

1.4.3 論理設計手順の比較

論理設計手順の比較

- リレーショナルデータベース
 - 正規化
 - > データの整合性や保守性を高めるために必要
 - 表定義など
 - > 表定義や、表制約、インデックスなどの定義
- XMLデータベース
 - スキーマ設計
 - > 要素名、属性名、データ型、XML構造などの決定
 - > 正規化などの手順を厳密に行わない論理設計が可能
 - スキーマ定義の作成
 - > スキーマの登録を必要としないデータベース製品もある
 - > スキーマを作成してもデータベースにスキーマを登録しない方法もある

論理設計手順の比較

リレーショナルデータベースを構築する場合は、正規化、表の定義などを厳密に行う必要があります。一方XMLデータベースの場合、データベース製品によってはスキーマ定義が不要な場合もあり、スキーマ定義を作成する手順と作成しない手順が考えられます。

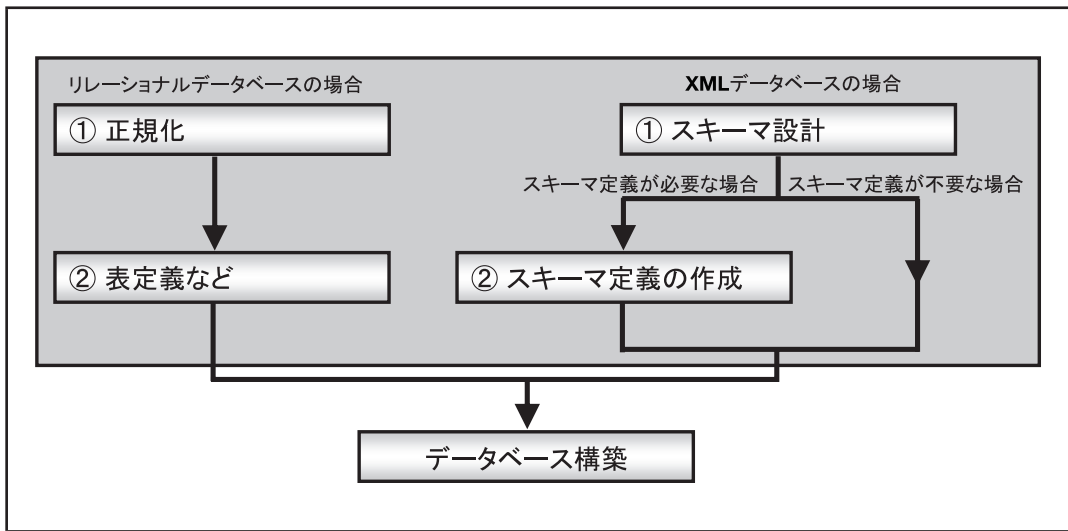


図1.4.7 論理設計手順の比較

リレーショナルデータベースの場合

1. 正規化

リレーショナルデータベースを利用するシステムでは、データの整合性や保守性を高めるためにデータの正規化が行われます。一般には第3正規形に正規化されますが、業務要件やアプリケーションからのデータの操作性などにより、正規化のレベルが調整されます。

2. 表定義など

正規化を行ったデータを元に表定義、表制約（主キーや一意キーなど）、インデックスなどのデータベースオブジェクトを作成するための定義を行います。

XMLデータベースの場合

1. スキーマ設計

XMLデータベースのスキーマ設計では、要素や属性の名称と、必要に応じてそのデータ型を決定します。そしてXML文書の構造と、併せてXMLの名前空間や、文字エンコーディングとして何を使用するかなども決定します。

XMLデータベースを利用するシステムでは、XML文書が自由な階層構造を表現でき、またスキーマ定義として要素の出現回数などに柔軟性を持たせることができますので、正規化などの手順を厳密に行わない論理設計が可能です。しかしデータの一貫性を考えた場合に、XMLデータの正規化を考慮することが必要なケースもあります。その場合もXMLデータの結合頻度や、アプリケーションからのデータの操作性などを考慮し、正規化のレベルを調整することでXMLデータの特性を活かした効率のよいデータ管理を行うことができます。

2. スキーマ定義

XMLデータベースでのスキーマ定義は、通常XML SchemaなどのXMLスキーマ定義言語で行います。スキーマの登録を必要としないデータベース製品の場合、定義言語での定義を省略する場合があります。

※XMLデータベースにスキーマを登録すると、XMLデータの挿入や更新時にスキーマ検証が実行され、データの更新が多い場合などでパフォーマンス的に不利に働く場合も考えられます。その場合は作成したスキーマをXMLデータベースに登録せず、必要時にアプリケーション側でスキーマ検証を実行する方法もあります。