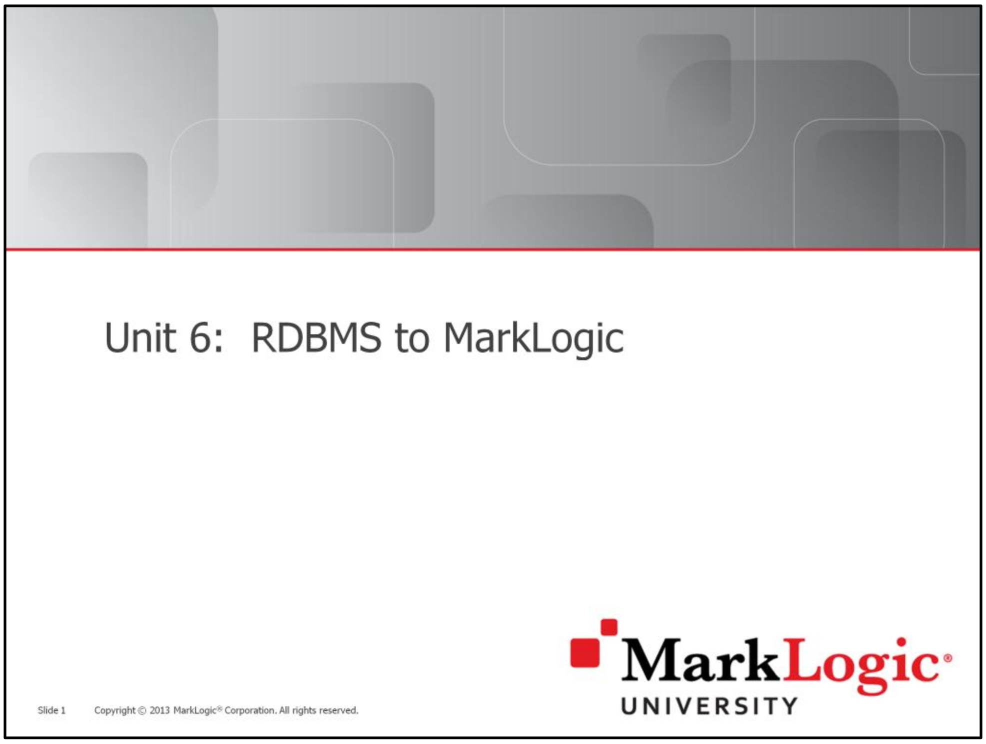
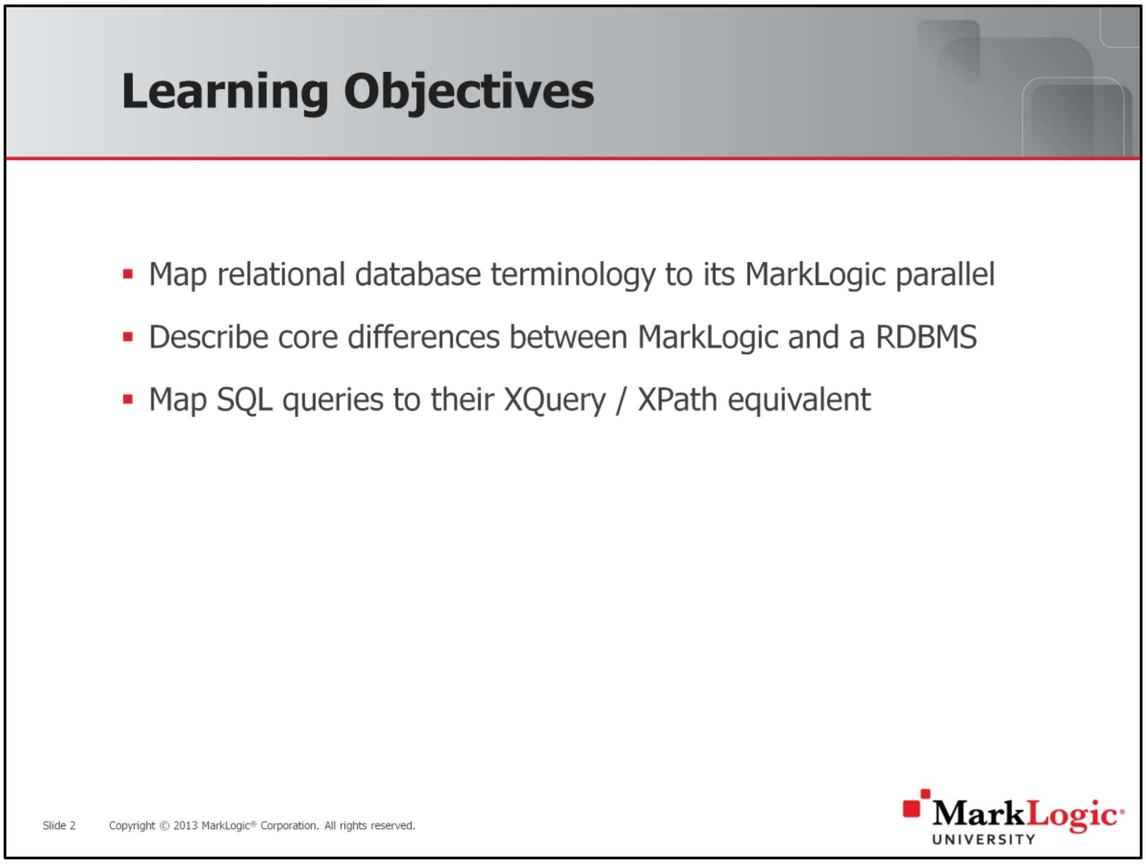


Unit 6: RDBMS to MarkLogic

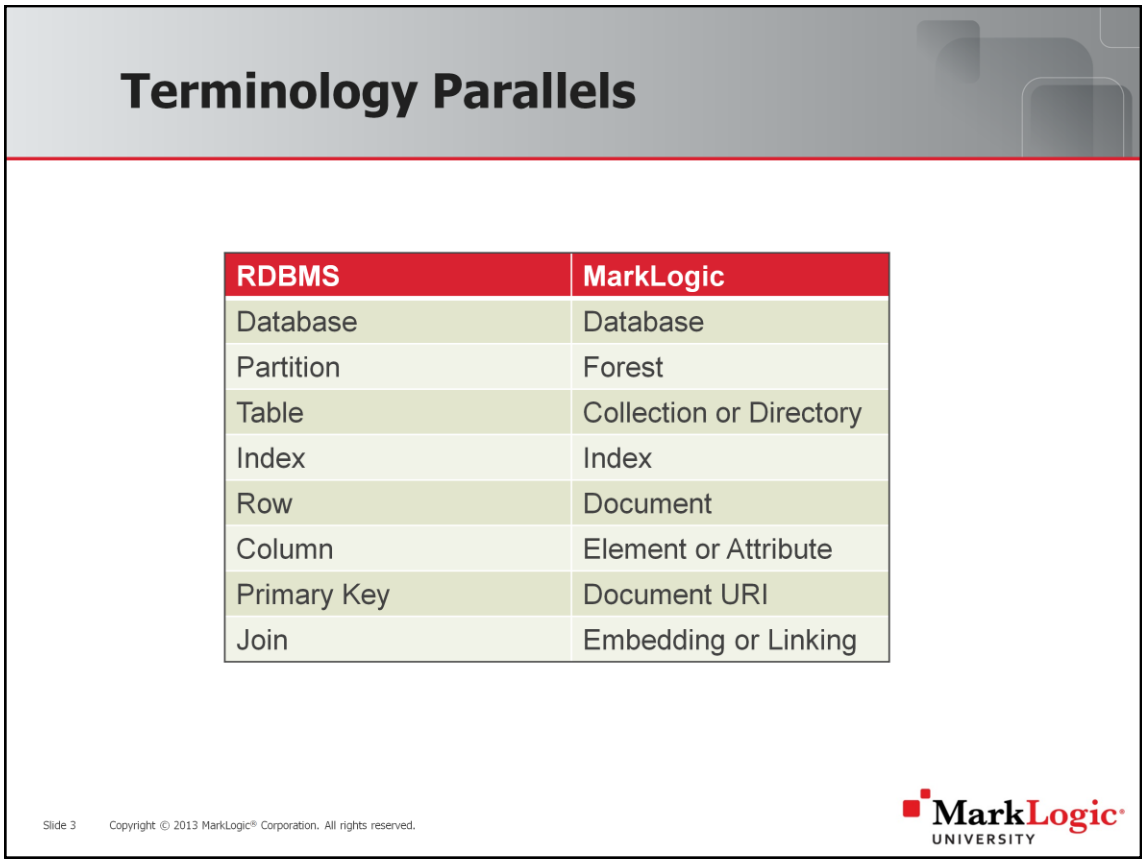
Slide 1 Copyright © 2013 MarkLogic® Corporation. All rights reserved.



Learning Objectives

- Map relational database terminology to its MarkLogic parallel
- Describe core differences between MarkLogic and a RDBMS
- Map SQL queries to their XQuery / XPath equivalent

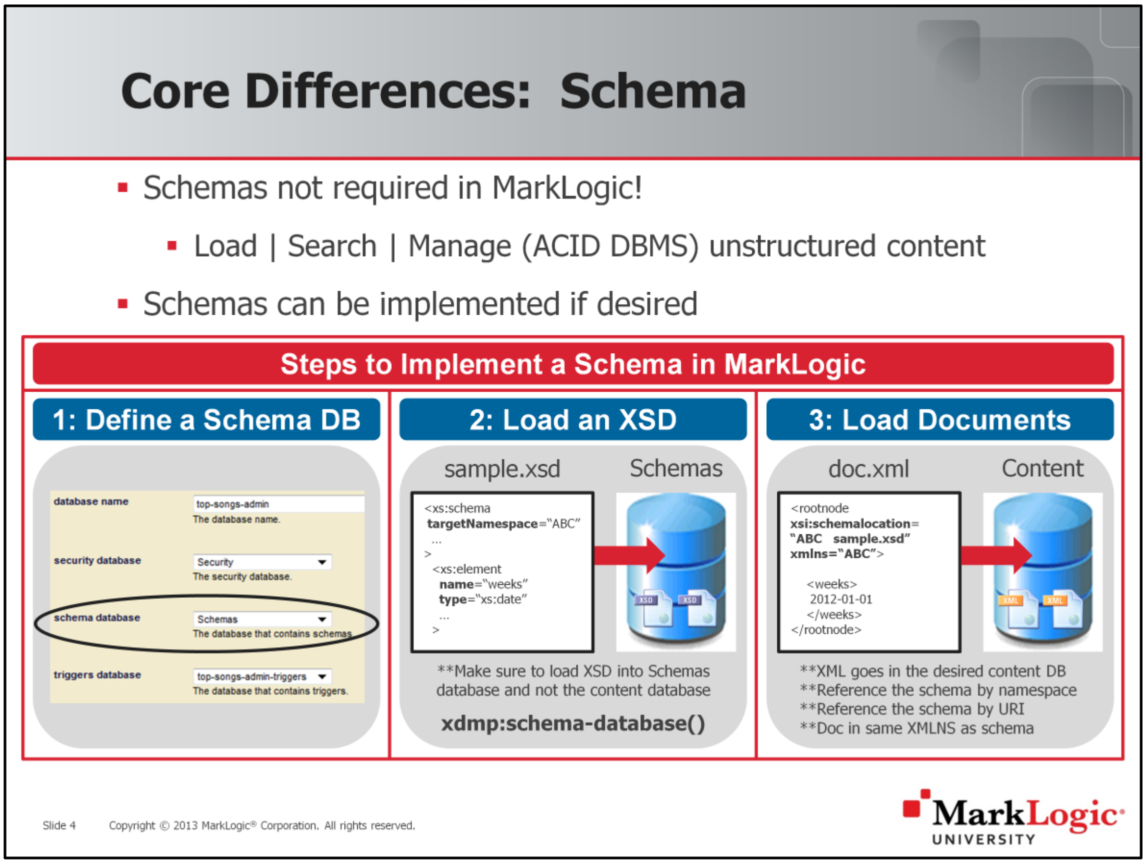
Slide 2 Copyright © 2013 MarkLogic® Corporation. All rights reserved.



Terminology Parallels

RDBMS	MarkLogic
Database	Database
Partition	Forest
Table	Collection or Directory
Index	Index
Row	Document
Column	Element or Attribute
Primary Key	Document URI
Join	Embedding or Linking

Slide 3 Copyright © 2013 MarkLogic® Corporation. All rights reserved.



Core Differences: Schema

- Schemas not required in MarkLogic!
  - Load | Search | Manage (ACID DBMS) unstructured content
- Schemas can be implemented if desired

Steps to Implement a Schema in MarkLogic

1: Define a Schema DB

database name	top-songs-admin The database name.
security database	Security The security database.
schema database	Schemas The database that contains schemas.
triggers database	top-songs-admin-triggers The database that contains triggers.

2: Load an XSD

sample.xsd → Schemas

```
<xs:schema
 targetNamespace="ABC"
  ...
>
 <xs:element
  name="weeks"
  type="xs:date"
  ...
 >
```

**Make sure to load XSD into Schemas database and not the content database

xdmp:schema-database()

3: Load Documents

doc.xml → Content

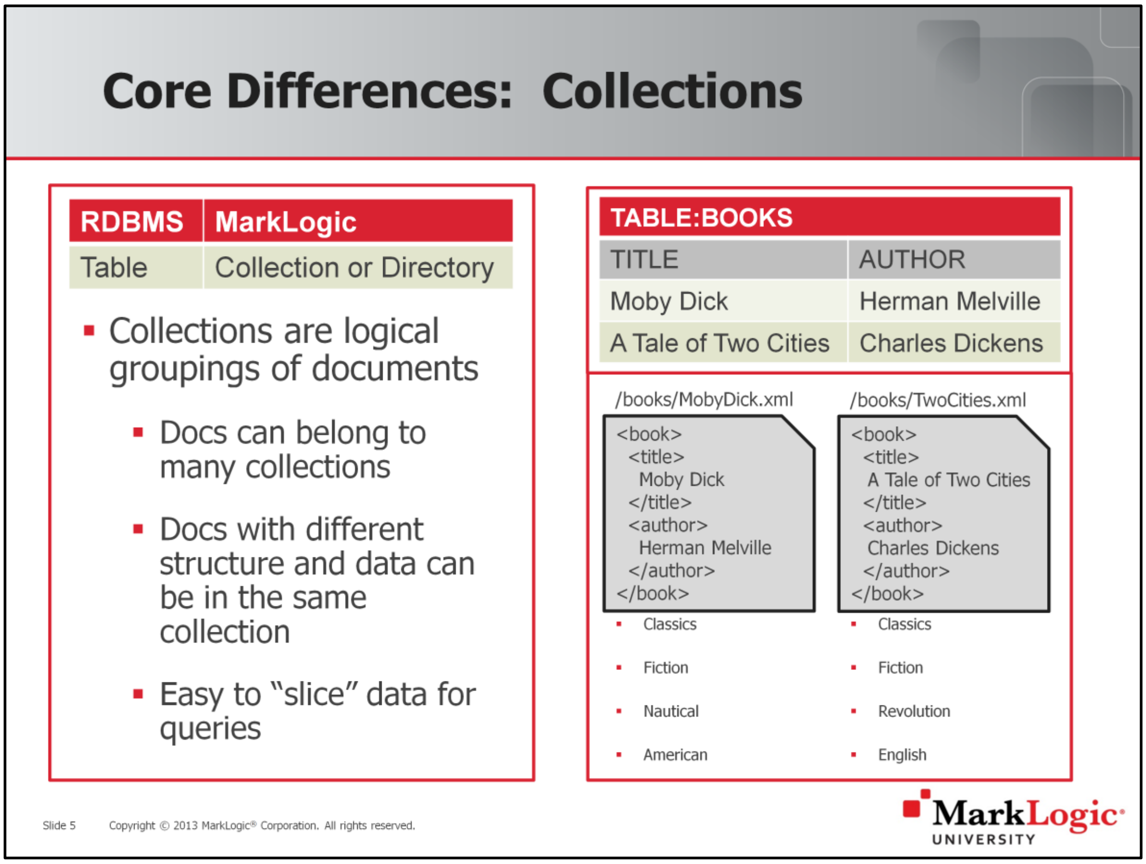
```
<rootnode
xsi:schemalocation=
"ABC   sample.xsd"
xmlns="ABC">

   <weeks>
    2012-01-01
   </weeks>
</rootnode>
```

**XML goes in the desired content DB
**Reference the schema by namespace
**Reference the schema by URI
**Doc in same XMLNS as schema

Slide 4

One of the benefits of MarkLogic is that it is built for today's unstructured content and therefore doesn't require a huge investment in upfront data modeling. Simply load your content and you can still take advantage of MarkLogic's robust search capability and all the benefits of a transactional DBMS.

However, should you want to exert more control over your data, MarkLogic also supports schemas.



Core Differences: Collections

RDBMS	MarkLogic
Table	Collection or Directory

- Collections are logical groupings of documents
  - Docs can belong to many collections
  - Docs with different structure and data can be in the same collection
  - Easy to "slice" data for queries

TABLE:BOOKS	
TITLE	AUTHOR
Moby Dick	Herman Melville
A Tale of Two Cities	Charles Dickens

/books/MobyDick.xml

```
<book>
  <title>
   Moby Dick
  </title>
  <author>
   Herman Melville
  </author>
</book>
```

- Classics
- Fiction
- Nautical
- American

/books/TwoCities.xml

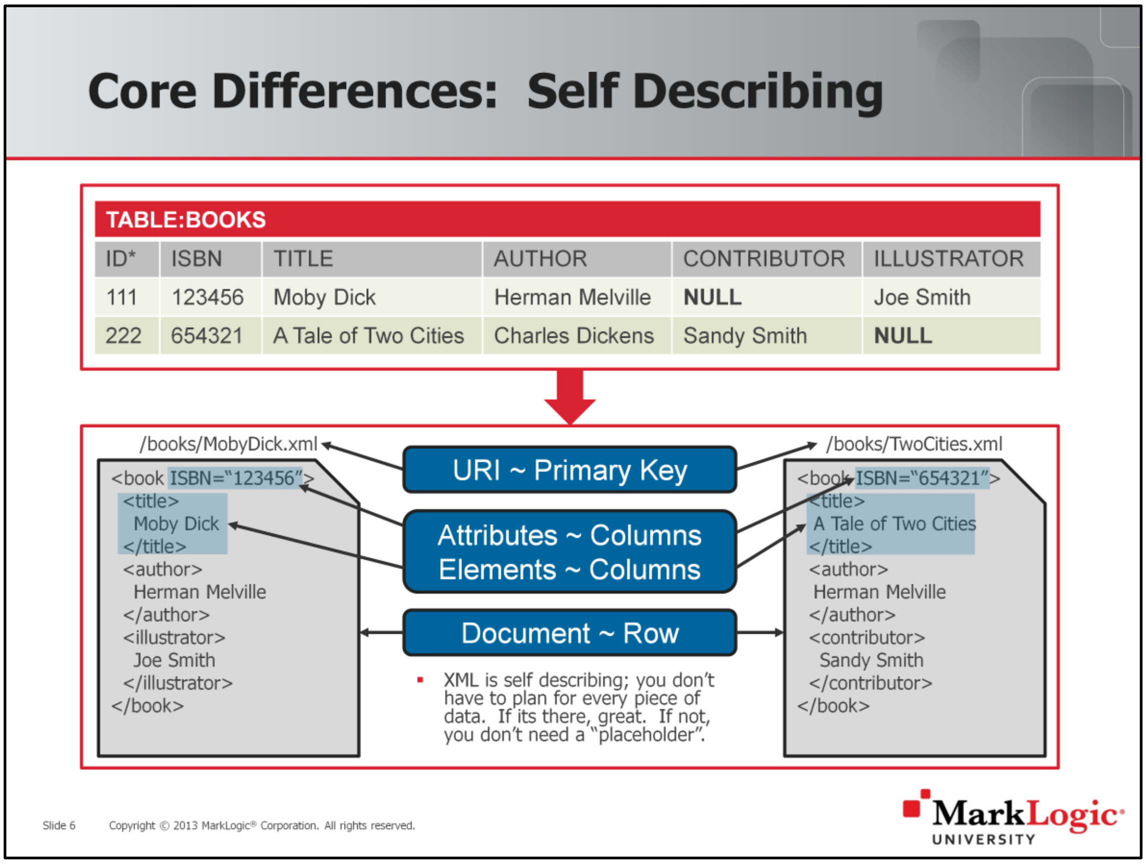
```
<book>
  <title>
   A Tale of Two Cities
  </title>
  <author>
   Charles Dickens
  </author>
</book>
```

- Classics
- Fiction
- Revolution
- English

Slide 5 Copyright © 2013 MarkLogic® Corporation. All rights reserved.

One can think of a Table in a RDBMS as the equivalent of a Collection in MarkLogic. However, the implementation of collections does differ significantly when you dig deeper.

Collections allow us to organize documents within a database. A single documents can belong to many collections. A many documents can be in the same collection. One key difference in the table / collection analogy is that when a collection contains multiple documents, those documents do not have to contain the same structure or data types. Collections are applied to logically group your data – and are very valuable when it comes to querying your database in "slices".



Core Differences: Self Describing

TABLE:BOOKS					
ID*	ISBN	TITLE	AUTHOR	CONTRIBUTOR	ILLUSTRATOR
111	123456	Moby Dick	Herman Melville	**NULL**	Joe Smith
222	654321	A Tale of Two Cities	Charles Dickens	Sandy Smith	**NULL**

/books/MobyDick.xml

```
<book ISBN="123456">
  <title>
    Moby Dick
  </title>
  <author>
    Herman Melville
  </author>
  <illustrator>
    Joe Smith
  </illustrator>
</book>
```

/books/TwoCities.xml

```
<book ISBN="654321">
  <title>
   A Tale of Two Cities
  </title>
  <author>
   Herman Melville
  </author>
  <contributor>
    Sandy Smith
  </contributor>
</book>
```

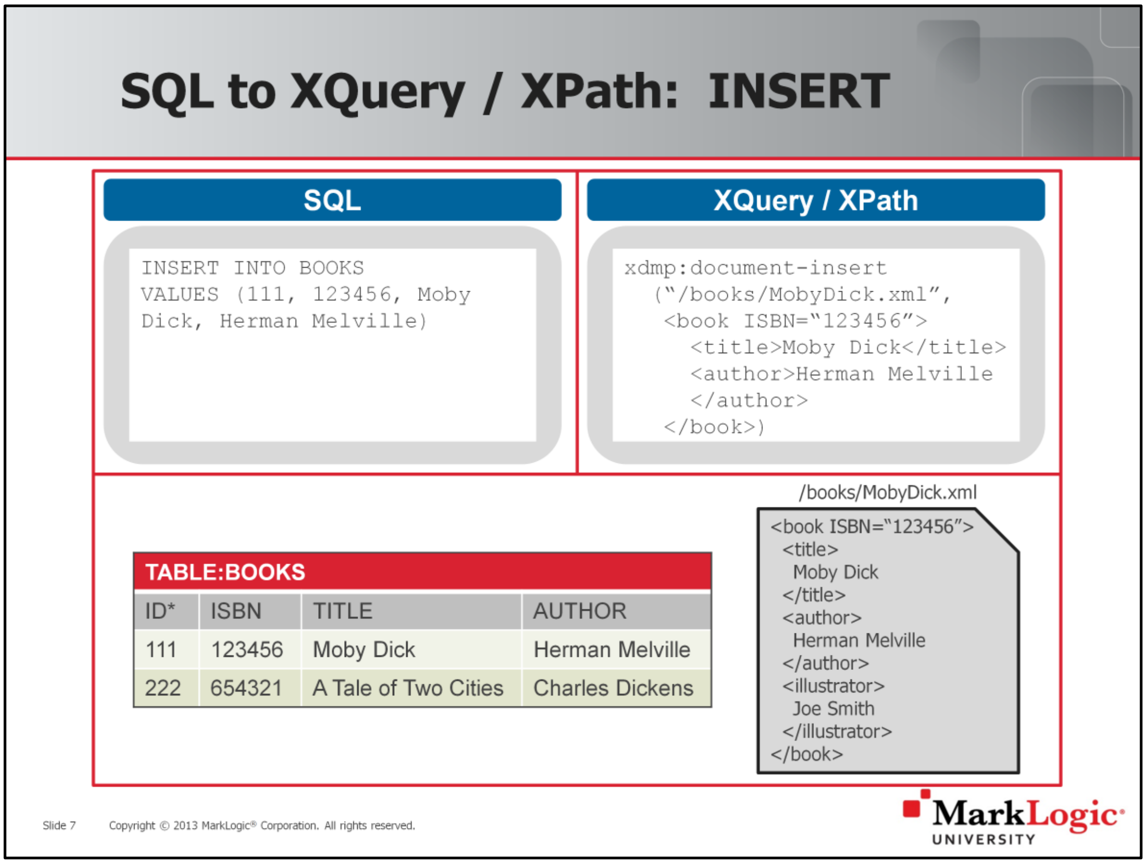
- XML is self describing; you don't have to plan for every piece of data. If its there, great. If not, you don't need a "placeholder".

Slide 6 Copyright © 2013 MarkLogic® Corporation. All rights reserved.

XML can be thought of as self describing. The structure helps to define and describe the data. But if documents vary from each other, there is no requirement to represent all the data the same between those documents.

Elements in XML are recursive, and you can have mixed content (elements mixed with text)—just like HTML. Which makes XML good for not only representing tabular data but also document-oriented data. To represent this using an RDBMS is highly impractical, entailing a ridiculous degree of "shredding" into many tables to represent what can easily be represented in one XML document.

Each document is given a URI. This is the documents unique identifier and is used in updates / deletes / etc to identify a specific document. It is like the primary key concept for a row in a relational table.



SQL to XQuery / XPath: INSERT

SQL

```
INSERT INTO BOOKS
VALUES (111, 123456, Moby
Dick, Herman Melville)
```

XQuery / XPath

```
xdmp:document-insert
  ("/books/MobyDick.xml",
   <book ISBN="123456">
     <title>Moby Dick</title>
     <author>Herman Melville
     </author>
   </book>)
```

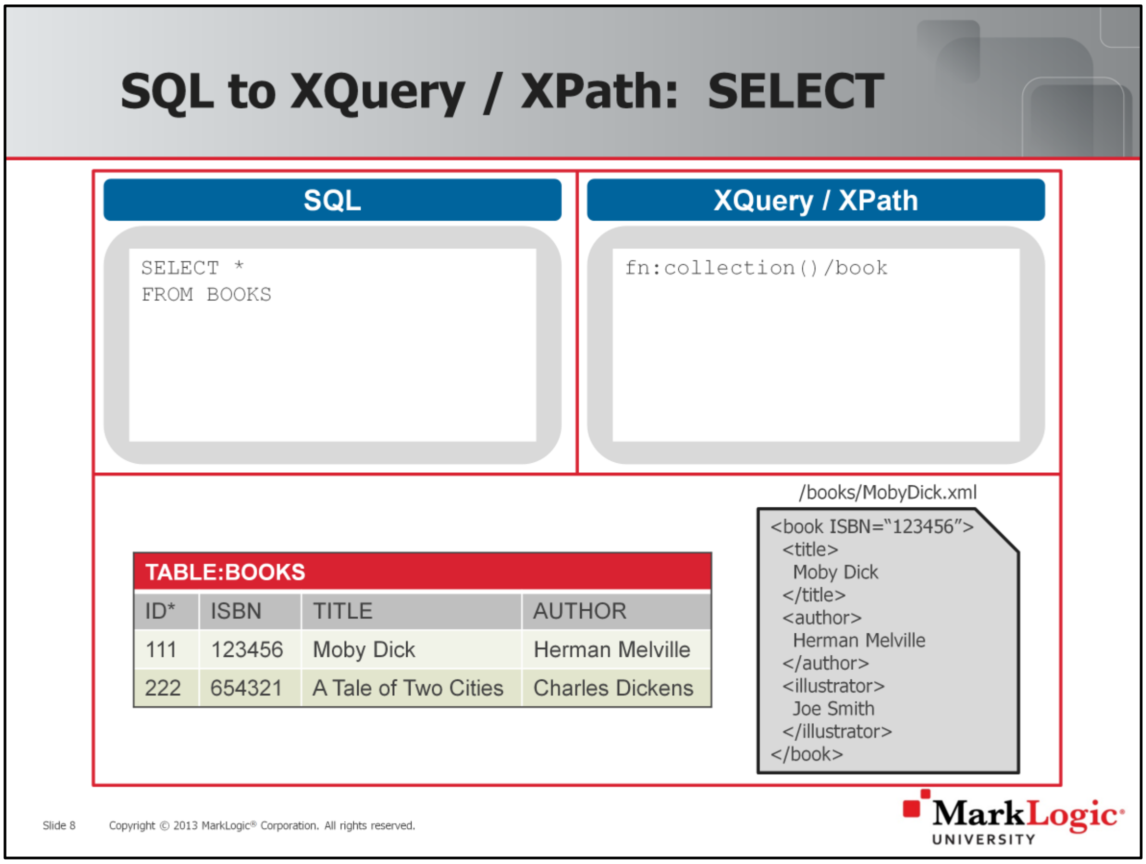
TABLE:BOOKS

ID*	ISBN	TITLE	AUTHOR
111	123456	Moby Dick	Herman Melville
222	654321	A Tale of Two Cities	Charles Dickens

/books/MobyDick.xml

```
<book ISBN="123456">
  <title>
    Moby Dick
  </title>
  <author>
    Herman Melville
  </author>
  <illustrator>
    Joe Smith
  </illustrator>
</book>
```

Slide 7 Copyright © 2013 MarkLogic® Corporation. All rights reserved.



SQL to XQuery / XPath: SELECT

SQL

```
SELECT *
FROM BOOKS
```

XQuery / XPath

```
fn:collection()/book
```

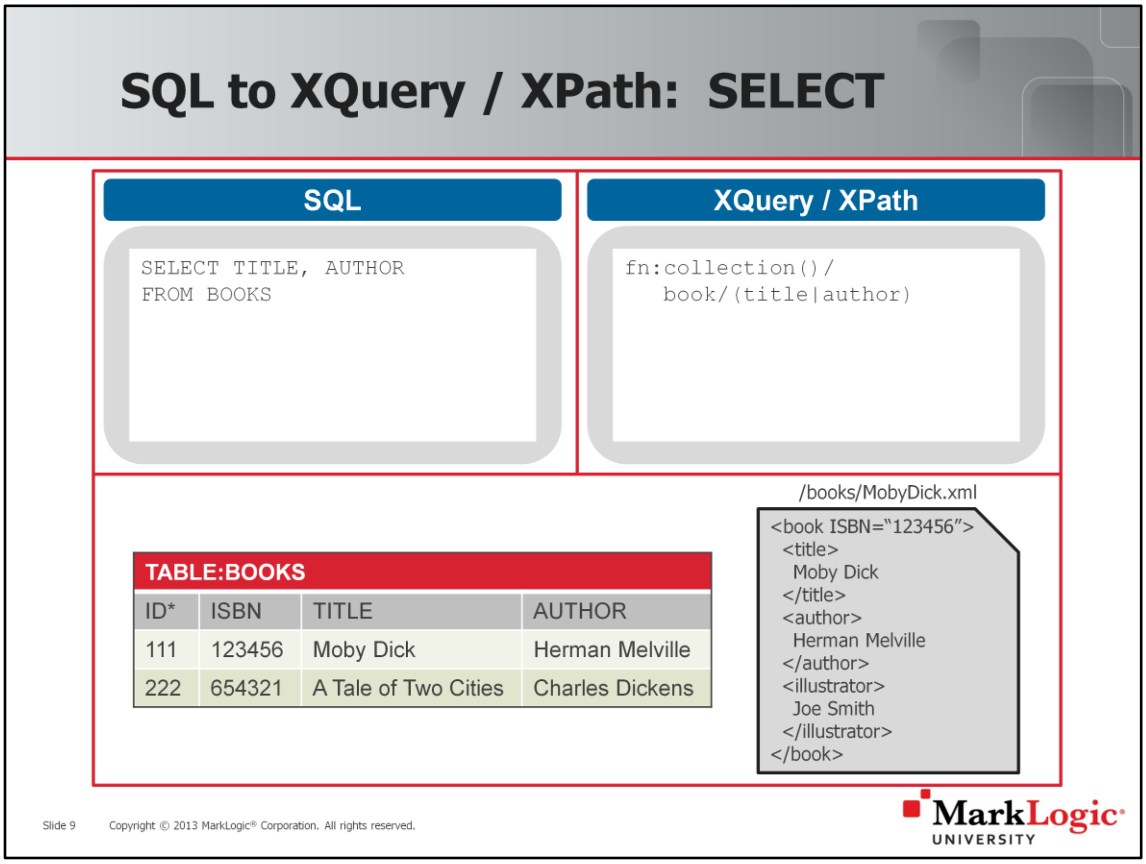
TABLE:BOOKS

ID*	ISBN	TITLE	AUTHOR
111	123456	Moby Dick	Herman Melville
222	654321	A Tale of Two Cities	Charles Dickens

/books/MobyDick.xml

```
<book ISBN="123456">
  <title>
    Moby Dick
  </title>
  <author>
    Herman Melville
  </author>
  <illustrator>
    Joe Smith
  </illustrator>
</book>
```

Slide 8 Copyright © 2013 MarkLogic® Corporation. All rights reserved.



SQL to XQuery / XPath: SELECT

SQL

```
SELECT TITLE, AUTHOR
FROM BOOKS
```

XQuery / XPath

```
fn:collection()/
   book/(title|author)
```

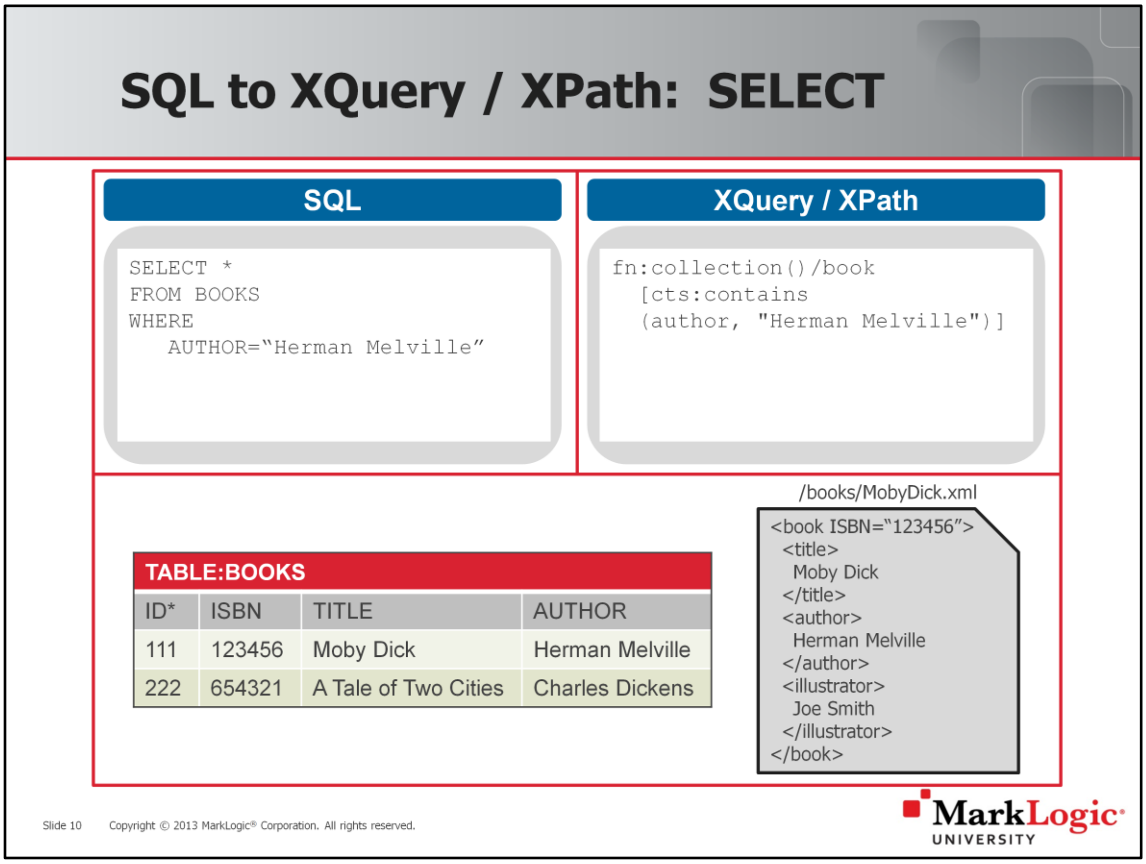
TABLE:BOOKS

ID*	ISBN	TITLE	AUTHOR
111	123456	Moby Dick	Herman Melville
222	654321	A Tale of Two Cities	Charles Dickens

/books/MobyDick.xml

```
<book ISBN="123456">
  <title>
    Moby Dick
  </title>
  <author>
    Herman Melville
  </author>
  <illustrator>
    Joe Smith
  </illustrator>
</book>
```

Slide 9 Copyright © 2013 MarkLogic® Corporation. All rights reserved.



SQL to XQuery / XPath: SELECT

SQL

```
SELECT *
FROM BOOKS
WHERE
   AUTHOR="Herman Melville"
```

XQuery / XPath

```
fn:collection()/book
  [cts:contains
  (author, "Herman Melville")]
```

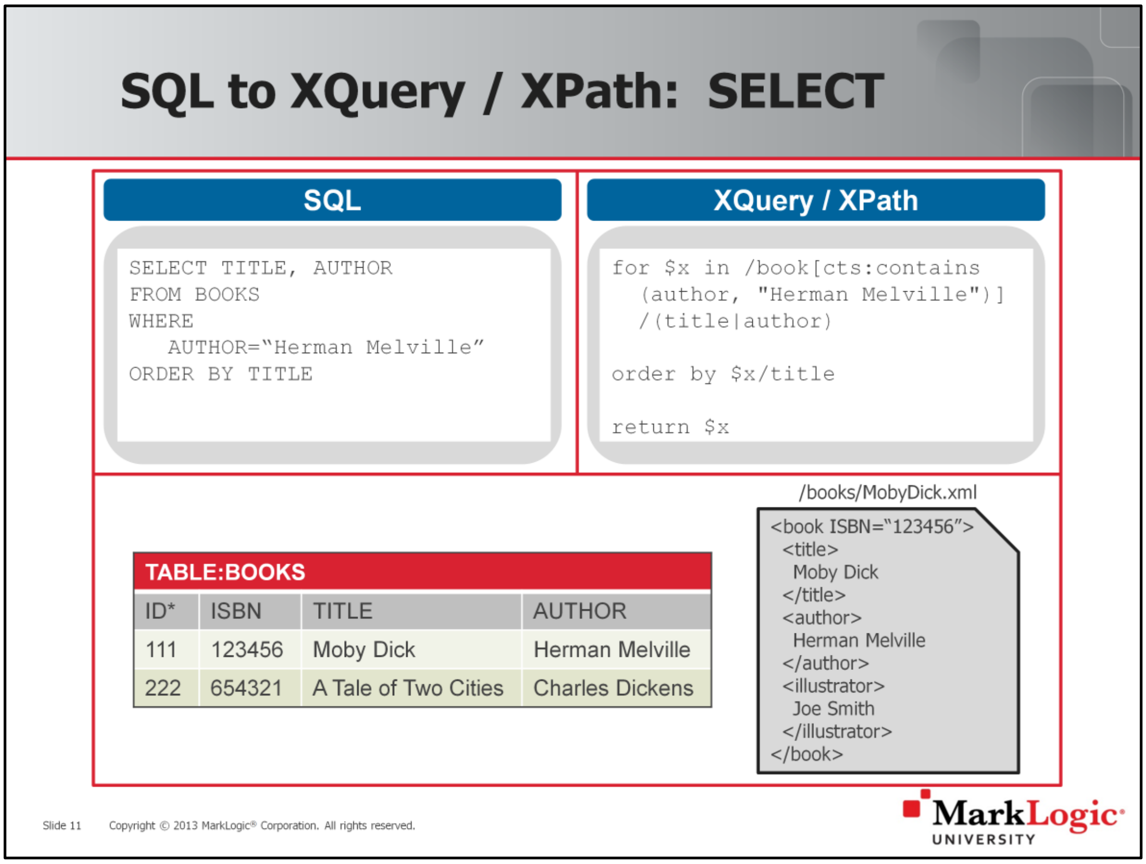
TABLE:BOOKS

ID*	ISBN	TITLE	AUTHOR
111	123456	Moby Dick	Herman Melville
222	654321	A Tale of Two Cities	Charles Dickens

/books/MobyDick.xml

```
<book ISBN="123456">
  <title>
    Moby Dick
  </title>
  <author>
    Herman Melville
  </author>
  <illustrator>
    Joe Smith
  </illustrator>
</book>
```

Slide 10 Copyright © 2013 MarkLogic® Corporation. All rights reserved.



SQL to XQuery / XPath: SELECT

SQL

```
SELECT TITLE, AUTHOR
FROM BOOKS
WHERE
   AUTHOR=“Herman Melville”
ORDER BY TITLE
```

XQuery / XPath

```
for $x in /book[cts:contains
   (author, "Herman Melville")]
   /(title|author)

order by $x/title

return $x
```

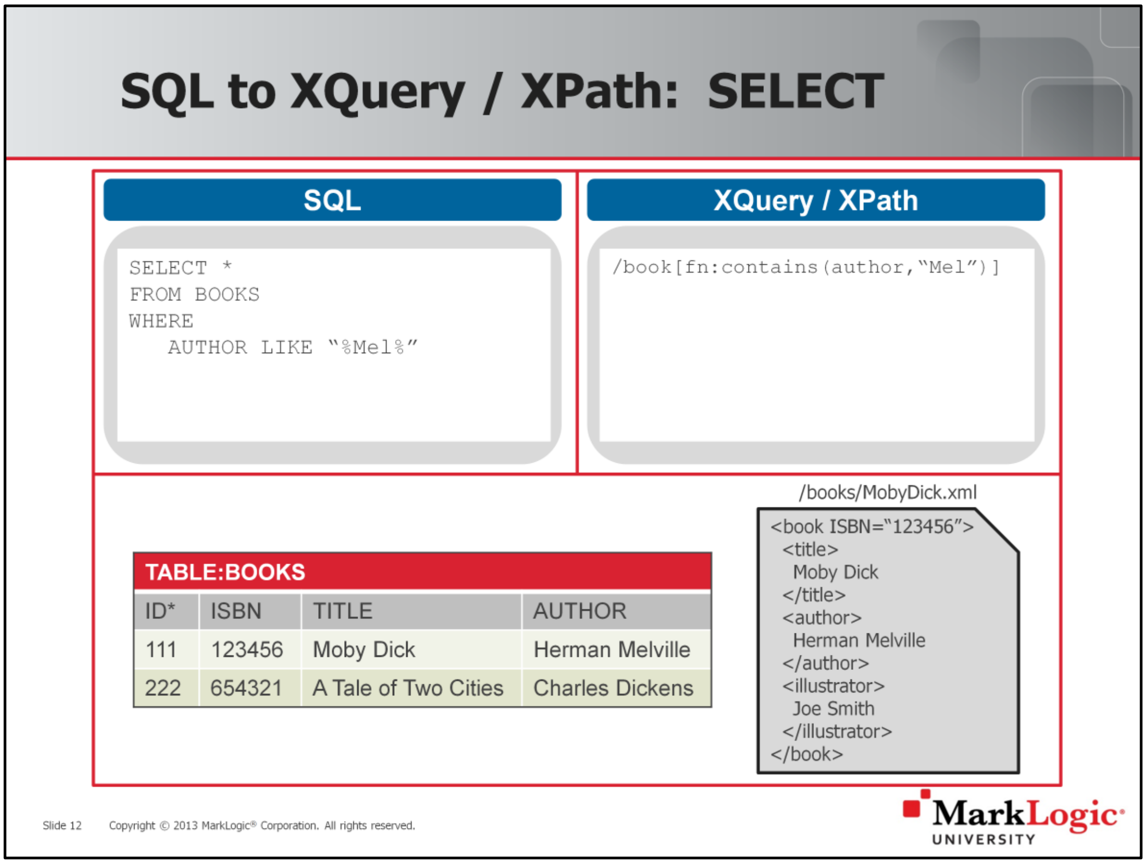
TABLE:BOOKS

ID*	ISBN	TITLE	AUTHOR
111	123456	Moby Dick	Herman Melville
222	654321	A Tale of Two Cities	Charles Dickens

/books/MobyDick.xml

```
<book ISBN=“123456”>
  <title>
    Moby Dick
  </title>
  <author>
    Herman Melville
  </author>
  <illustrator>
    Joe Smith
  </illustrator>
</book>
```

Slide 11 Copyright © 2013 MarkLogic® Corporation. All rights reserved.



SQL to XQuery / XPath: SELECT

SQL

```
SELECT *
FROM BOOKS
WHERE
   AUTHOR LIKE “%Mel%”
```

XQuery / XPath

```
/book[fn:contains(author,“Mel”)]
```

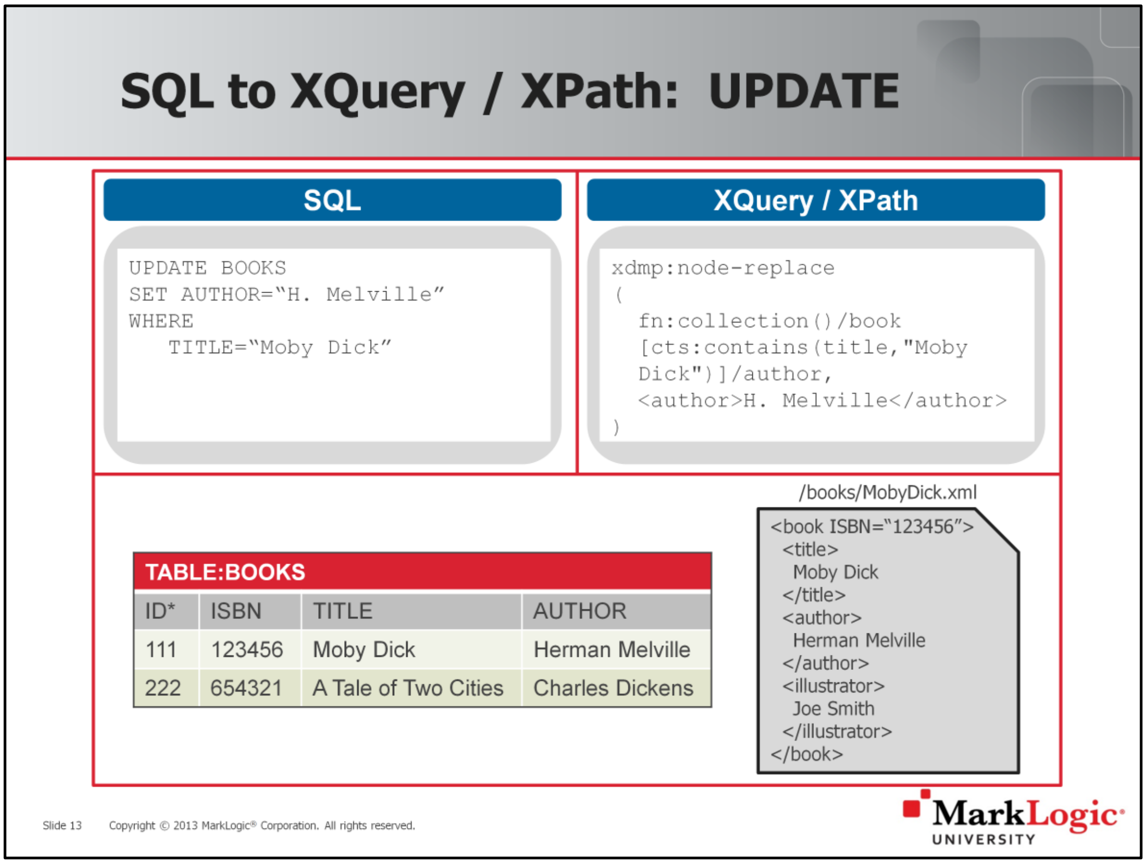
TABLE:BOOKS

ID*	ISBN	TITLE	AUTHOR
111	123456	Moby Dick	Herman Melville
222	654321	A Tale of Two Cities	Charles Dickens

/books/MobyDick.xml

```
<book ISBN=“123456”>
  <title>
    Moby Dick
  </title>
  <author>
    Herman Melville
  </author>
  <illustrator>
    Joe Smith
  </illustrator>
</book>
```

Slide 12 Copyright © 2013 MarkLogic® Corporation. All rights reserved.



SQL to XQuery / XPath: UPDATE

SQL

```
UPDATE BOOKS
SET AUTHOR="H. Melville"
WHERE
   TITLE="Moby Dick"
```

XQuery / XPath

```
xdmp:node-replace
(
  fn:collection()/book
  [cts:contains(title,"Moby
  Dick")]/author,
  <author>H. Melville</author>
)
```

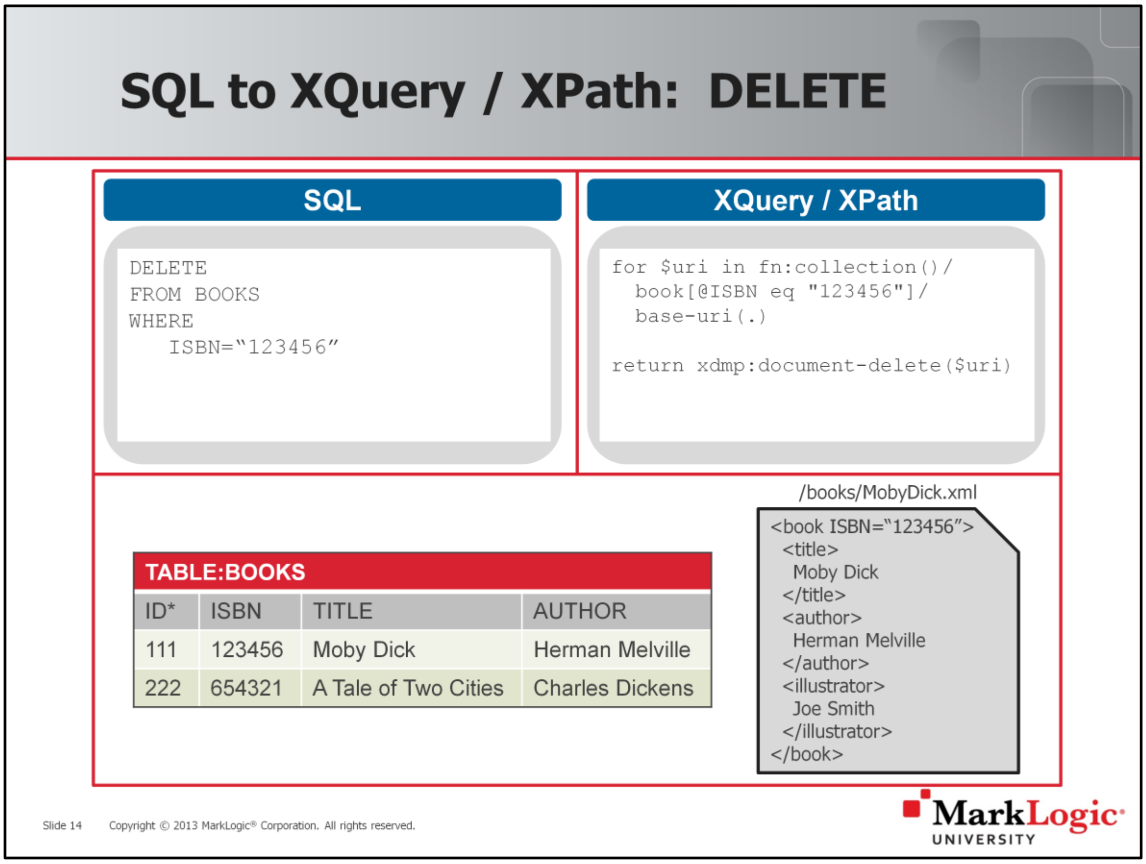
TABLE:BOOKS

ID*	ISBN	TITLE	AUTHOR
111	123456	Moby Dick	Herman Melville
222	654321	A Tale of Two Cities	Charles Dickens

/books/MobyDick.xml

```
<book ISBN="123456">
  <title>
    Moby Dick
  </title>
  <author>
    Herman Melville
  </author>
  <illustrator>
    Joe Smith
  </illustrator>
</book>
```

Slide 13 Copyright © 2013 MarkLogic® Corporation. All rights reserved.



SQL to XQuery / XPath: DELETE

SQL

```
DELETE
FROM BOOKS
WHERE
   ISBN="123456"
```

XQuery / XPath

```
for $uri in fn:collection()/
  book[@ISBN eq "123456"]/
  base-uri(.)

return xdmp:document-delete($uri)
```

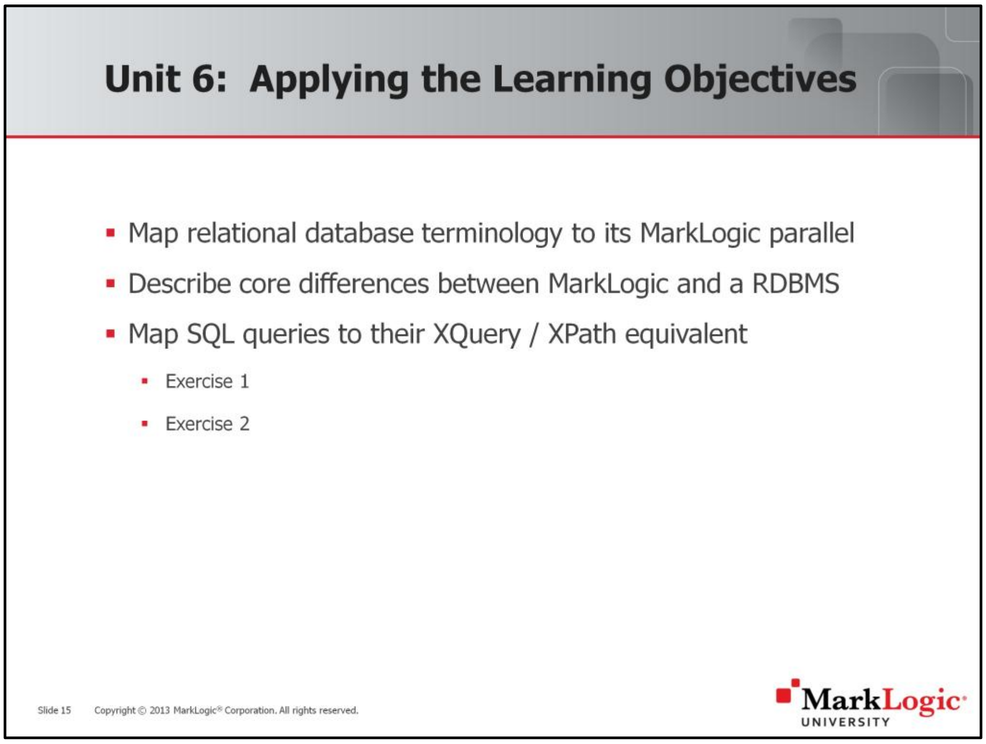
TABLE:BOOKS

ID*	ISBN	TITLE	AUTHOR
111	123456	Moby Dick	Herman Melville
222	654321	A Tale of Two Cities	Charles Dickens

/books/MobyDick.xml

```
<book ISBN="123456">
  <title>
    Moby Dick
  </title>
  <author>
    Herman Melville
  </author>
  <illustrator>
    Joe Smith
  </illustrator>
</book>
```

Slide 14 Copyright © 2013 MarkLogic® Corporation. All rights reserved.



Unit 6: Applying the Learning Objectives

- Map relational database terminology to its MarkLogic parallel
- Describe core differences between MarkLogic and a RDBMS
- Map SQL queries to their XQuery / XPath equivalent
  - Exercise 1
  - Exercise 2

Slide 15 Copyright © 2013 MarkLogic® Corporation. All rights reserved.