosa> }</baile>)

Query:

for \$baile in /bailes/baile order by \$baile/nombre return <baile> { \$baile/nombre }<precio>{ \$baile/precio/text() }</precio> <fam_numerosa>{ \$baile/precio * 0.85 }</fam_numerosa> </baile>

Result:

- Hit(s): 6 items
- Updated: 0 items
- Printed: 640 b
- Read Locking: ejemplo
- Write Locking: (none)

Timing:

- Parsing: 0.42 ms
- Compiling: 0.51 ms
- Evaluating: 1.18 ms
- Printing: 0.11 ms
- Total Time: 2.2 ms

Query Plan:

```
<QueryPlan compiled="true" updating="false">
  <GLWCR type="element(baile)" size="0">
    <For type="element(baile)" size="1" name="$baile" id="0">
      <IterPath type="element(baile)" size="0" database="ejemplo">
        <DBNode pre="0" type="document-node()" size="1" database="ejemplo">
          <IterStep axis="child" test="$bailes" type="element(bailes)">
```

- Mostrar todos los datos de cada baile excepto la fecha de comienzo y de fin.

for \$baile in /bailes/baile

return <baile>{\$baile/* except \$baile/comienzo except \$baile/fin}</baile>

The screenshot shows the BaseX 9.7.1 interface with the following components:

- Editor:** Contains the XQuery query:


```
1 for $baile in /bailes/baile
2 return <baile>{$baile/* except $baile/comienzo except $baile/fin}</baile>
```
- Results:** Displays 6 results, 1085 b. The results are XML fragments for different dances, each containing details like name, price, and location.

| 1 | 2 | 3 | 4 | 5 | 6 |
|---|--|--|--|---|--|
| <pre><nombre>Tango/<nombre> <precio cuota="mensual" moneda="euro">27</precio> <plazas>28</plazas> <profesor>Roberto Garcia</profesor> <sala>1</sala> </baile></pre> | <pre><nombre>Cha-cha-cha/<nombre> <precio cuota="trimestral" moneda="euro">88</precio> <plazas>18</plazas> <profesor>Miriam Gutierrez</profesor> <sala>1</sala> </baile></pre> | <pre><nombre>Rock/<nombre> <precio cuota="mensual" moneda="euro">30</precio> <plazas>15</plazas> <profesor>Laura Mendiola</profesor> <sala>1</sala> </baile></pre> | <pre><nombre>Merengue/<nombre> <precio cuota="trimestral" moneda="dolares">75</precio> <plazas>12</plazas> <profesor>Jesus Lozano</profesor> <sala>2</sala> </baile></pre> | <pre><nombre>Salsa/<nombre> <precio cuota="mensual" moneda="euro">32</precio> <plazas>18</plazas> <profesor>Jesus Lozano</profesor> <sala>2</sala> </baile></pre> | <pre><nombre>Pasodoble/<nombre> <precio cuota="anual" moneda="euro">328</precio> <plazas>8</plazas> <profesor>Miriam Gutierrez</profesor> <sala>1</sala></pre> |
- Compiling:** Shows the compilation process with the following steps:
 - rewrite context value to document-node() item: ...-> db-open-pre("ejemplo", 0)
 - rewrite util:root(nodes) to document-node() item: util:root(db-open-pre("ejemplo", 0))-> db-open-pre("ejemplo", 0)
 - remove true() from and
 - remove true() from and
 - inline for \$baile_0 in db-open-pre("ejemplo", 0)/bailes/baile
 - simplify FLWOR expression: db-open-pre("ejemplo", 0)/bailes/baile | <baile>{ (* except comienzo except fin) }-><baile>
- Optimized Query:**

```
db-open-pre("ejemplo", 0)/bailes/baile | <baile>{ (* except comienzo except fin) }-><baile>
```
- Query:**

```
for $baile in /bailes/baile return <baile>{$baile/* except $baile/comienzo except $baile/fin}</baile>
```
- Result:**
 - Hit(s): 6 items
 - Updated: 0 items
 - Printed: 1085 b
 - Read Locking: ejemplo
 - Write Locking: (none)
- Timing:**
 - Parsing: 0.25 ms
 - Compiling: 1.48 ms
 - Evaluating: 1.36 ms
 - Printing: 0.18 ms
 - Total Time: 3.27 ms
- Query Plan:**

```
<QueryPlan compiled="true" updating="false">
  <DualMap type="element(baile)" size="0">
    <IterPath type="element(baile)" size="0" database="ejemplo">
      <DBNode pre="0" type="document-node()" size="1" database="ejemplo">
        <IterStep axis="child" test="bailes" type="element(bailes)">
          <IterStep axis="child" test="baile" type="element(baile)">
            <IterPath>
              <CElem type="element(baile)" size="1">
                <QNm type="xs:QName" size="1">baile</QNm>
                <Except Iterative="true" type="element()" database="ejemplo">
                  <SingleIterPath type="element()" database="ejemplo">
                    <IterStep axis="child" test="" type="element()">
                      <SingleIterPath>
                        <SingleIterPath type="element(comienzo)" database="ejemplo">
```


- **Mostrar en una tabla de HTML los nombres de los bailes y su profesor, cada uno en una fila**

```
<table> {
  for $baile in /bailes/baile
  return <tr><td>{$baile/nombre/text()}</td><td>{$baile/profesor/text()}</td></tr>
} </table>
```

File [ejemplo] - BaseJ 5.7.1

Database Editor View Visualization Options Help

Find

C:\Program Files (x86)\BaseJ\

- *xml*.xsd*

Find contents...

- basej
- bin
- etc
- log
- lib
- nom
- src
- webapp
- BaseJ.jar (4942 KB)
- CHANGELOG (47 KB)
- LICENSE (1024 B)
- main.xml (5502 B)
- uninstall.exe (91 KB)

OK

Editor Context: db:open("ejemplo")

```
<?xml version='1.0' encoding='UTF-8'?>
<table> {
1   <tr><td>$baile/nombre/text()</td><td>$baile/profesor/text()</td></tr>
2 } </table>
```

1 Result, 425 b

```
<table>
<tr>
<td>Tango</td>
<td>Roberto Garcia</td>
</tr>
<tr>
<td>Cha-cha-cha</td>
<td>Miriam Gutierrez</td>
</tr>
<tr>
<td>Rock</td>
<td>Laura Mendiola</td>
</tr>
<tr>
<td>Merengue</td>
<td>Jesus Lozano</td>
</tr>
<tr>
<td>Salsa</td>
<td>Jesus Lozano</td>
</tr>
<tr>
<td>Pasodoble</td>
<td>Miriam Gutierrez</td>
</tr>
</table>
```

Result Total Time: 1.58 ms

| nomb. | precio | comienzo | fin | nomb. | precio | comienzo | fin | nomb. | precio | comienzo | fin |
|---|----------|----------|-----------|----------|----------|----------|-----------|--|------------|----------|-----|
| Tango | 27 | 1/1/2011 | 1/12/2011 | Rock | 30 | Mer.. | 75 | 1/1/2011 | 1/12/2011 | Pas.. | 320 |
| profesor | sala | | | profesor | sala | | | profesor | sala | | |
| Roberto | Garcia | | | | | | | Jesus | Lozano | | |
| plazas | | | | plazas | | | | plazas | | | |
| 20 | | | | 15 | | | | 8 | | | |
| nomb. <th>precio</th> <th>comienzo</th> <th>fin</th> <td></td> <td></td> <td></td> <td></td> <td>nomb. <th>precio</th> <th>comienzo</th> <th>fin</th> </td> | precio | comienzo | fin | | | | | nomb. <th>precio</th> <th>comienzo</th> <th>fin</th> | precio | comienzo | fin |
| Cha- | 80 | 1/2/2011 | 31/7/2011 | Salsa | 32 | 1/5/2011 | 1/12/2011 | 1/1/2011 | 31/12/2011 | | |
| profesor | sala | | | profesor | sala | | | profesor | sala | | |
| Miriam | Gutier.. | | | Laura | Mendiola | | | Jesus | Lozano | | |
| plazas | | | | plazas | | | | plazas | | | |
| 18 | | | | 10 | | | | | | | |

Time required: 1.58 ms

Info

Compiling:

- rewrite context value to document-node() item: -> db:open-pre("ejemplo", 0)
- rewrite util/root(nodes) to document-node() item: util/root(db:open-pre("ejemplo", 0)) -> db:open-pre("ejemplo", 0)
- mine for \$baile, 0 in db:open-pre("ejemplo", 0)/bailes/baile
- simplify FLWOR expression: db:open-pre("ejemplo", 0)/bailes/baile |<tr> <td>{ nombre;text() }</td> <td>{ profesor;text() }</td></tr>

Optimized Query:

```
<table> (db:open "ejemplo", 0)/bailes/baile |<tr> <td>(nombre;text())</td> <td>(profesor;text())</td></tr> </table>
```

Query:

```
<table> (for $baile in /bailes/baile return <tr><td>{$baile/nombre/text()}</td><td>{$baile/profesor/text()}</td></tr>) </table>
```

Result:

- Hit(s): 1 Item
- Updated: 0 Items
- Printed: 425 b
- Read Locking: ejemplo
- Write Locking: (none)

Timing:

- Parsing: 0.28 ms
- Compiling: 0.53 ms
- Evaluating: 0.71 ms
- Printing: 0.05 ms
- Total Time: 1.58 ms

Query Plan:

```
<QueryPlan compiled="true" updating="false">
  <CElem type="element/table" size="14">
```