WHY THE NEW 23C DATABASE FEATURE JSON RELATIONAL DUALITY VIEWS MAY CHANGE THE LIFE OF DEVELOPERS

NLOUG Database Cloud day 8 dec 2023

Peter de Vaal

Transfer Solutions

TRANSFER SOLUTIONS

IS THERE STILL PLACE FOR THE RELATIONAL DATABASE IN AN OBJECT-ORIENTED WORLD?

A HISTORY OF DB APPLICATION DEVELOPMENT

- **Database server with hosted application**
  - Application runs on DB server
  - Has persistent connection to the database
  - Uses SQL for DB operations
- **Client-Server**
  - Application runs on client (Oracle Forms, VB, Delphi, Java, etc.)
  - Has persistent connection to the database
  - Uses SQL and PL/SQL for DB operations

A HISTORY OF DB APPLICATION DEVELOPMENT:
ORACLE MOVES TO THE WEB

- **DB server with web Forms**
  - Application runs on the middletier
  - Has persistent process and connection per session to the database
  - Uses SQL and PL/SQL for DB operations
- **APEX**
  - Application runs in the database, with listener on the middletier
  - Accesses the database using shared connection pool
  - Uses SQL and PL/SQL for DB operations

A HISTORY OF DB APPLICATION DEVELOPMENT:
MOVING TO OBJECT ORIENTED APPLICATIONS

- **3-tier or webarchitecture with application on middletier**
  - Application runs on the middletier
  - Has shared process and connection pool for all users
  - Developed using object oriented language/framework
  - May use an ORM framework for access to the Database
  - SQL generated or written by non-db developers/parties
- **SOA architecture with services on middletier**
  - Application and services run on the middletier
  - Services often use XML as input and output format
    - A service bus or SOA composite maps XML to SQL operations
    - May use an ORM framework for the mapping
    - SQL generated or written by non-db developers/parties

A HISTORY OF DB APPLICATION DEVELOPMENT:
MOVING TO REST SERVICES

- **Modern webarchitecture based on REST services**
  - Application runs on the middletier
  - Application uses REST services
  - Services mostly use JSON as input and output format
  - A service bus or API platform maps JSON to SQL operations
- **Oracle REST data Services (ORDS)**
  - Same as above, but …
  - Service definition in the database, maps input to SQL or PL/SQL
  - All SQL mainatined by DB developers

WHAT CHANGED FOR A DEVELOPER?

- **Traditionally one team developed both database schema and GUI**
  - SQL and PL/SQL was called for (data) logic and persistence
  - All developers had good SQL skills
- **In 3-tier and SOA architecture 2 or more teams are involved**
  - GUI specialists for the front-end
  - Integration specialists for the ORM and/or Service frameworks
  - Database specialists for the database schema

STRONG ASPECTS OF 3-TIER ARCHITECTURE

- **Separation of concerns**
  - GUI design separate from logic and persistence
  - Services instead of direct db connections for well defined APIs for integration and reusability
  - Database developers can focus on good relational model design

WEAK ASPECTS OF 3-TIER ARCHITECTURE

- **ORM frameworks all have weaknesses**
  - Difficult to get optimal performance
    - Lack of SQL knowledge in middletier team
    - Sometimes extreme data overhead needed for business rule checking on the middletier
  - Problems on any relational model changes
    - Refactoring ORM mapping might be difficult
    - Impact analysis of database schema changes hardly possible
- **Services may cause a performance overhead**
  - SOAP services in particular
- **Need for specialists for the middletier operational aspects**
  - Deployment, tuning, upgrading and patching etc. all requires skills that are not readily available with developers

APPROACH TO EFFECTIVE 3-TIER ARCHITECTURE

Move Object-Relational Mapping to the database!

ORM IN THE DATABASE : HOW AND WHY

- All modern relational databases understand XML and JSON
- XML or JSON input can be transformed to SQL result sets
- Contents of relational tables can be represented as XML or JSON
- Only one call to the database needed instead of multiple in case of ORM on the middletier
- Use database views and database packages as APIs
- Can be used in transactional architectures with support for distributed transactions, as well as in REST architectures
- Easy impact analysis and refactoring of mapping in the database

DATA RETRIEVAL USING DATABASE MAPPING

- **Use one PL/SQL packaged function per object type or document**
  - Use an XML or JSON type parameter for specifying filters
  - 2 options for retrieving the data in the package
    - Use queries on tables or views
    - use pl/sql pipelined functions for parametrized queries
    - Both use SQL/XML or SQL/JSON functions to convert to that format
  - Function returns XML or JSON document (as retrun value or in OUT parameter)
- **Use views**
  - 2 options for producing output
    - Map SQL to XML or JSON output using SQL/XML or SQL/JSON functions
    - Use Oracle REST Dataservices to return the result set as JSON

DATA MANIPULATION USING DATABASE MAPPING

- **Use a PL/SQL packaged procedure per object/document**
  - Most objects or documents map to more than one table
    - Code in the packages required to do DML on all involved tables
    - Rework needed when datamodel changes are made

INTRODUCING JSON RELATIONAL DUALITY VIEWS

- **Available in Oracle 23c**
- **New db object that maps a JSON document to relational tables**
  - View has only one column (DATA) containing the JSON document
  - A SELECT on the view returns the document enriched with data from the tables
  - View body uses SQL or GraphQL to define the mapping
  - View can be used for queries
  - View can be used for DML on one or more of the tables

ADVANTAGE OF DUALITY VIEWS

- **Declarative approach to ORM mapping in the database**
  - No more coding for DML
- **Supports ACID compliant DML, even from webapplications**
  - No more concurrent updates that overwrite each other
- **Developers can focus on area of their own skills**
  - Application and middleware developers use JSON
  - Database developers use SQL or GraphQL

TRANSFER SOLUTIONS

WHAT ABOUT XML SUPPORT?

- **Duality views only support JSON**
- **Many industry standards are (still) based on XML**
- **Solution: use transformation on middleware**

- **Recommendation for Oracle for next release:**
  - XML Relational Duality Views to support many industry standards

QUESTIONS?

TRANSFER SOLUTIONS