Practical 6 – XML

1. Create the following table with xml type.

   -- Example: Untyped XML Column in Table

```
CREATE TABLE AdminDocs (
id int  primary key,
xDoc Xml not null
)
```

   -- Example: Inserting Data into Untyped XML Column

```
INSERT INTO AdminDocs VALUES (1,
'<catalog>
 <product dept="WMN">
  <number>557</number>
  <name language="en">Fleece Pullover</name>
  <colorChoices>navy black</colorChoices>
 </product>
 <product dept="ACC">
  <number>563</number>
  <name language="en">Floppy Sun Hat</name>
 </product>
 <product dept="ACC">
  <number>443</number>
  <name language="en">Deluxe Travel Bag</name>
 </product>
 <product dept="MEN">
  <number>784</number>
  <name language="en">Cotton Dress Shirt</name>
  <colorChoices>white gray</colorChoices>
  <desc>Our <i>favorite</i> shirt!</desc>
 </product>
</catalog>')

INSERT INTO AdminDocs VALUES (2,
'<doc id="123">
  <sections>
   <section num="1"><title>XML Schema</title></section>
```

```
  <section num="3"><title>Benefits</title></section>
  <section num="4"><title>Features</title></section>
   </sections>
</doc>')

select *
from AdminDocs
```

2. Practice the following XPath expressions

    -- Example: Using Query() Method

    ```
    SELECT id, xDoc.query('/catalog/product')
    FROM   AdminDocs

    SELECT id, xDoc.query('//product')
    FROM   AdminDocs

    SELECT id, xDoc.query('/*/product')
    FROM   AdminDocs

    SELECT id, xDoc.query('/*/product[@dept="WMN"]')
    FROM   AdminDocs

    SELECT id, xDoc.query('/*/child::product[attribute::dept="WMN"]')
    FROM   AdminDocs

    SELECT id, xDoc.query('//product[dept="WMN"]')
    FROM   AdminDocs

    SELECT id, xDoc.query('descendant-or-self::product[attribute::dept="WMN"]')
    FROM   AdminDocs

    SELECT id, xDoc.query('//product[number > 500]')
    FROM   AdminDocs
    where id=1

    SELECT id, xDoc.query('//product/number[. gt 500]')
    FROM   AdminDocs
    where id=1

    SELECT id, xDoc.query('/catalog/product[4]')
    FROM   AdminDocs
    where id=1
    ```

```
SELECT id, xDoc.query('//product[number > 500][@dept="ACC"]')
FROM   AdminDocs
where id=1

SELECT id, xDoc.query('//product[number > 500][1]')
FROM   AdminDocs
where id=1
```

3. Practice the following XQuery expressions.

```
SELECT xDoc.query('   for $prod in //product
                let $x:=$prod/number
                return $x')
FROM   AdminDocs
where id=1

SELECT xDoc.query('   for $prod in //product
                let $x:=$prod/number
                where $x>500
                return $x')
FROM   AdminDocs
where id=1

SELECT xDoc.query('   for $prod in //product
                let $x:=$prod/number
                return $x')
FROM   AdminDocs
where id=1

SELECT xDoc.query('   for $prod in //product
                let $x:=$prod/number
                where $x>500
                return (<Item>{$x}</Item>)')
FROM   AdminDocs
where  id=1

SELECT xDoc.query('   for $prod in //product[number > 500]
                let $x:=$prod/number
                return (<Item>{$x}</Item>)')
FROM   AdminDocs
where  id=1

SELECT xDoc.query('   for $prod in //product
```

```
              let $x:=$prod/number
              where $x>500
              return (<Item>{data($x)}</Item>)')
FROM   AdminDocs
where  id=1

SELECT xDoc.query('   for $prod in //product
              let $x:=$prod/number
              return  if ($x>500)
                  then <book>{data($x)}</book>
                  else <paper>{data($x)}</paper>')
FROM   AdminDocs
where  id=1
```

4. Practice the following XML DML XQuery expressions.

    --Example: Insertion of Subtree into XML Instances

    select * from AdminDocs where id=2

```
UPDATE AdminDocs SET xDoc.modify('
 insert
  <section num="2">
     <title>Background</title>
  </section>
 after (/doc//section[@num=1])[1]')

UPDATE AdminDocs SET xDoc.modify('
 delete
  //section[@num="2"]')
```