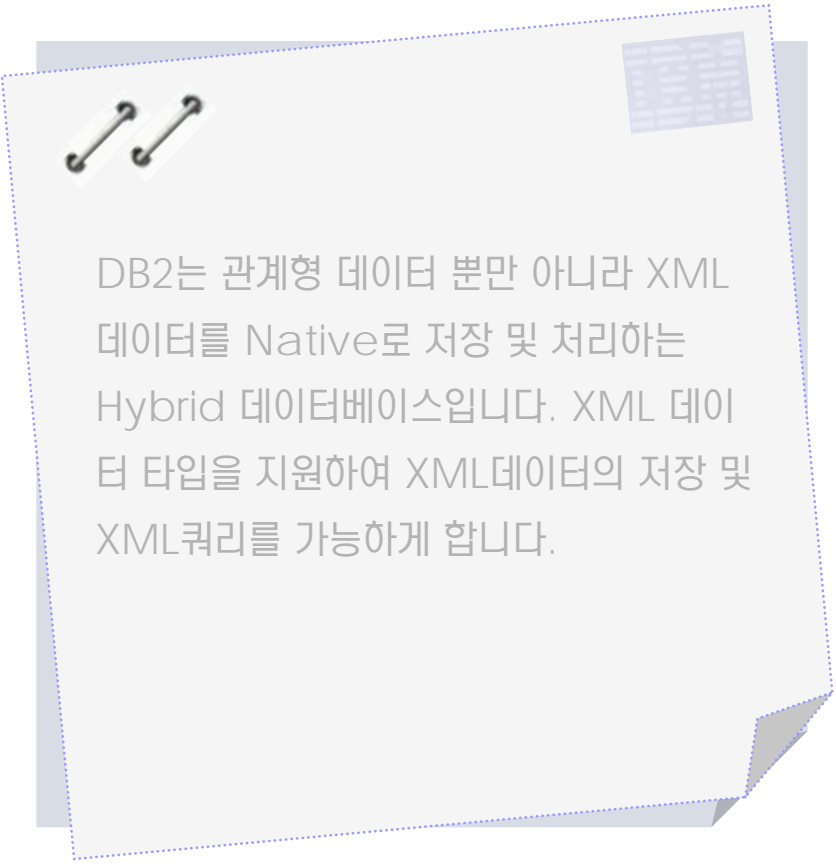




UNIT 21

XML 데이터 관리



DB2는 관계형 데이터 뿐만 아니라 XML 데이터를 Native로 저장 및 처리하는 Hybrid 데이터베이스입니다. XML 데이터 타입을 지원하여 XML데이터의 저장 및 XML쿼리를 가능하게 합니다.

DB2 9.7 운영자 가이드

Administrator Edition

- XML 구조 및 데이터베이스
- XML 테이블 및 인덱스 생성
- XML 쿼리문
- XML 쿼리문-XPath
- XML 쿼리문-Xquery
- XML 쿼리문-SQL/XML



Point



DB2는 XML데이터를 Native로 저장하는 Pure XML을 지원함으로써, XML데이터의 관리 및 성능상 매우 유리한 기능을 제공하고 있습니다.

Tip

Pure XML의 장점

- 1) XML 함수의 활용으로 프로그램 코드의 복잡성을 줄입니다.
- 2) 스키마 변경작업등 XML 문서 관리 작업이 쉽습니다.
- 3) XML 애플리케이션에서 많은 성능 향상을 얻을 수 있습니다.

1 9.1이전 XML 데이터가 데이터베이스에 저장하는 방식입니다.

-A type: XML 문서가 데이터베이스의 clob 또는 varchar 컬럼에 저장됩니다.

-B type: XML문서가 테이블에 단편화 부분으로 나뉘어서 저장됩니다.

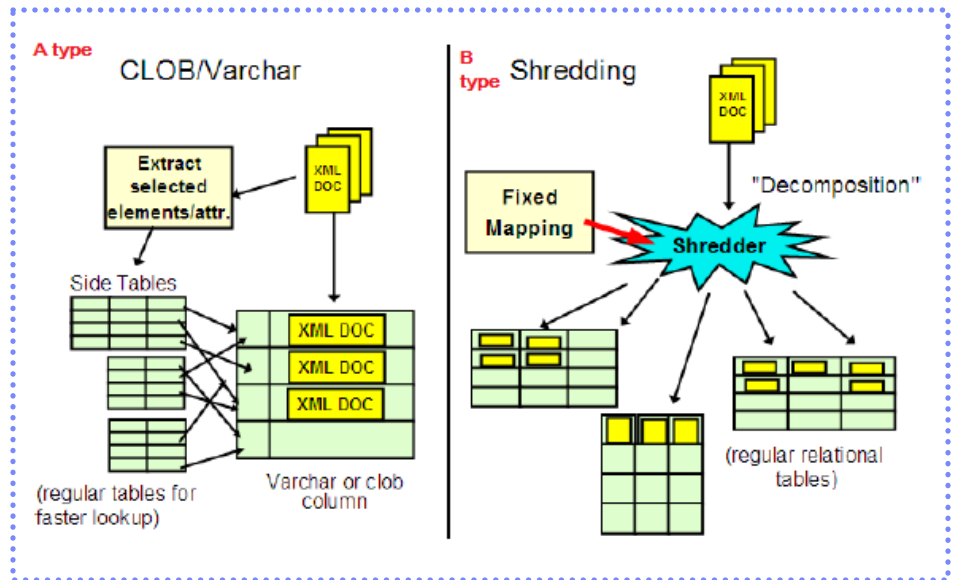


Figure 2101A... 9.1 이전 XML 문서 저장 방식

2 9.1 이후, 하나의 테이블은 관계형 데이터와 계층적 데이터로 나누어서 저장됩니다.

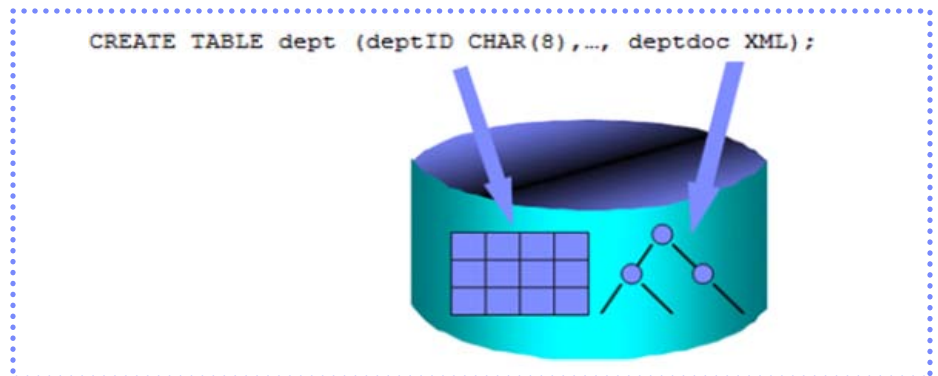


Figure 2101B... 9.1 이후 XML 컬럼이 있는 테이블 구조

Tip

Sample DB를 이용하여 XML 데이터를 생성할 때는 db2sample -sql -xml 옵션을 사용합니다.

Point



XML 데이터는 XML 데이터타입으로 저장된 후 일반 테이블과 같이 인덱스를 생성할 수 있습니다.

1 테이블 생성을 위한 샘플 XML 문장입니다.

```
<customerinfo Cid="1004">
<name>Matt Foreman</name>
<addr country="Canada">
<street>1596 Baseline</street>
<city>Toronto</city>
<state>Ontario</state>
<pcode>M3Z-5H9</pcode>
</addr>
<phone type="work">905-555-4789</phone>
<phone type="home">416-555-3376</phone>
<assistant>
<name>Peter Smith</name>
<phone type="home">416-555-3426</phone>
</assistant>
</customerinfo>
```

2 테이블 생성 문입니다.

```
CREATE TABLE customer(info XML)
```

2 인덱스 생성 문입니다. XML 문의 element, attribute, 그리고 text node에 대해서 인덱스를 생성할 수 있습니다.

-- Attribute Cid에 인덱스 생성

```
CREATE UNIQUE INDEX idx1 ON customer(info)
GENERATE KEY USING
xmlpattern '/customerinfo/@Cid'
AS sql DOUBLE
```

-- Element name에 인덱스 생성

```
CREATE INDEX idx2 ON customer(info)
GENERATE KEY USING
xmlpattern '/customerinfo/name'
AS sql VARCHAR(40)
```

-- 모든 elements name에 인덱스 생성

```
CREATE INDEX idx3 ON customer(info)
GENERATE KEY USING
xmlpattern '//name'
AS sql VARCHAR(40);
```

Tip

XML 인덱스는 단일 컬럼에서만 적용됩니다. 결합인덱스는 사용할 수 없습니다.

Tip

XML 인덱스가 가능한 타입은 DOUBLE, VARCHAR(n), VARCHAR HASHED, DATE, TIMESTAMP이 있습니다.

Tip

모든 텍스트 노드값에 인덱스를 생성하는 것은 인덱스 사이즈가 필요이상으로 커지게 되므로 권장하지 않습니다.

Point



XML문서를 파싱된 계층형 포맷으로 이미 저장하여 쿼리 수행 시 XML 파싱이 필요 없이, Xquery,Xpath,SQL/XML를 이용할 수 있습니다.

- 1 XML 문의 트리 구조에서 document, element, attribute, text, comment 노드별 구조입니다.

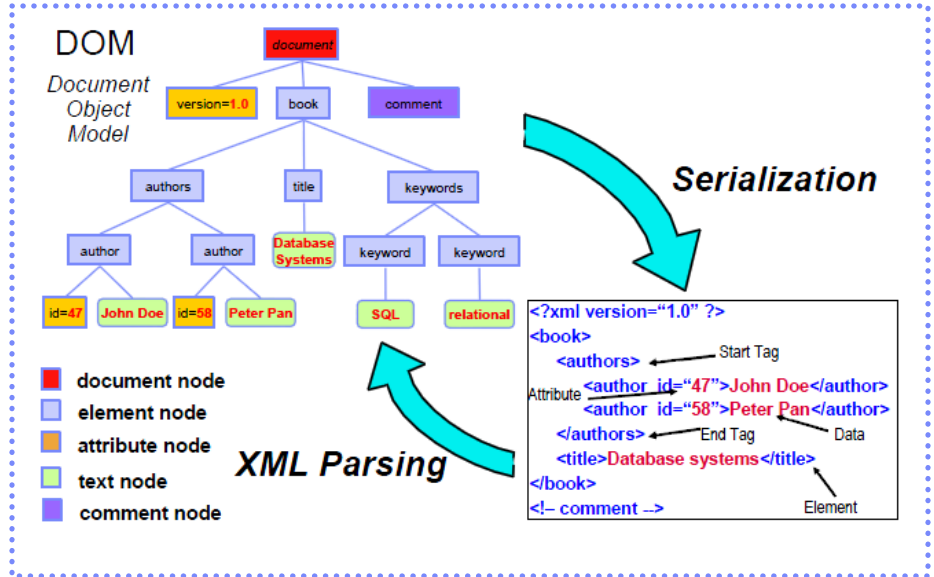


Figure 2103A... DB2의 XML 문 트리 구조

- 2 SQL/Xml문과 XQuery는 각각 ISO 및 W3C에서 인증하는 언어로써, 사용법이 다릅니다.

- SQL : 관계형 데이터에 접근하기 위한 표준 언어입니다.
- Xquery : XML 쿼리언어로 W3C에서 인증하는 언어입니다.
- SQL/XML : SQL과 XML를 같이 다루는 ISO에서 인증하는 표준 언어입니다.

- 3 SQL/Xml문에서 사용하는 주요 함수입니다.

함수 이름	설명
XMLPARSE	XML 값을 만들기 위해 character 형이나 오브젝트 바이너리 데이터 형식으로 파싱합니다.
XMLSERIALIZE	Character 또는 Large 오브젝트 형식으로 XML value 값을 변환합니다.
XMLVALIDATE	디폴트값을 포함하여 XML 스키마 유효성 확인에서 확보한 정보에 의해 증가된 입력 XML 값의 사본을 리턴합니다.
XMLEXISTS	업무가 수행되는 동안 사용될 시스템 자원의 허용 범위 지정. 허용된 범위를 초과할 경우 지정된 문장이 실행됩니다.
XMLQUERY	연결속성이 아닌 실행 유형(read, write, call등)에 따라 임계값 설정 및 특정 서비스 클래스로 매핑합니다.

Tip

- Before 트리거 문으로 문서 validation을 확인할 수 있습니다.

Point



XPath는 XML문서를 조회하는데 사용하는 언어입니다. XML의 attribute와 element를 상세하게 표현할 수 있습니다.

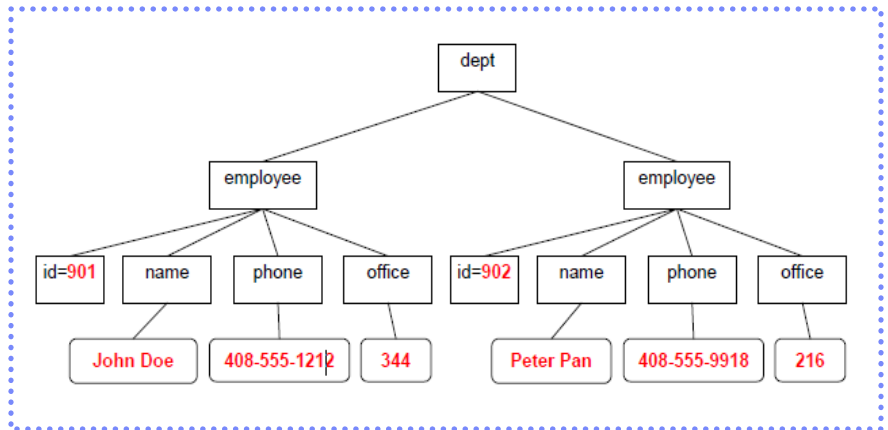
1 XPath를 사용하기 위해 XML 문서 예제를 살펴보겠습니다.

```
<dept bldg="101">
  <employee id="901">
    <name>John Doe</name>
    <phone>408 555 1212</phone>
    <office>344</office>
  </employee>
  <employee id="902">
    <name>Peter Pan</name>
    <phone>408 555 9918</phone>
    <office>216</office>
  </employee>
</dept>
```

Tip

응용 프로그래밍은 반드시 대소문자를 구분하여 사용하며 사용자 명은 반드시 대문자를 사용할 것을 권장합니다.


2 위의 문서를 구조적으로 표현하면 아래와 같습니다.



3 XPath를 사용하여 아래와 같이 결과값을 얻을 수 있습니다.

XPath	결과
/dept/@bldg	101
/dept/employee/@id	901 902
/dept/employee/name	<name>Peter Pan</name> <name>John Doe</name>
/dept/employee/name/text()	Peter Pan John Doe

Point

 XPath에서 조건문 사용 방법입니다.

Tip

Xpath와 Xquery는 대소문자를 구분합니다.

- 4 XPath에서 “[]”를 사용하면 SQL문에서 where 구문을 사용하는 것처럼 조건문으로 사용할 수 있습니다.

XPath	결과
/dept/employee[@id="902"]/name	<name>Peter Pan</name>
/dept[@bldg="101"]/employee[office > "300"]/name	<name>John Doe</name>
//employee[office="344" OR office="216"]/@id	901 902
/dept/employee[2]/@id	902

- 5 XPath에서 “.” 은 현재 위치를, “..”는 상위 정보를 알려주는 데 사용합니다.

XPath	결과
/dept/employee/name[../@id="902"]	<name>Peter Pan</name>
/dept/employee/office[.>"300"]	<office>344</office>
/dept/employee[office > "300"]/office	101
/dept/employee[name="John Doe"]/../@bldg	101

Point



Xquery는 XML를 사용하기 위한 쿼리문 입니다. SQL문의 select - from- where 표현식처럼 Xquery에서는 FLWOR 표현식을 사용합니다.

1 FLOWR 표현식은 아래와 같습니다.

XPath	설명
FOR:	시퀀스를 반복하여 변수를 아이템에 바인딩
LET:	하나의 변수를 시퀀스에 바인딩
WHERE:	반복 아이템에 대한 제한 조건
ORDER:	반복 아이템을 재정렬
RETURN:	쿼리 결과값을 구성

2 Xquery 사용 예제입니다.

조회문)

```
xquery
for $d in db2-fn:xmlcolumn('dept.deptdoc')/dept
let $emp := $d//employee/name
where $d/@bldg > 95
order by $d/@bldg
return
<EmpList>
{$d/@bldg, $emp}
</EmpList>
```

조회결과)

```
<EmpList bldg="101">
<name>
John Doe
</name>
<name>
Peter Pan
</name>
</EmpList>
```


Point



SQL/XML 함수를 사용하여 XML 데이터를 조회하는 방법입니다.

1 Xmlexists 사용 예입니다.

-- xmlexists 사용 예제

```
SELECT name FROM clients
WHERE xmlexists(
'$c/Client/Address[zip="95116"]'
passing clients.contact as "c"
)
```

2 Xmlquery 사용 예입니다.

```
SELECT xmlquery('$c/Client/email' passing contact as "c")
FROM clients
WHERE status = 'Gold'
```

3 Xmltable 사용 예입니다.

```
SELECT t.comment#, i.itemname, t.customerID, Message
FROM items i,
xmltable('$c/Comments/Comment' passing i.comments as "c"
columns Comment# integer path 'CommentID',
CustomerID integer path 'CustomerID',
Message varchar(100) path 'Message') AS t
```

Tip

찾은 쿼리문은 뷰를 작성하여 사용하시면 편리합니다.

4 Clients 테이블의 contract xml 컬럼의 데이터 조회

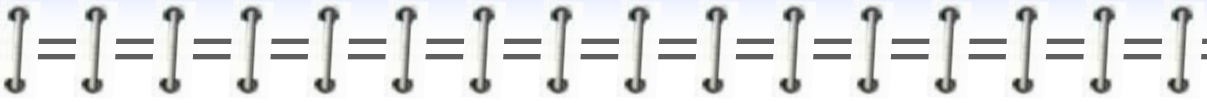
```
xquery db2-fn:xmlcolumn('CLIENTS.CONTRACT')
→ 위 문장의 경우 "SELECT contact FROM clients" 와 동일합니다.
```

5 클라이언트 팩스 데이터 조회

```
xquery
for $y in db2-fn:xmlcolumn('CLIENTS.CONTRACT')/Client/fax
return $y
```

6 두 테이블(dept, unit)에 대해서 조인문 쿼리

```
SELECT u.unitID
FROM dept d, unit u
WHERE XMLEXISTS (
'$e//employee[name = $m]'
passing d.deptdoc as "e", u.manager as "m")
```



Memo ▶

