

Databanken 3

Data Transformatie

Data transformaties

Inhoudstafel

1. **Data transformatie**
2. ETL
3. XML, XSL en XPATH
4. JSON

Data transformatie

**Methodiek om data van één
vorm naar een andere
vorm te transformeren.**



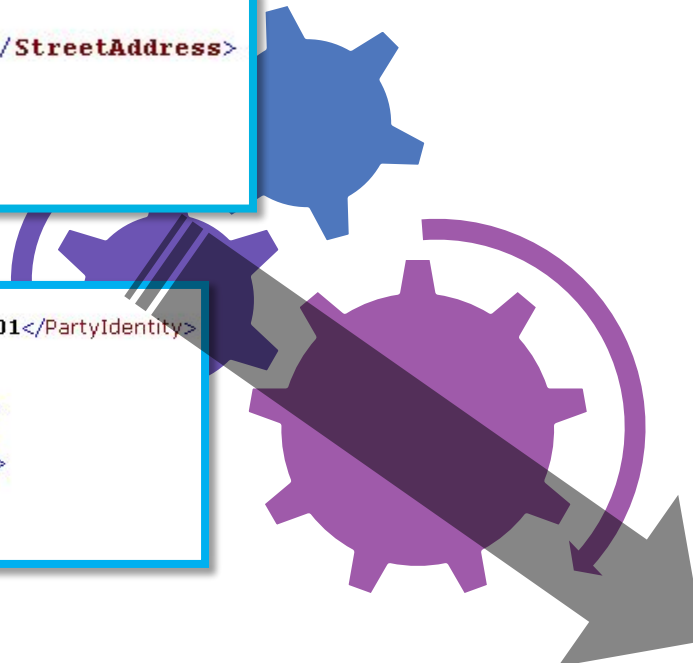
Data transformatie

Use cases

- **Verschillende externe formaten importeren in eigen systeem**

```
<Address>
  <StreetAddress>512 S. State St</StreetAddress>
  <City>Provo</City>
  <State>UT</State>
  <Zip>84601</Zip>
</Address>
```

```
<Party Role="Buyer">
  <PartyIdentity Type="EAN">5013546000001</PartyIdentity>
  <Name>ABC Stores Ltd.</Name>
  <Address>
    <AddressLine>Head Office</AddressLine>
    <AddressLine>ABC House</AddressLine>
    <AddressLine>Fremlington</AddressLine>
    <City>Maidenhead</City>
  </Address>
</Party>
```



Land	Stad	Postcode	Straat	Huis nummer
USA	Provo	84601	S. State ST	512
USA	Maidenhead	65240	Flemington	

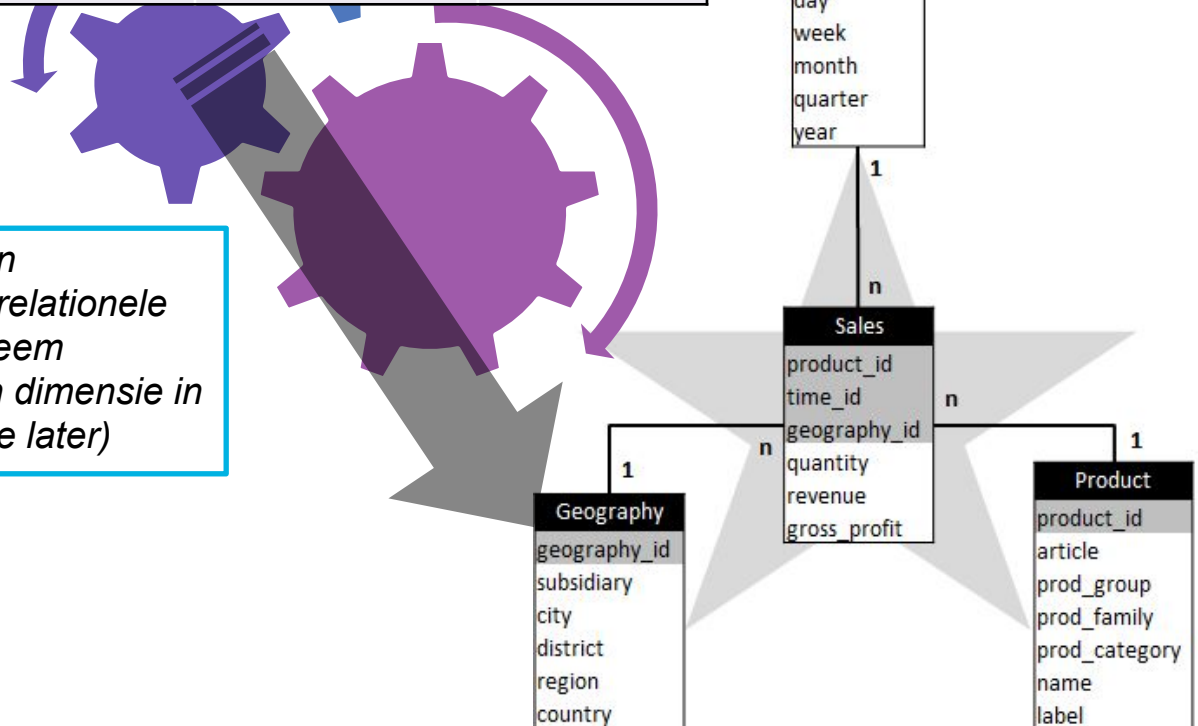
Data transformatie

Use cases

- Overbrengen en/of synchroniseren gegevens tussen verschillende systemen

Land	Stad	Postcode	Straat	Huisnummer
USA	Provo	84601	S. State ST	512
USA	Maidenhead	65240	Flemington	

In dit voorbeeld worden adresgegevens uit de relationele databank van het systeem overgebracht naar een dimensie in het datawarehouse (zie later)



ETL

Extract Transform Load

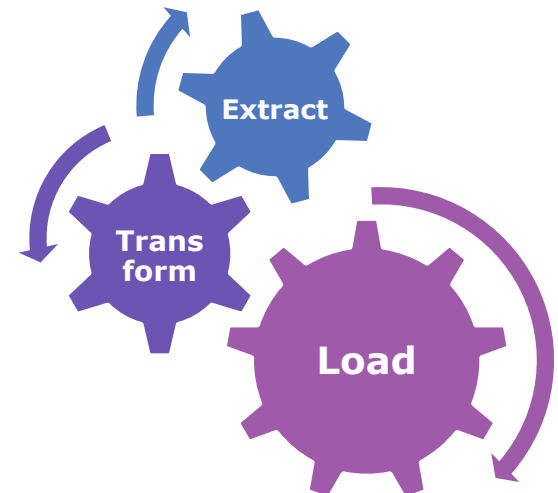
Inhoudstafel

1. Data transformatie
2. **ETL**
3. XML, XSL en XPATH
4. JSON

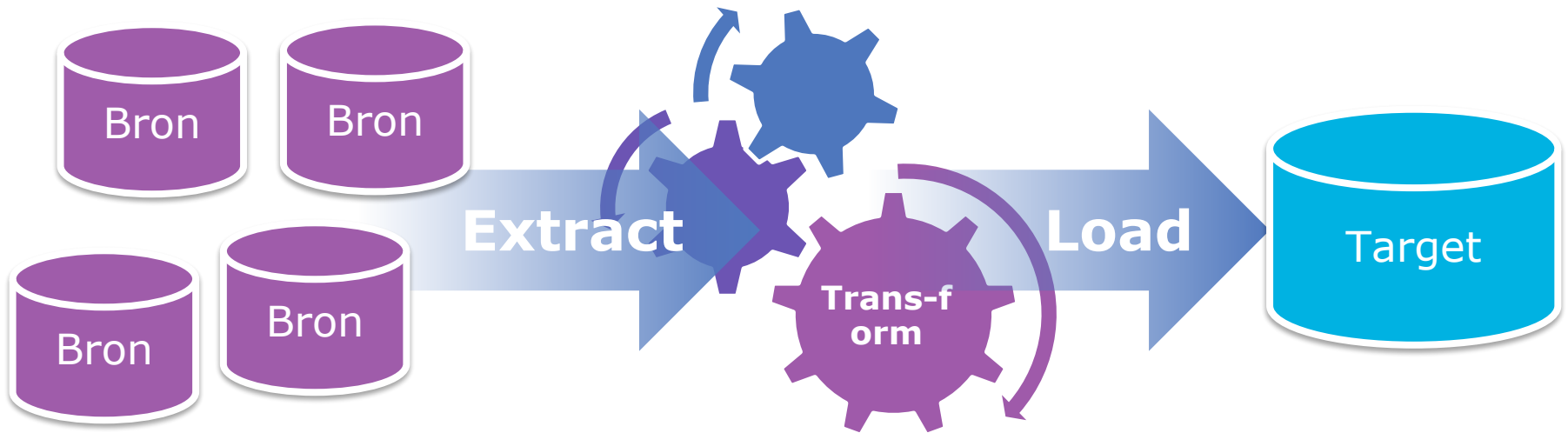
ETL

Extract transform load

- ETL:
Extraheren van data **uit** verschillende **bronnen** om het vervolgens te **Transformeren** naar het gewenste formaat en het tenslotte te **Laden** in de **target**.
- **Hoe ETL programmeren?**
 - Door **zelf** alles te **ontwikkelen** en coderen
 - Door **gebruik** te **maken** van **ETL tools** die out of the box:
 - Veel connectiemogelijkheden bieden
 - Veel transformatiemogelijkheden bieden



Extract Transform Load



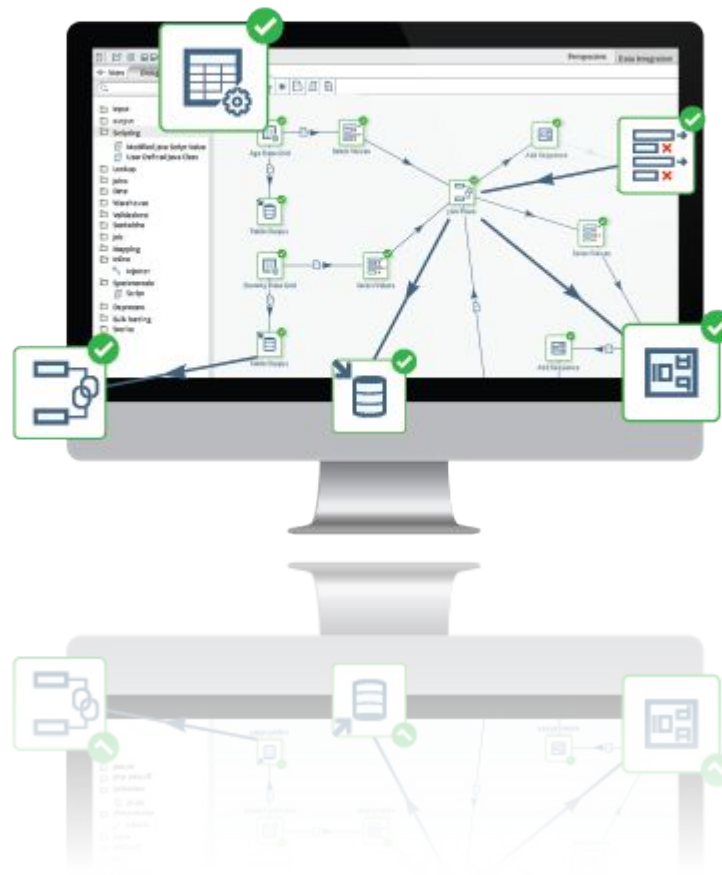
- Locatie:
 - Intern
 - Extern
- Type:
 - Database
 - CSV
 - XML
 - ...

- Join
- Filter
- Transform
- Lookup
- Split
- Sort
- Aggregate
- ...

- Systeem
 - Datawarehouse
 - Operationeel
 - ...
- Type
 - Database
 - CSV
 - XML
 - ...

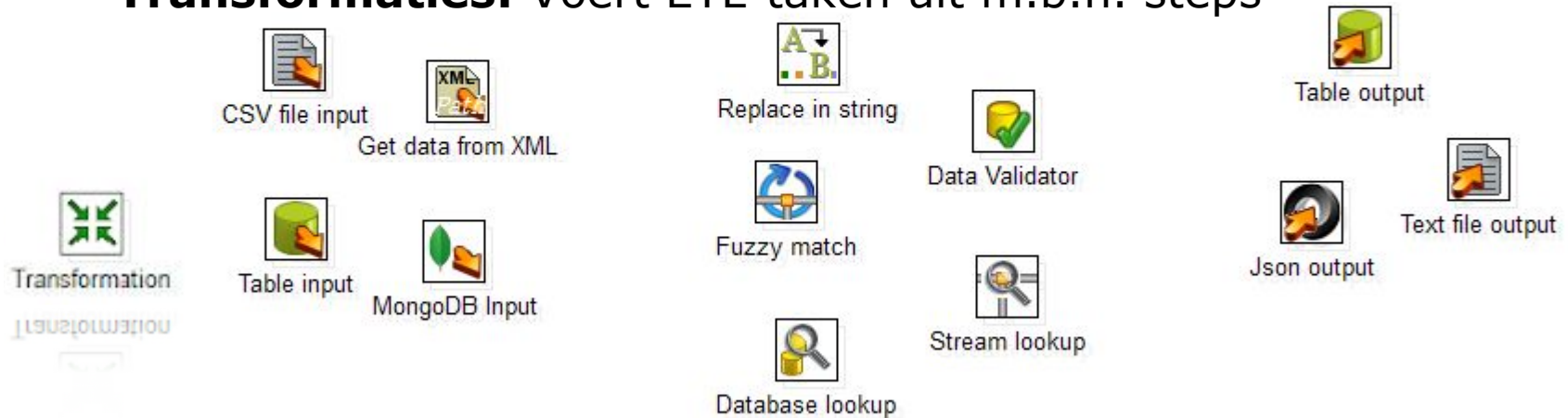
ETL Tools

- Bieden functionaliteit om een heel brede waaier aan transformaties uit te voeren
- Wij maken gebruik van Pentaho Data Integration

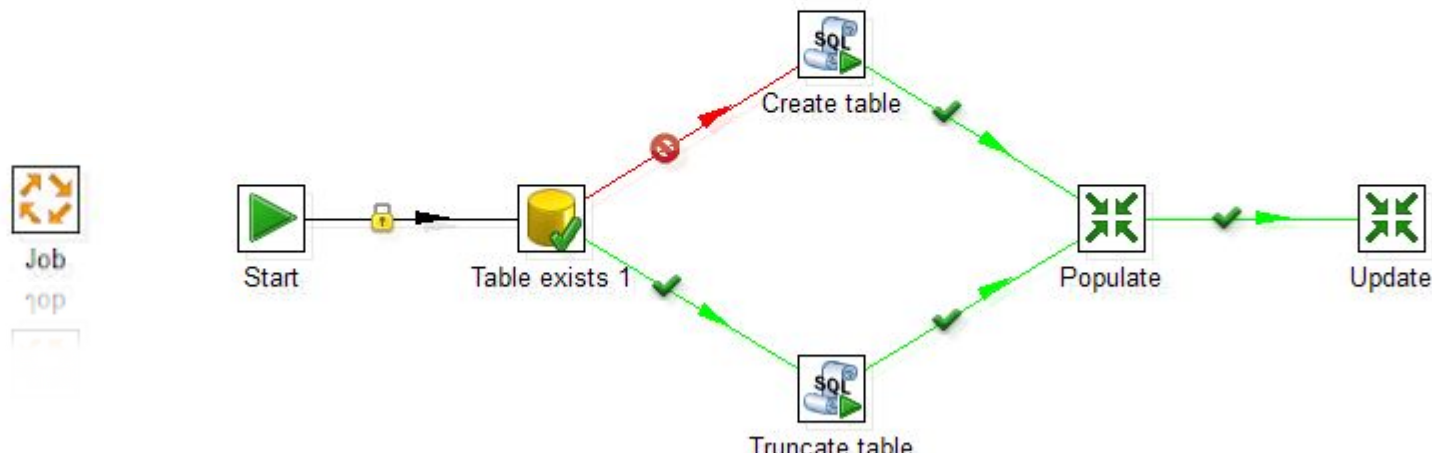


ETL Tool Componenten

- **DatabaseConnections:** Vooraf ingesteld om ze daarna in transformaties en jobs te gebruiken
- **Transformaties:** Voert ETL-taken uit m.b.h. steps



- **Jobs:** Staan in om ETL-taken te coördineren



ETL Tool Pentaho Data Integration

The screenshot shows the Pentaho Data Integration (Spoon) interface. The top menu bar includes File, Edit, View, Action, Tools, and Help. Below the menu is a toolbar with icons for adding, saving, and running jobs. The left sidebar shows a 'Steps' panel with categories like Input, Output, Transform, Utility, Flow, Scripting, BA Server, and Lookup. The main workspace displays a workflow diagram with steps: Table input, Database lookup, Dimension lookup/update, and Insert. The 'Execution Results' panel at the bottom shows a table with columns: Stepname, Copynr, Read, Written, and Inj. The table contains data for the four steps in the workflow.

Druk hierop om reeds gemaakte transformaties/jobs te laden

Als je transformatie klaar is kan je ze uitvoeren

- Hop: Geeft aan dat de uitput van step Table de input is van step Database lookup

Hier kies je de steps. Die je in je transformatie wil plaatsen. Typisch start je met een Input Step en eindig je met een Output step. Daartussen komen transformaties / Joins / Lookups...

Bij het uitvoeren van een transformatie krijg je heel veel informatie

Stepname	Copynr	Read	Written	Inj
Table input	0	0	0	
Database lookup	0	0	0	
Dimension lookup/update	0	0	0	
Insert Update	0	0	0	

`XML, XSL en XPATH`

Inhoudstafel

1. Data transformatie
2. ETL
3. **XML, XSL en XPATH**
4. JSON

XML

- **XML wordt vaak gebruikt als:**
 - **formaat om gegevens over te dragen tussen partijen**
 - taal om UI op te bouwen
 - formaat voor configuratiebestanden

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Comment element-->
<user>
  <name>Casper</name>
  <date-of-birth>
    23-10-2002
  </date-of-birth>
  <address type="BE">
    <street>Nationalestraat</street>
    <number>5</number>
    <postalcode>2000</postalcode>
    <city>Antwerp</city>
  </address>
  <scores>
    <score>16</score>
    <score>9</score>
  </scores>
  <messages/>
</user>
```

Eerste
element =
root element

start
tag

content

End
tag

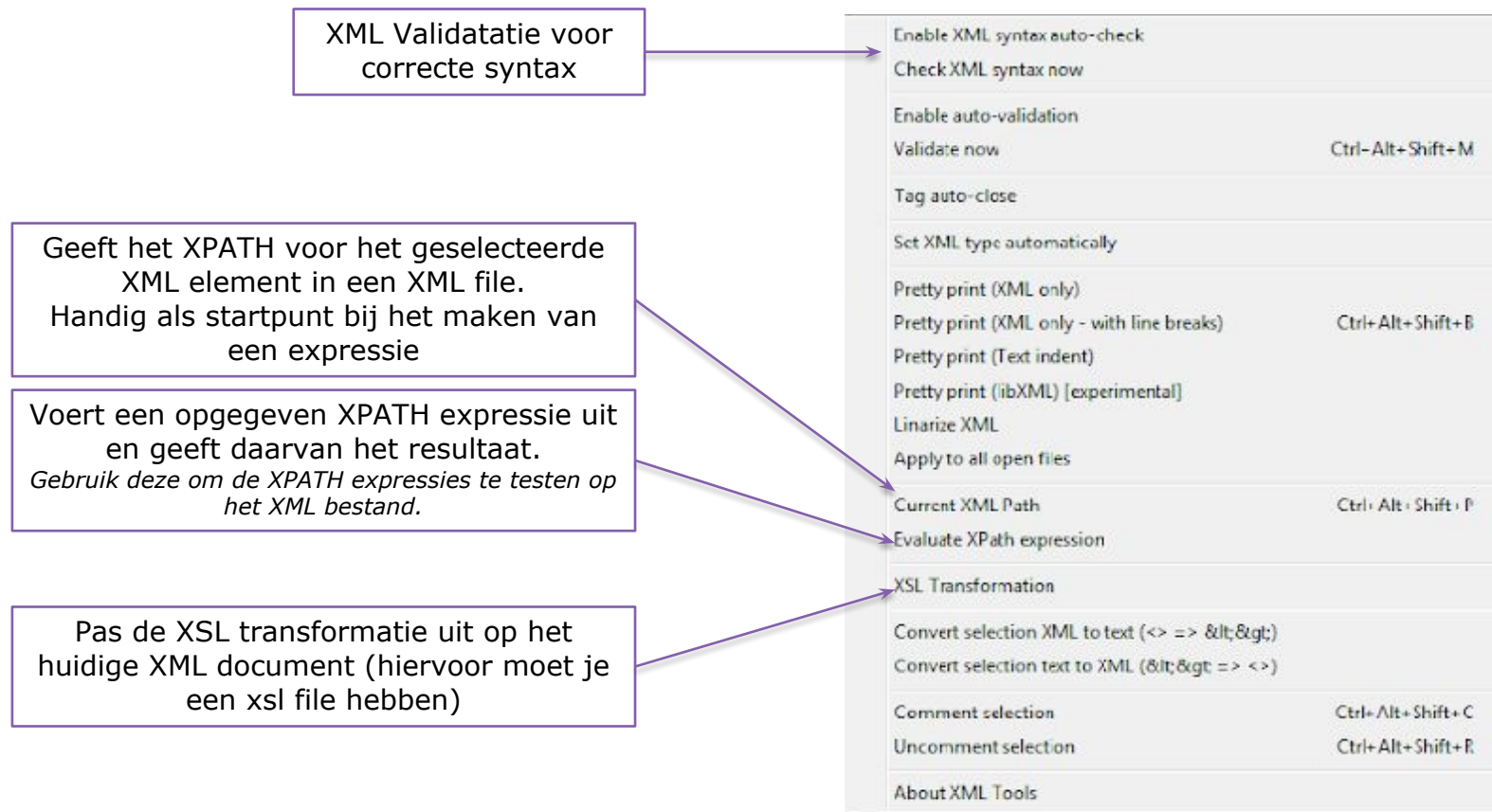
Attribute

Parent versus children

Element zonder content

XML, XSL en XPATH tools

- **Gebruik notepad++ om XML te editen.**
- **Installeer via Plugins > PluginManager: XML Tools**
- **Selecteer XML Tools onder plugins**



XPATH

- querytaal voor XML
- *Zie voor meer info*

http://www.w3schools.com/xsl/xpath_syntax.asp

```
<?xml version="1.0" encoding="UTF-8"?>
<Collections testDescription="1 - simple functionality
test">
  <Rows rowID="1">
    <Row><v1>1.1.1</v1><v2>1.1.2</v2></Row>
    <Row><v1>1.2.1</v1><v2>1.2.2</v2></Row>
  </Rows>
  <Rows rowID="2">
    <Row><v1>2.1.1</v1><v2>2.1.2</v2></Row>
    <Row><v1>2.2.1</v1><v2>2.2.2</v2></Row>
  </Rows>
  <Rows rowID="3">
    <Row><v1>3.1.1</v1><v2>3.1.2</v2></Row>
    <Row><v1>3.2.1</v1><v2>3.2.2</v2></Row>
  </Rows>
</Collections>
```

XPathexpressie	Result	Uitleg
/Collections/Rows	Lijst met 3 Node-elements met de naam Rows (met rowID=1, 2 en 3)	Geef alle Rows elementen binnen Collections elementen. Als er meerdere rijen voldoen aan de query dan worden deze als lijst met elementen teruggegeven.
/Collections/Rows[@rowID='1']	1 Node element: <Rows rowID="1"> <Row><v1>1.1.1</v1><v2>1.1.2</v2></Row> <Row><v1>1.2.1</v1><v2>1.2.2</v2></Row> </Rows>	[@rowID='1'] is een filter: Selecteer enkel de Rows nodes met het attribuut rowID gelijk aan 1
//Rows	Zelfde als eerste result	// Wil zeggen 'zoek eender waar in de hiërarchie naar Rows elementen (vaak als shortcut gebruikt)

XPATH

- querytaal voor XML
- *Zie voor meer info*

http://www.w3schools.com/xsl/xpath_syntax.asp

```
<?xml version="1.0" encoding="UTF-8"?>
<Collections testDescription="1 - simple functionality
test">
  <Rows rowID="1">
    <Row><v1>1.1.1</v1><v2>1.1.2</v2></Row>
    <Row><v1>1.2.1</v1><v2>1.2.2</v2></Row>
  </Rows>
  <Rows rowID="2">
    <Row><v1>2.1.1</v1><v2>2.1.2</v2></Row>
    <Row><v1>2.2.1</v1><v2>2.2.2</v2></Row>
  </Rows>
  <Rows rowID="3">
    <Row><v1>3.1.1</v1><v2>3.1.2</v2></Row>
    <Row><v1>3.2.1</v1><v2>3.2.2</v2></Row>
  </Rows>
```

XPathexpressie	Result	Uitleg
/Collections/Rows	Lijst met 3 Node-elements met de naam Rows (met rowID=1, 2 en 3)	Geef alle Rows elementen binnen Collections elementen. Als er meerdere rijen voldoen aan de query dan worden deze als lijst met elementen teruggegeven.
/Collections/Rows[@rowID='1']	1 Node element: <Rows rowID="1"> <Row><v1>1.1.1</v1><v2>1.1.2</v2></Row> <Row><v1>1.2.1</v1><v2>1.2.2</v2></Row> </Rows>	[@rowID='1'] is een filter: Selecteer enkel de Rows nodes met het attribuut rowID gelijk aan 1
//Rows	Zelfde als eerste result	// Wil zeggen 'zoek eender waar in de hiërarchie naar Rows elementen (vaak als shortcut gebruikt)

XPATH

- querytaal voor XML
- *Zie voor meer info*

http://www.w3schools.com/xsl/xpath_syntax.asp

```
<?xml version="1.0" encoding="UTF-8"?>
<Collections testDescription="1 - simple functionality
test">
  <Rows rowID="1">
    <Row><v1>1.1.1</v1><v2>1.1.2</v2></Row>
    <Row><v1>1.2.1</v1><v2>1.2.2</v2></Row>
  </Rows>
  <Rows rowID="2">
    <Row><v1>2.1.1</v1><v2>2.1.2</v2></Row>
    <Row><v1>2.2.1</v1><v2>2.2.2</v2></Row>
  </Rows>
  <Rows rowID="3">
    <Row><v1>3.1.1</v1><v2>3.1.2</v2></Row>
    <Row><v1>3.2.1</v1><v2>3.2.2</v2></Row>
  </Rows>
</Collections>
```

XPathexpressie	Result	Uitleg
/Collections/Rows	Lijst met 3 Node-elements met de naam Rows (met rowID=1, 2 en 3)	Geef alle Rows elementen binnen Collections elementen. Als er meerdere rijen voldoen aan de query dan worden deze als lijst met elementen teruggegeven.
/Collections/Rows[@rowID='1']	1 Node element: <Rows rowID="1"> <Row><v1>1.1.1</v1><v2>1.1.2</v2></Row> <Row><v1>1.2.1</v1><v2>1.2.2</v2></Row> </Rows>	[@rowID='1'] is een filter: Selecteer enkel de Rows nodes met het attribuut rowID gelijk aan 1
//Rows	Zelfde als eerste result	// Wil zeggen 'zoek eender waar in de hiërarchie naar Rows elementen (vaak als shortcut gebruikt)

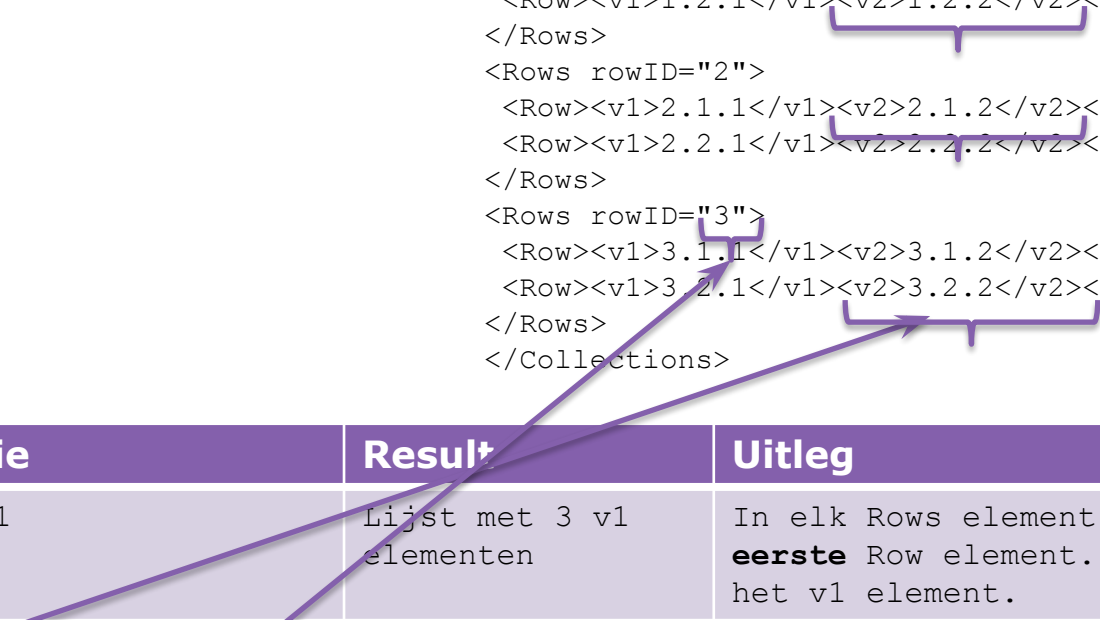
XPATH

```
<?xml version="1.0" encoding="UTF-8"?>
<Collections testDescription="1 - simple functionality
test">
  <Rows rowID="1">
    <Row><v1>1.1.1</v1><v2>1.1.2</v2></Row>
    <Row><v1>1.2.1</v1><v2>1.2.2</v2></Row>
  </Rows>
  <Rows rowID="2">
    <Row><v1>2.1.1</v1><v2>2.1.2</v2></Row>
    <Row><v1>2.2.1</v1><v2>2.2.2</v2></Row>
  </Rows>
  <Rows rowID="3">
    <Row><v1>3.1.1</v1><v2>3.1.2</v2></Row>
    <Row><v1>3.2.1</v1><v2>3.2.2</v2></Row>
  </Rows>
</Collections>
```

XPathexpressie	Result	Uitleg
<code>//Rows/Row[1]/v1</code>	Lijst met 3 v1 elementen	In elk Rows element selecteer je het eerste Row element. Daarin selecteer je het v1 element.
<code>//Rows[Row/v2='3.2.2']/@rowID</code> of <code>//Rows[./Row/v2='3.2.2']/@rowID</code>	Attribuutwaarde 3	Zoek alle Rows elementen waarin een Row/v2 element voorkomt met waarde '3.2.2' (het puntje kan gebruikt worden om het current element aan te duiden). Geef voor die Rows elementen de waarde van het attribuut rowID terug
<code>//Row[v2='3.2.2']/../@rowID</code>	Attribuutwaarde 3	Zoek Row elementen die een v2 node bevatten met waarde '3.2.2'. Voor die elementen zoek de attribuutwaarde rowID van de parent (voor Row is deze Rows.

XPATH

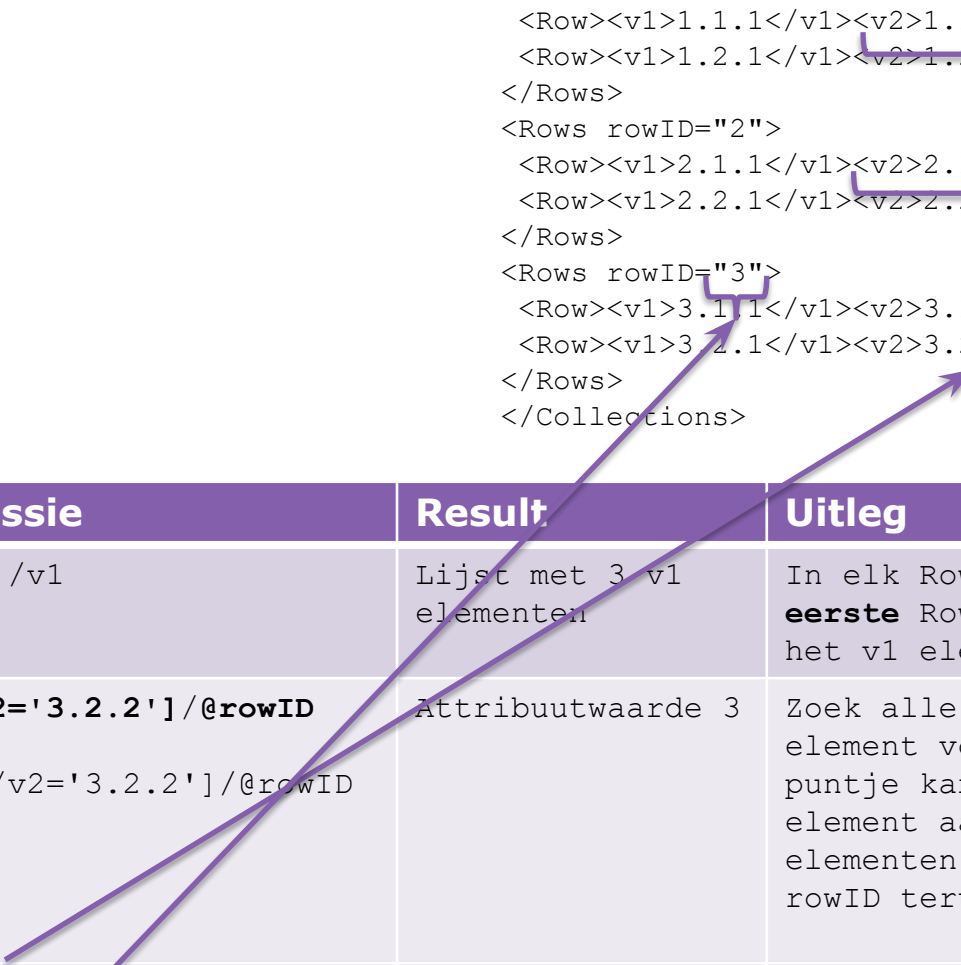
```
<?xml version="1.0" encoding="UTF-8"?>
<Collections testDescription="1 - simple functionality
test">
  <Rows rowID="1">
    <Row><v1>1.1.1</v1><v2>1.1.2</v2></Row>
    <Row><v1>1.2.1</v1><v2>1.2.2</v2></Row>
  </Rows>
  <Rows rowID="2">
    <Row><v1>2.1.1</v1><v2>2.1.2</v2></Row>
    <Row><v1>2.2.1</v1><v2>2.2.2</v2></Row>
  </Rows>
  <Rows rowID="3">
    <Row><v1>3.1.1</v1><v2>3.1.2</v2></Row>
    <Row><v1>3.2.1</v1><v2>3.2.2</v2></Row>
  </Rows>
</Collections>
```



XPathexpressie	Result	Uitleg
<code>//Rows/Row[1]/v1</code>	Lijst met 3 v1 elementen	In elk Rows element selecteer je het eerste Row element. Daarin selecteer je het v1 element.
<code>//Rows[Row/v2='3.2.2']/@rowID</code> of <code>//Rows[./Row/v2='3.2.2']/@rowID</code>	Attribuutwaarde 3	Zoek alle Rows elementen waarin een Row/v2 element voorkomt met waarde '3.2.2' (het puntje kan gebruikt worden om het current element aan te duiden). Geef voor die Rows elementen de waarde van het attribuut rowID terug
<code>//Row[v2='3.2.2']/../@rowID</code>	Attribuutwaarde 3	Zoek Row elementen die een v2 node bevatten met waarde '3.2.2'. Voor die elementen zoek de attribuutwaarde rowID van de parent (voor Row is deze Rows.

XPATH

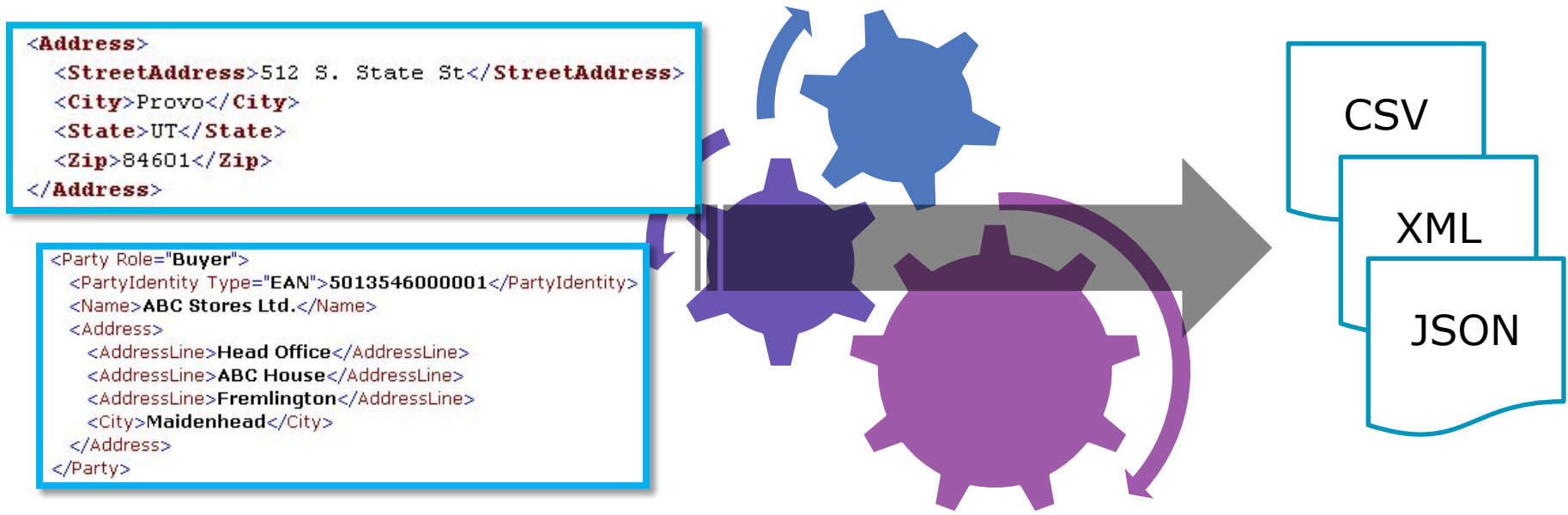
```
<?xml version="1.0" encoding="UTF-8"?>
<Collections testDescription="1 - simple functionality
test">
<Rows rowID="1">
  <Row><v1>1.1.1</v1><v2>1.1.2</v2></Row>
  <Row><v1>1.2.1</v1><v2>1.2.2</v2></Row>
</Rows>
<Rows rowID="2">
  <Row><v1>2.1.1</v1><v2>2.1.2</v2></Row>
  <Row><v1>2.2.1</v1><v2>2.2.2</v2></Row>
</Rows>
<Rows rowID="3">
  <Row><v1>3.1.1</v1><v2>3.1.2</v2></Row>
  <Row><v1>3.2.1</v1><v2>3.2.2</v2></Row>
</Rows>
</Collections>
```



XPathexpressie	Result	Uitleg
<code>//Rows/Row[1]/v1</code>	Lijst met 3 v1 elementen	In elk Rows element selecteer je het eerste Row element. Daarin selecteer je het v1 element.
<code>//Rows[Row/v2='3.2.2']/@rowID</code> of <code>//Rows[./Row/v2='3.2.2']/@rowID</code>	Attribuutwaarde 3	Zoek alle Rows elementen waarin een Row/v2 element voorkomt met waarde '3.2.2' (het puntje kan gebruikt worden om het current element aan te duiden). Geef voor die Rows elementen de waarde van het attribuut rowID terug
<code>//Row[v2='3.2.2']/../@rowID</code>	Attribuutwaarde 3	Zoek Row elementen die een v2 node bevatten met waarde '3.2.2'. Voor die elementen zoek de attribuutwaarde rowID van de parent (voor Row is deze Rows.

XSL

- Het komt vaak voor dat je gelijkaardige gegevens van verschillende bronnen binnen krijgt in verschillende XML formaten
- XSL(T): Extensible Stylesheet Language (for Transformations) is een XML gebaseerde taal waarmee je xml kan transformeren naar een ander formaat
 - *Oorspronkelijk ontwikkeld om XML met formattering weer te geven in de browser, maar die techniek is nooit echt doorgebroken*
 - *Bedrijven gebruiken XSL echter wel om XML uit verschillende bronnen om te zetten in andere formaten.*



XSL

- Open in notepad++ slides.xml en slides.xsl
- Je kan de code uitvoeren als volgt:
 - Selecteer de tab van slides.xml
 - Plugins > XML Tools > XSL Transformation
 - Selecteer het bestand slides.xsl
 - Voer uit en bekijk het aangemaakte bestand

Altijd dit rootelement gebruiken.
Anders geen valid xsl document.

Geeft aan dat de output een
plain tekst formaat is. Andere
mogelijkheden zijn html en xml

Dit is de xsl tegenhanger van
main() bij java. Hier start je xsl
transformatie

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet
  version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="text"/>
  <xsl:template match="/">
    <xsl:for-each select="/Collections/Rows/Row">
      <xsl:value-of select="normalize-space(../@rowID)"/>
      <xsl:text>;</xsl:text>
      <xsl:value-of select="normalize-space(v1)"/>
      <xsl:text>;</xsl:text>
      <xsl:value-of select="normalize-space(v2)"/>
      <xsl:text>&#xa;</xsl:text>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

For-each is een equivalent van een loop.
Voor elke node uit het resultaat van de XPATH query
wordt de code binnen het for-each element
uitgevoerd
Hier dus voor elke "Row" node.

XSL

../@rowID: Binnen de foreach zit je al op het niveau van je "Row" element

Bij de eerste iteratie zit je dus op /Collections/Rows/Row[1] niveau.

Als je van daaruit naar andere elementen wil navigeren doe je dat dus relatief.

In dit geval ga je naar de parent van het eerste Row element en zoek je daar naar @rowID

normalize-space() is een functie die de spaces en enters binnen een element negeert. Dat is vooral belangrijk als je in het xml bestand tabs en dergelijke hebt gebruikt om de xml leesbaar te maken.

Zie http://www.w3schools.com/xsl/xsl_functions.asp voor meer functies.

Value-of geeft de waarde van de expressie in select terug als tekst.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet
  version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="text"/>
  <xsl:template match="/">
    <xsl:for-each select="/Collections/Rows/Row">
      <xsl:value-of select="normalize-space(../@rowID)"/>
      <xsl:text>;</xsl:text>
      <xsl:value-of select="normalize-space(v1)"/>
      <xsl:text>;</xsl:text>
      <xsl:value-of select="normalize-space(v2)"/>
      <xsl:text>&#xa;</xsl:text>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

Dit is een new line

Xsl:text voegt tekst toe aan de output

Inhoudstafel

1. Data transformatie
2. ETL
3. XML, XSL en XPATH
4. **JSON**

JSON

- Wordt vaak gebruikt als communicatietaal bij services (vb. RESTCALLS)
- kan eenvoudig gebruikt worden binnen Javascript (zie ook http://www.w3schools.com/json/json_intro.asp)
 - Javascript kan hier dus als tegenhanger van XSL gebruikt worden

Een valid JSON object/file heeft een rootelement

Arrays hebben de vorm:
[element1,element2]

Een complex object heeft de vorm
{element1, element2}

Een eenvoudig object heeft de vorm
"Naam": waarde

```
{ "Collections": [
  { "ID": 4, "Rows": [
    { "V1": "4.1.1", "V2": "4.1.2" },
    { "V1": "4.2.1", "V2": "4.2.2" } ] },
  { "ID": 5, "Rows": [
    { "V1": "5.1.1", "V2": "5.1.2" },
    { "V1": "5.2.1", "V2": "5.2.2" } ] },
  { "ID": 6, "Rows": [
    { "V1": "6.1.1", "V2": "6.1.2" },
    { "V1": "6.2.1", "V2": "6.2.2" } ] }
] }
```

Thank you.