



Einsteiger Stream - Anwendungsentwicklung: Fit für MicroServices – REST und JSON aus der Datenbank

Robert Marz



Robert Marz – Independent Consultant



Client

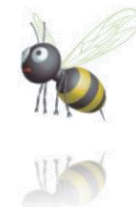
Senior Technical Architect
with database centric view of the world



ORACLE
ACE

its-people

Portfolio Manager Database Technologies
Blog Editor



DOAG (German Oracle User Group)

Active Member of Database Community
Responsible for Cloud Topics



@RobbieDatabee



blog.its-people.de



Robert.Marz
@its-people.de



Microservices, Containers & The Database

API

Microservices

Architectural Style,
variant of Service-
Oriented-Architecture

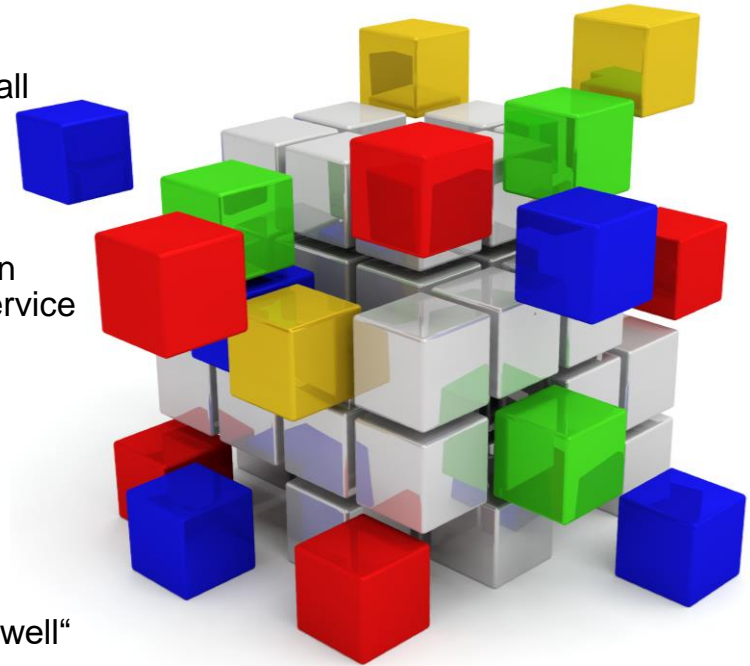
Deploy Independently
Couple Loosely
(e.g. REST API)

Every Microservice implements a
single Business Function

Split complex Systems in small
Services

A single Team is in
Charge for the Service

„Do One Thing and Do it well“



Containers (e.g. Docker)

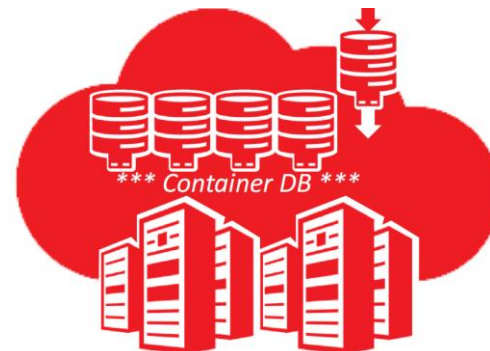
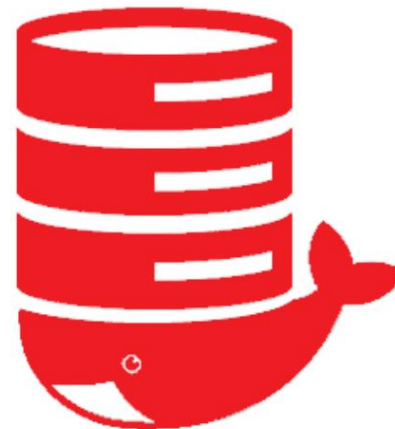
- are OS-level virtualization
- compose and deploy lightweight Libraries and Components
- Docker Containers can be deployed in parallel for Scaling
- Orchestration Software like Kubernetes

Oracle RDMS

- is **NOT** lightweight
- is supported to run in Docker
- even with RAC
- Ideal for Development Environments

Oracle RDBMS as container Host

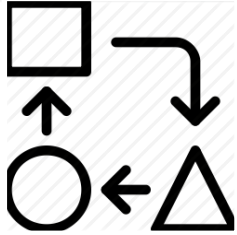
- Multitenant Option
- Application Containers
- Use this for Production



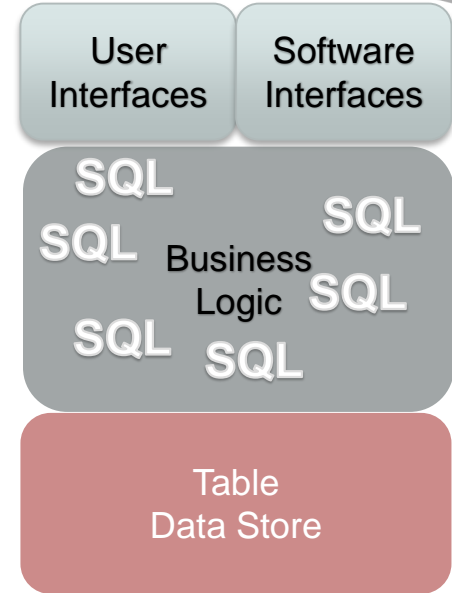


Where Business Logic belongs

Where to put the Business Logic?

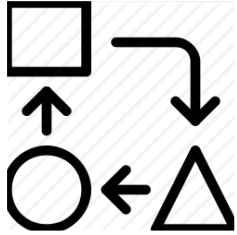


Most Business Apps are Data-Centric



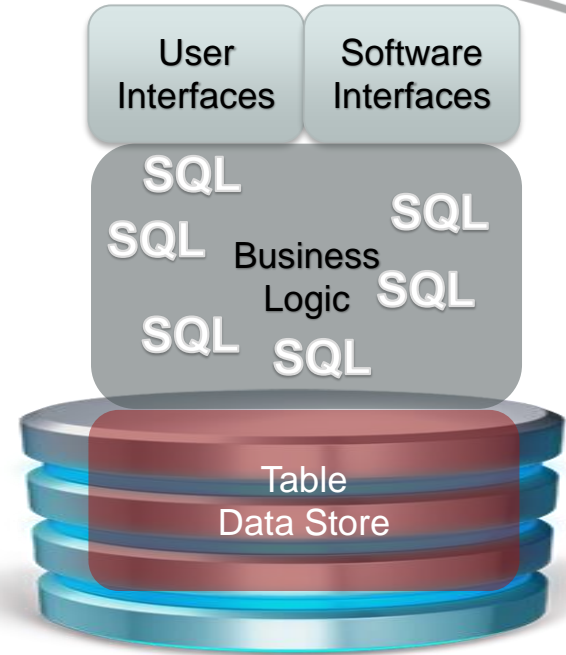
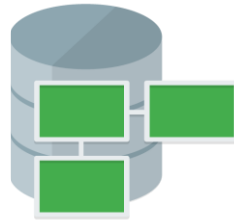
Application Layers

Where to put the Business Logic?



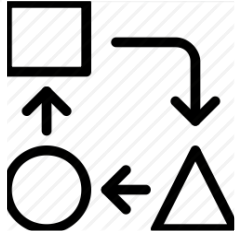
Most Business Apps are Data-Centric

Business Logic is all about Data



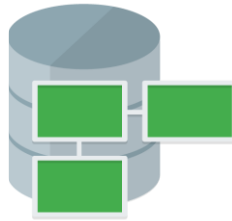
DBMS = Persistence Layer
"NoPL/SQL"

Where to put the Business Logic?

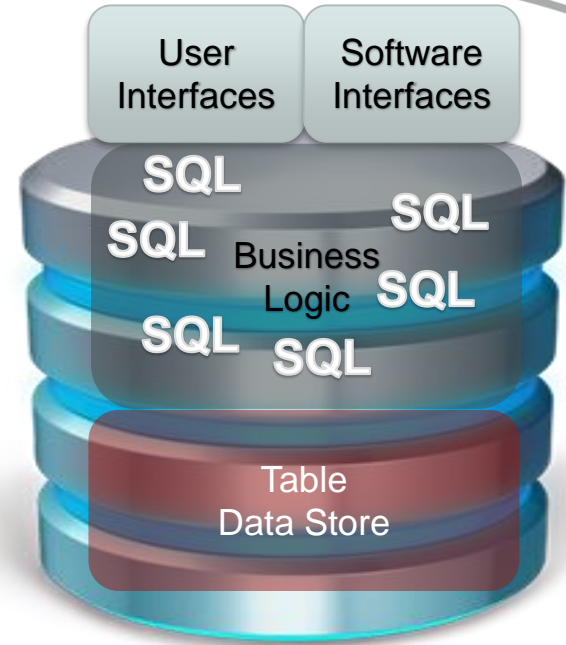


Most Business Apps are Data-Centric

Business Logic is all about Data



It belongs as close as possible to the data



DBMS = Processing Engine
"SmartDB"

#ThickDB

Hide the Database
away

Client-side code only
invokes Database
Procedures

Don't let App access
Tables directly

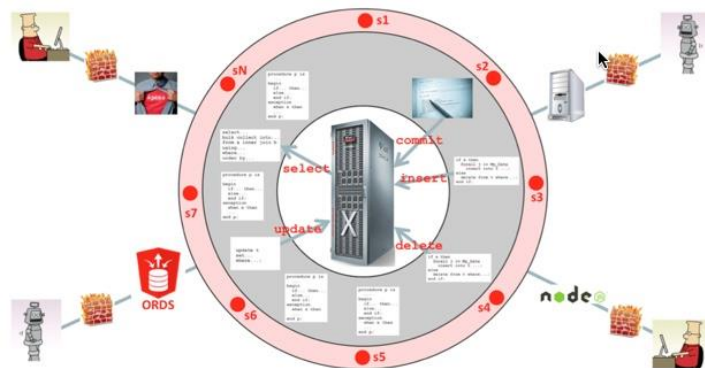


Figure 2. The thick database paradigm.

No select

No insert

No update

No delete

No commit

No rollback

#SmartDB

Evolution of #ThickDB

Use the Database as Processing Engine, not only as Persistence Layer

Protect your Data with a #HardShell API

Makes your Apps faster and more secure

A large, 3D-rendered blue cylinder with a metallic sheen is the central focus. A large, semi-transparent red arrow points from the left side of the cylinder towards the right. Three semi-transparent dark red rounded rectangular boxes are overlaid on the cylinder, each containing white text. The background is a dark blue field with a pattern of glowing, out-of-focus light blue squares and rectangles, suggesting a digital or data environment.

Protect your
Data with
PL/SQL APIs

Publish them
as REST APIs

These are
your
Microservices

A woman is sleeping peacefully in a bed, wearing a white t-shirt and a grey eye mask. She is covered with a light blue blanket. To her left, a black metal plant stand holds several potted plants, including a succulent and a fern. A white mug sits on one of the shelves. The room has dark grey curtains on the left and a plain white wall in the background.

RESTful Services with ORDS AutoREST

What is REST



REST	Representational state transfer	doctoral dissertation by Roy Fielding, 2000
	programming paradigm	distributed systems Web services.
RESTful Applications	implements 6 constraints	Uniform Interface (API via URIs) Stateless , Client-Server, Layered System Cacheable , Code on Demand
Implementation	Transport protocol	http(s)
	content	JSON Documents

Oracle REST Data Services (ORDS)



Evolved

from APEX Listener

Deploy in Application Server

- Tomcat
- Glassfish (deprecated)
- WebLogic

ords.war

Java Web Archive

Standalone mode

Brings own http-server
Supported for Production
java -jar ords.war



AutoREST

REST enable Table or Procedure very easy

Read / Write

Stateless / no transaction


SQLDeveloper Wizards

PL/SQL API

```
begin
  ords.enable_schema;
  commit; -- This commit is important!
end;
/
```

```
begin
  ords.enable_object (
    P_OBJECT      => 'STOCKTICKER',
    P_OBJECT_TYPE => 'TABLE',
    P_OBJECT_ALIAS => 'tab-StockTicker');
  commit; -- This commit is important, too!
end;
/
```

Case Sensitive

A thick red arrow pointing horizontally to the left, positioned to the left of the vertical text 'Case Sensitive'.

Anatomy of a ORDS AutoREST URL



Protocol

ORDS Base

Table

<http://192.168.56.101:8080/ords/rdbl/Tab-StockTicker/?offset=55&limit=100>

/ORCL,2016-01-08

Host:Port

Schema

Parameter

PKCol1,PKCol2
or Primary Key

HTTP Method	ORDS AutoREST Action
GET	Retrieve Data – Single Row or Rowset
PUT	Insert or Modify Row
POST	Bulk Insert csv-data
DELETE	Delete Row

Interpreting the ORDS AutoREST Responses (1)



```
1 {
2   "items": [
3     {
4       "symbol": "TDC",
5       "id1": 56,
6       "tstamp": "2017-05-06T23:27:00Z",
7       "price": 20.625,
8       "links": [
9         {
10          "rel": "self",
11          "href": "http://127.0.0.1:8080/ords/rdb1/Tab-StockTicker/56"
12        }
13      ]
14    },
15    {
16      "symbol": "ORCL",
17      "id1": 57,
18      "tstamp": "2017-05-06T23:28:00Z",
19      "price": 42,
```

Table Rows

Columns & Values

```
1 {
2   "items": [↔],
28  "hasMore": true,
29  "limit": 2,
30  "offset": 55,
31  "count": 2,
32  "links": [
33    {↔},
37    {↔},
41    {↔},
45    {↔},
49    {
50      "rel": "next",
51      "href": "http://127.0.0.1:8080/ords/rdbl/Tab-StockTicker/?offset=57&limit=2"
52    },
53    {↔}
57  ]
58 }
```

Hypermedia as the Engine of Application State (HATEOAS)



Building ORDS RESTful Services

Connection Tree

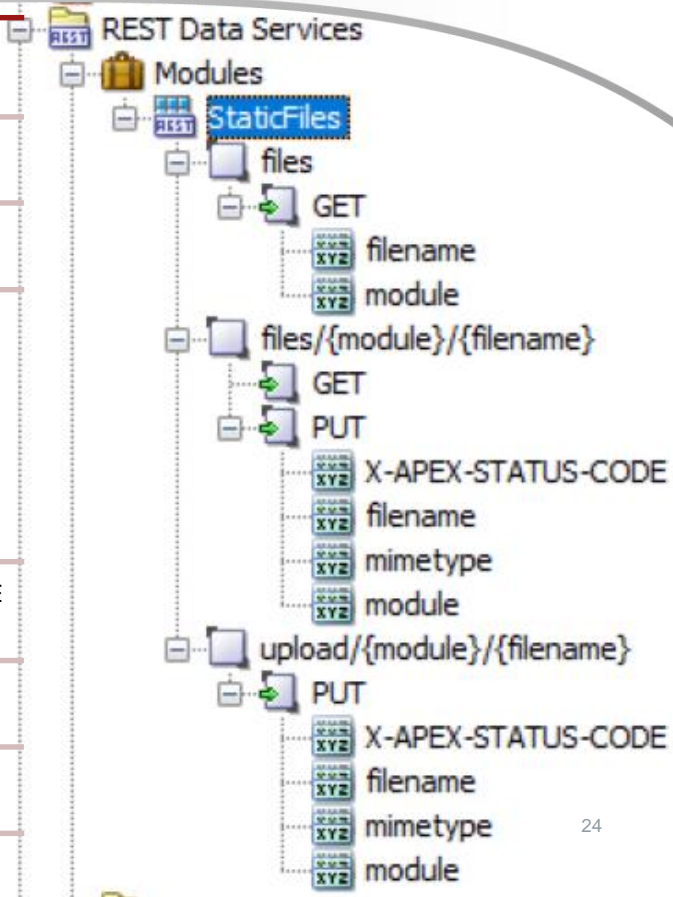
Right Click
Schemas & Objects
REST Data Service Node

SQL Worksheet

Right Click Query Results

REST Data Service

Module	logically groups a set of URLs	Like a PL/SQL Package
	Name	Include API Version# In Name & URI Prefix
	URI Prefix	Part of the URL
Template	URI Pattern	following URI Prefix from Module may contain parameters
	Handler	http-Methods one per Template
Handler	Parameters	http-header URI
	Your Code	goes here



Handlers & Parameters



Handler: Source Type

All Handlers:

PL/SQL

GET Handler
(Output Format varies)

Collection Query [Item]; Query [One Row]; Feed

Media Ressource

Handler: Results

All Handler

JSON

GET Source-Type

Query: JSON or CSV

Media Ressource: Binary

Parameter

Types

In

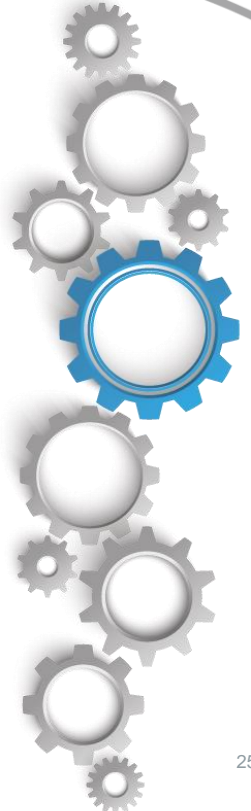
Out, In / Out

passed as
Bind Variables

Source Type

http-Header

URI



Generate & Process JSON



What is JSON?



JSON

Java Script
Object Notation

Document Format

“Key”:“Value” Pairs

{ } – Groupings

[] - Arrays

Think of XML with
<Tags> replaced
by brackets

UTF-8

English Numbers & Keywords
e.g. true, false

Locale

```
{
  "name": "STOCKTICKER",
  "primarykey": [
    "symbol",
    "tstamp"
  ],
  "members": [
    {
      "name": "symbol",
      "type": "VARCHAR2"
    },
    {
      "name": "tstamp",
      "type": "DATE"
    },
    {
      "name": "price",
      "type": "NUMBER"
    }
  ],
  "links": [
    {
      "rel": "collection",
      "href": "http://192.168.56.101:8080/ords/rmougprov/metadata-catalog/",
      "mediaType": "application/json"
    },
    {
      "rel": "canonical",
      "href": "http://192.168.56.101:8080/ords/rmougprov/metadata-catalog/tab-StockTicker/"
    },
    {
      "rel": "describes",
      "href": "http://192.168.56.101:8080/ords/rmougprov/tab-StockTicker/"
    }
  ]
}
```



Oracle deals with JSON Documents natively

JSON is stored in Varchar2 or LOB Columns

SQL-Functions

JSON_TABLE Operator

Extended PL/SQL Support since 12cR2

JSON_TABLE

Lives inside the SQL-From-Clause

Produces **Rows** and **Columns**

Accepts LOBs with JSON data

Included in SQL:2016 Standard

The JSON_TABLE Operator (2/2)



The JSON Document

```
select wert
  from json_table( '["Eins", "Zwei", "Drei",
                    "Vier", "Fünf", "Sechs"]'
                  , '$[*]'
                  columns wert varchar2 path '$'
                )
/
```

Produces rows

Produces columns

WERT

Eins
Zwei
Drei
Vier
Fünf
Sechs

6 rows selected

Elapsed: 00:00:00.011

Generate JSON Documents

SQL-Functions starting with
12gR2

- can be nested

JSON_Object

JSON_Array

JSON_ArrayAgg



```
SELECT JSON_OBJECT ('id2' VALUE cust_id,  
                   'name' VALUE (first || ' ' || last),  
                   'joined' VALUE join_date) FROM customers;
```

```
{"id2":1,"name":"Eric Cartman","joined":"2017-07-10T16:26:15"}  
{"id2":2,"name":"Kenny McCormick","joined":"2017-07-10T16:26:15"}  
{"id2":3,"name":"Kyle Brofloski","joined":"2017-07-10T16:26:15"}  
{"id2":4,"name":"Stan Marsh","joined":"2017-07-10T16:26:15"}
```

```
SELECT JSON_ARRAY(first, last)
       FROM customers;
```

```
["Eric", "Cartman"]
["Kenny", "McCormick"]
["Kyle", "Brofloski"]
["Stan", "Marsh"]
```

```
SELECT JSON_ARRAYAGG(  
    JSON_OBJECT('id' VALUE item_id,  
               'name' VALUE name,  
               'quantity' VALUE stock_quantity))  
FROM items  
WHERE stock_quantity > 10;
```

```
[{"id":345,"name":"Border Patrol Costume","quantity":20},  
{"id":429,"name":"Air Guitar","quantity":14}]
```

ā'pěks

(#orclapex)

Oracle APEX
Low Code App Development



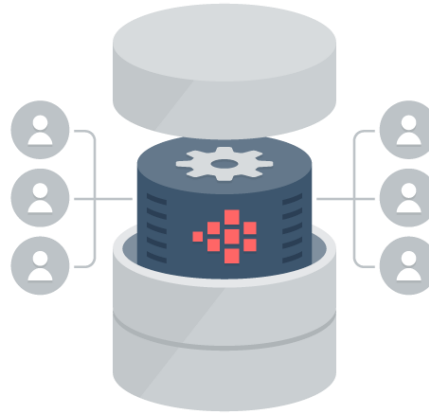
Oracle Application Express

Database-centric web
application
development
framework

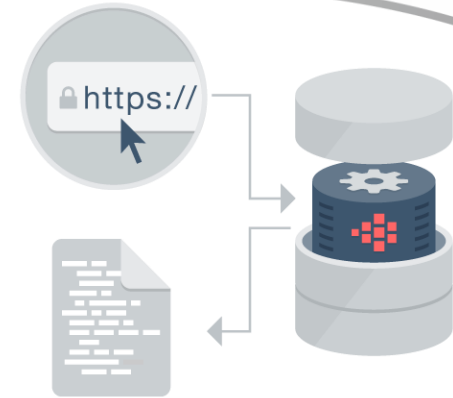




App Development
IDE is a web browser.
No client software
needed



App definitions are stored
in the database as meta
data.
Declarative – No code
generation



Page generation is
efficient with only one
request and one response.
Data processing done in
the Database

Oracle APEX: Application Builder



The image displays two overlapping screenshots of the Oracle APEX Application Builder interface. The background screenshot shows a dashboard with three circular gauges: 'Pages' at 2,998, 'Packaged Applications' at 1, and 'Worksheets' at 8. Below these is a 'Page Events' chart showing activity from May 31 to June 08, with two highlighted data points: 'HILARY' at 15% (1513) and 'SHAKEEB' at 64% (6409). The foreground screenshot shows the 'Property Editor - Region' for a 'Week's Top Developers' region, displaying a SQL statement and various configuration options.

```
select * from user_developer, count(*) page_views from www_flow_builder_audit_trail where audit_date > sysdate - 7 and security_group_id = vflow_security_group_id group by *flow_user
```

Integrated Development Environment (IDE) with Application Development Graphical “Page Designer”

Features tight SQL and PL/SQL integration

REST Support continuously improved

Provides REST SERVICES

similar to ORDS REST Services

Consumes REST Services

e.g. samples REST API

Build Interactive Grids based on REST Services



Conclusion



Oracle REST Data Services:

[ORDS Download](#)

[ORDS Documentation](#)

[Oracle Visual Builder Add-in for Excel](#)

[Oracle DB Tools GitHub ORDS examples](#)

PL/SQL:

[#SmartDB](#)

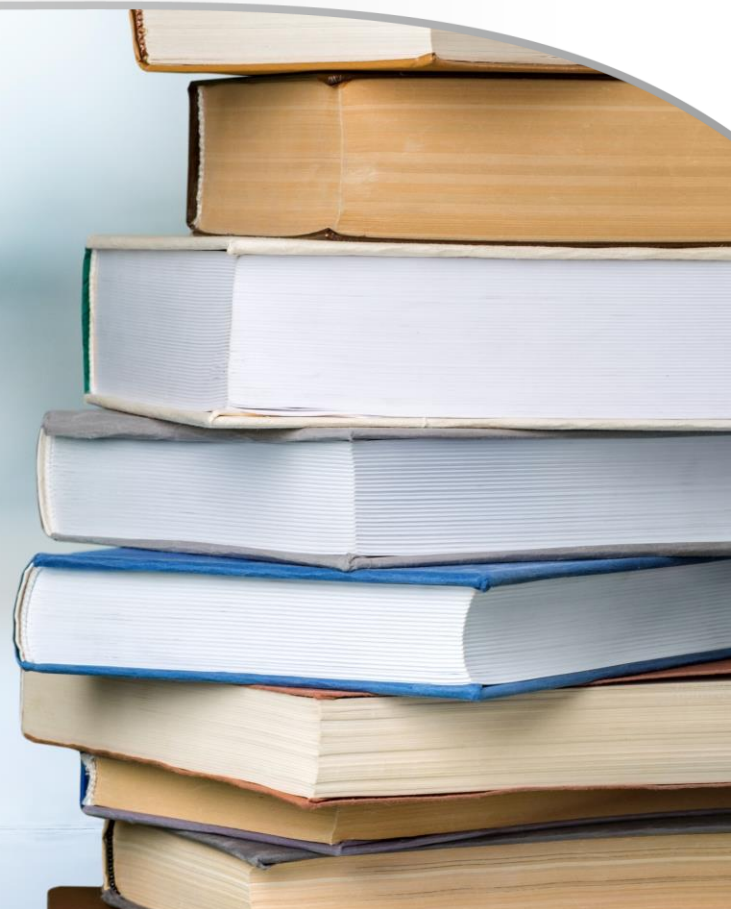
[#ThickDB](#)

[#PinkDB](#)

Blogs:

[Jeff Smith \(Oracle Productmanager\)](#)

[Kris Rice \(Oracle VP Development\)](#)



- The Oracle RDBMS fits perfectly into Microservice Architectures
- Business Logic belongs close to the Data. Think about #SmartDB
- REST and JSON support turns the Database into ideal Backend for Web Apps
- Oracle Apex rocks! #LetsWreckThisTogether





Vielen Dank für Ihre
Aufmerksamkeit.