

XML Conversion Cheat Sheet

1. SQL Snippets for XML Conversion

A. SQL Server: FOR XML Clause

```
sql

-- Basic XML Output (Element-Centric)
SELECT * FROM Customers
FOR XML AUTO, ELEMENTS;

-- Customized XML (with Attributes)
SELECT
    CustomerID AS "@ID",
    FirstName,
    LastName
FROM Customers
FOR XML PATH('Customer'), ROOT('Customers');
```

B. Oracle: XMLGEN

```
sql

-- Using XMLELEMENT and XMLAGG
SELECT
    XMLELEMENT("Employee",
        XMLATTRIBUTES(employee_id AS "ID"),
        XMLELEMENT("Name", first_name || ' ' || last_name)
    ) AS xml_output
FROM employees;
```

C. MySQL Workaround

sql

```
-- Using CONCAT and GROUP_CONCAT
SELECT CONCAT(
    '<Employees>',
    GROUP_CONCAT(
        CONCAT('<Employee><ID>', id, '</ID><Name>', name, '</Name></Employee>')
        SEPARATOR ' '
    ),
    '</Employees>'
) AS xml_data FROM employees;
```

2. Python/PHP Code Templates

A. Python (xml.etree.ElementTree)

python

```
import sqlite3
import xml.etree.ElementTree as ET

# Database Query
conn = sqlite3.connect('company.db')
cursor = conn.cursor()
cursor.execute("SELECT id, name, salary FROM employees")

# XML Generation
root = ET.Element("Employees")
for row in cursor:
    emp = ET.SubElement(root, "Employee", attrib={"id": str(row[0])})
    ET.SubElement(emp, "Name").text = row[1]
    ET.SubElement(emp, "Salary").text = str(row[2])

# Save to File
ET.ElementTree(root).write("employees.xml", encoding="utf-8", xml_declaration=True)
```

B. PHP (SimpleXML + MySQLi)

php

```
<?php
$conn = new mysqli("localhost", "user", "password", "db");
$result = $conn->query("SELECT id, name, department FROM staff");

$xml = new SimpleXMLElement('<Organization/>');
while ($row = $result->fetch_assoc()) {
    $staff = $xml->addChild('Staff');
    $staff->addAttribute('emp_id', $row['id']);
    $staff->addChild('Name', $row['name']);
    $staff->addChild('Department', $row['department']);
}

$xml->asXML('organization_data.xml');
?>
```

3. Common XSD Schema Examples

A. Basic Employee Schema

xml

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Employees">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Employee" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ID" type="xs:integer"/>
              <xs:element name="Name" type="xs:string"/>
              <xs:element name="JoinDate" type="xs:date"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

B. Schema with Attributes

xml

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Products">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Product" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attribute name="sku" type="xs:string" use="required"/>
            <xs:attribute name="price" type="xs:decimal"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

4. Pro Tips for Validation

Validate XML against XSD in Python:

python

```
from lxml import etree
schema = etree.XMLSchema(file="schema.xsd")
xml = etree.parse("data.xml")
schema.assertValid(xml) # Raises error if invalid
```

Online Validation Tools:

1. [XML Validation \(W3Schools\)](#)
2. [XSD Validator \(FreeFormatter\)](#)

"Liked this cheat sheet? Read the [full guide](#) on our blog!"