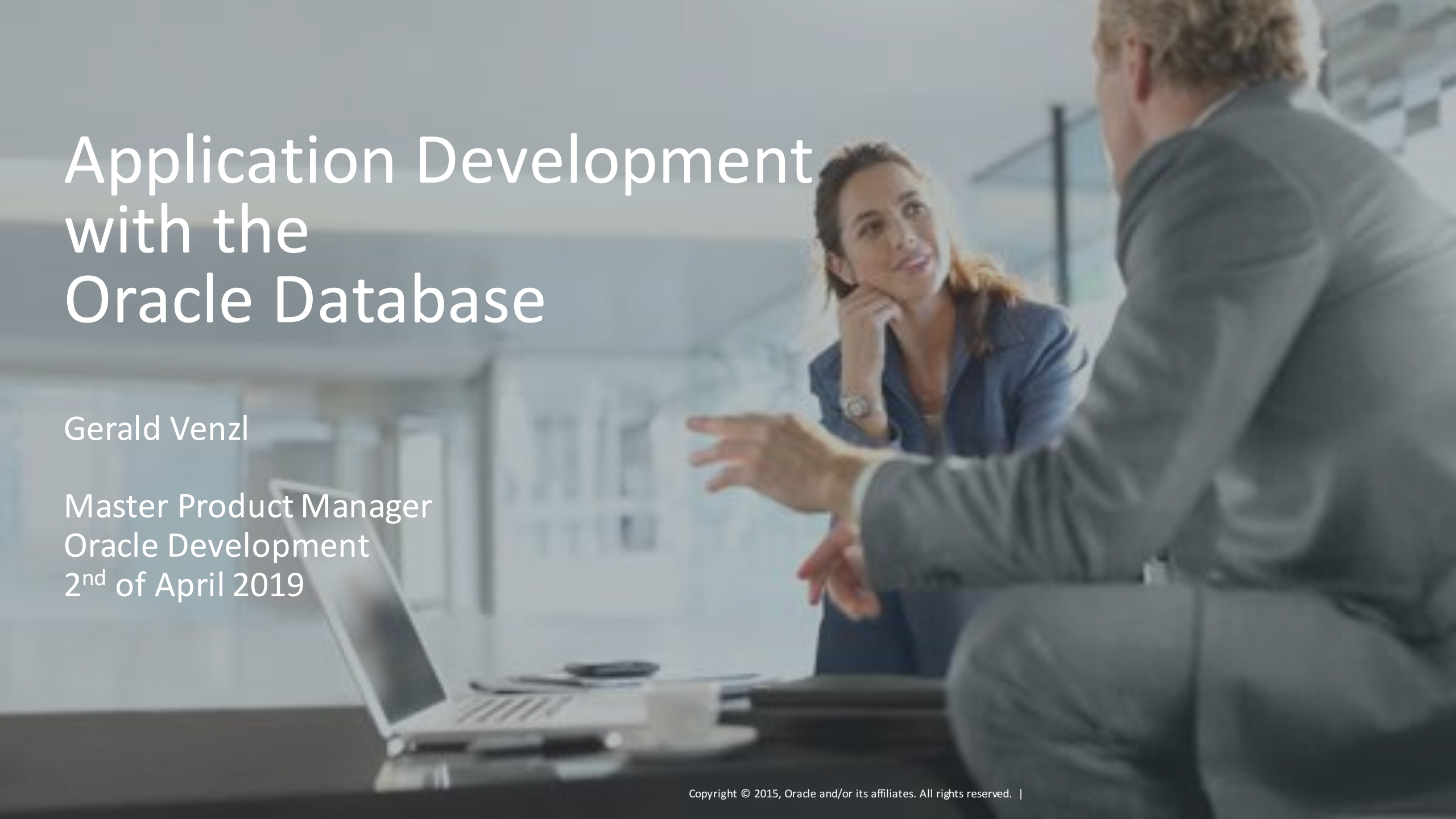

ORACLE®

Application Development with the Oracle Database

Gerald Venzl

Master Product Manager
Oracle Development
2nd of April 2019



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted

Program Agenda

1. Overview and History
2. Vision
3. Data Management Strategy
4. Oracle and Modern Development
5. Open Source initiatives
6. Developer centric functionalities
7. Q & A

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

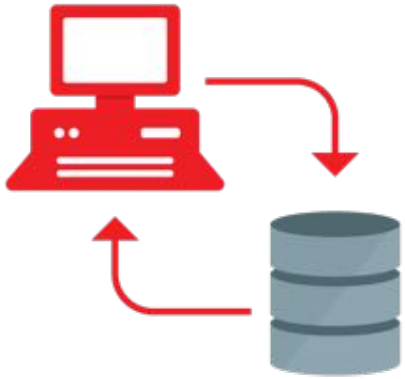
Program Agenda

1. Overview and History
2. Vision
3. Data Management Strategy
4. Oracle and Modern Development
5. Open Source initiatives
6. Developer centric functionalities
7. Q & A

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Continuous Oracle Database Innovations

Preserving customer's investment though each new Computing Era



Client-Server

Stored Procedures
Partitioning
Parallel Query
Unstructured Data

Internet

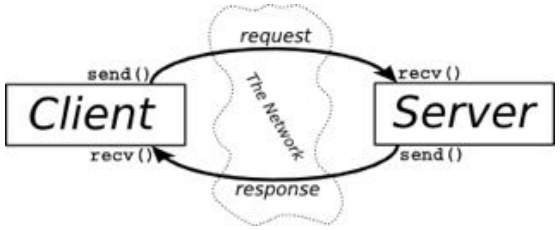
Resource Management
Real Application Clusters
Data Guard
XML

Big Data & Cloud

Big Data SQL
Multitenant
In-Memory
JSON

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Application Development Over The Years



Mobile & Cloud

Release	1985 – 1997: 6, 7 and 8	1998 – 2012: 8i, 9i, 10g, 11g	2013 - 2019: 19c
Developer	Stored Procedures & Triggers Referential Integrity Distributed Transactions AQ LOBs Spatial	Java .NET PHP XML APEX	Open Source Drivers (Python, Node.js and R) Pattern Matching OpenSource Drivers JSON REST Data Services NoSQL Database Application Continuity Migration Framework HTML5 – Desktop & Browser Javascript Opensource Cloud
Engine	OLTP throughput (Row Locking, MVRC) Parallel Query Partitioning	Online Operations RAC Data Guard Flashback Self-Managing Database Enterprise Manager Resource Management Automatic Storage Mgmt Encryption Real Application Testing Row Compression Columnar Compression Smart Scans Flash Cache	Multitenant Database In-Memory Column Store

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. |

Program Agenda

1. Overview and History
2. Vision
3. Data Management Strategy
4. Oracle and Modern Development
5. Open Source initiatives
6. Developer centric functionalities
7. Q & A

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Oracle Database as a Data Platform

Development Services

Node.js, Python, .NET, Java, PHP, Ruby, PL/SQL, C, C++, Perl, Go, EBR, REST Services, Advanded Queuing, APEX, SODA, Docker

Platform Services

Cloud to On-Premise, Clustering, Microservices, Sharding, Security, High Availability, Isolation, Zero Data Loss, Administration

Analytical Services

SQL, R, Columnar In-Memory, Advanced Analytics, Machine Learning, AI

Data Support

Relational, JSON, XML, Spatial, Graph, RDF, Text, Binary. Object Stores, HDFS, Kafka, NoSQL Stores

Infrastructure Services

Public Cloud, Cloud at Customer, Exadata, BDA, ZDLRA

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Program Agenda

1. Overview and History
2. Vision
3. Data Management Strategy
4. Oracle and Modern Development
5. Open Source initiatives
6. Developer centric functionalities
7. Q & A

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

The future is:

~~NoSQL Databases~~

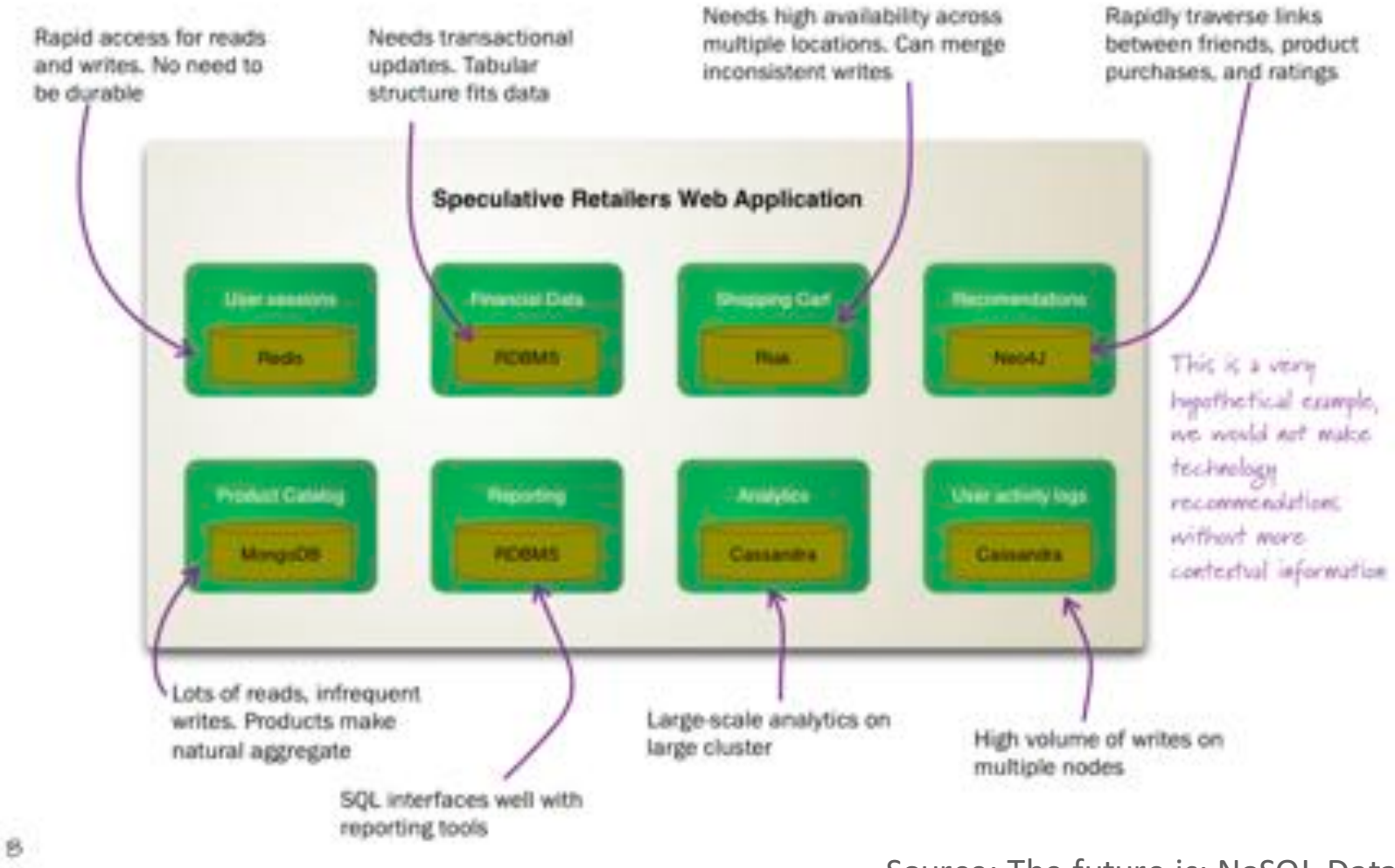
Polyglot Persistence

"Polyglot persistence will occur over the enterprise as different applications use different data storage technologies. It will also occur within a single application as different parts of an application's data store have different access characteristics."

Martin Fowler & Pramod Sadalage, Feb. 2012
http://martinfowler.com/articles/nosql-intro-original.pdf

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

what might Polyglot Persistence look like?



Source: The future is: ~~NoSQL Databases~~ Polyglot Persistence
http://martinfowler.com/articles/nosql-intro-original.pdf

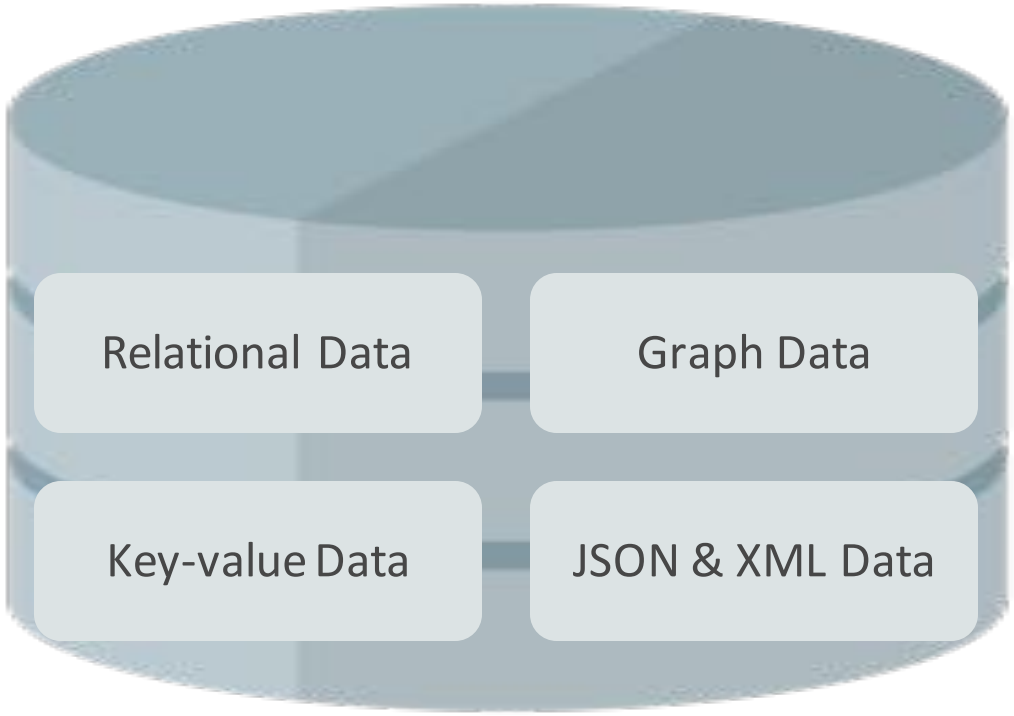
Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Two Approaches to Polyglot Persistence

Single-model

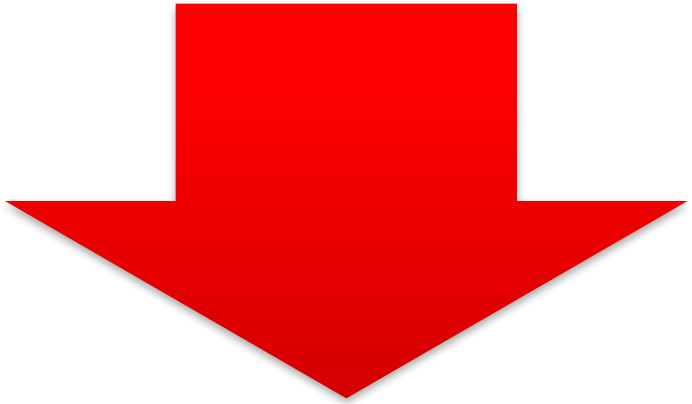
- Relational Database
- Graph Database
- Key-value Store
- JSON/XML Database

Multi-model



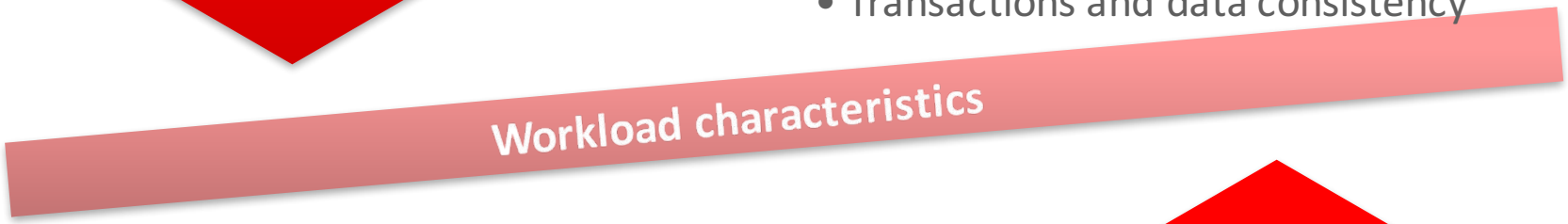
Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Considerations for Polyglot Persistence



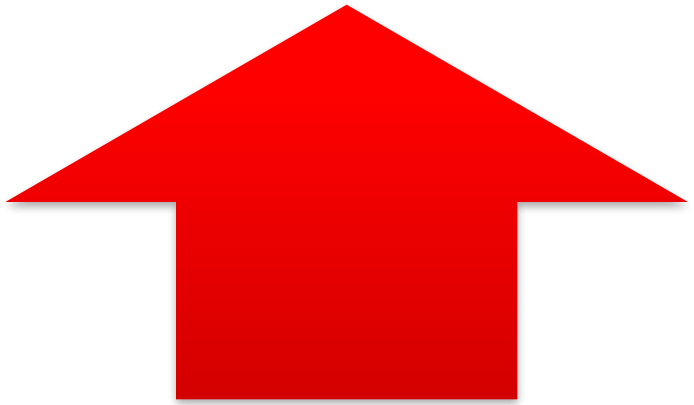
Multi-model Polyglot:

- Benefits of **consolidation and standardization**
  - Standardized administration
  - Consistent data security policies
  - Simple integration across multiple data formats
  - Transactions and data consistency



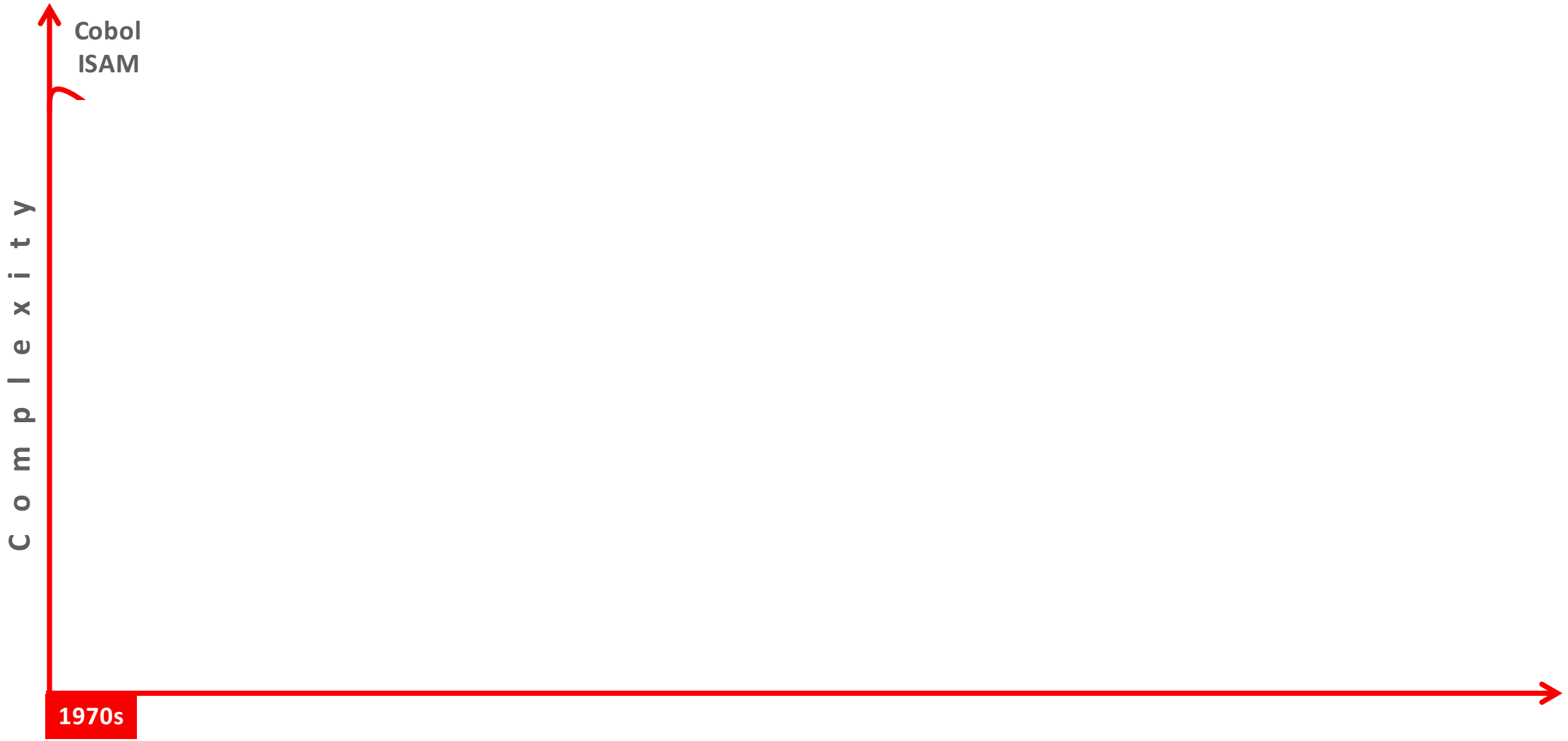
Single-model Polyglot:

- Benefits of **specialization**
  - Specialized APIs
  - Specialized data formats
  - Specialized access methods and indexes



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Evolution of data management

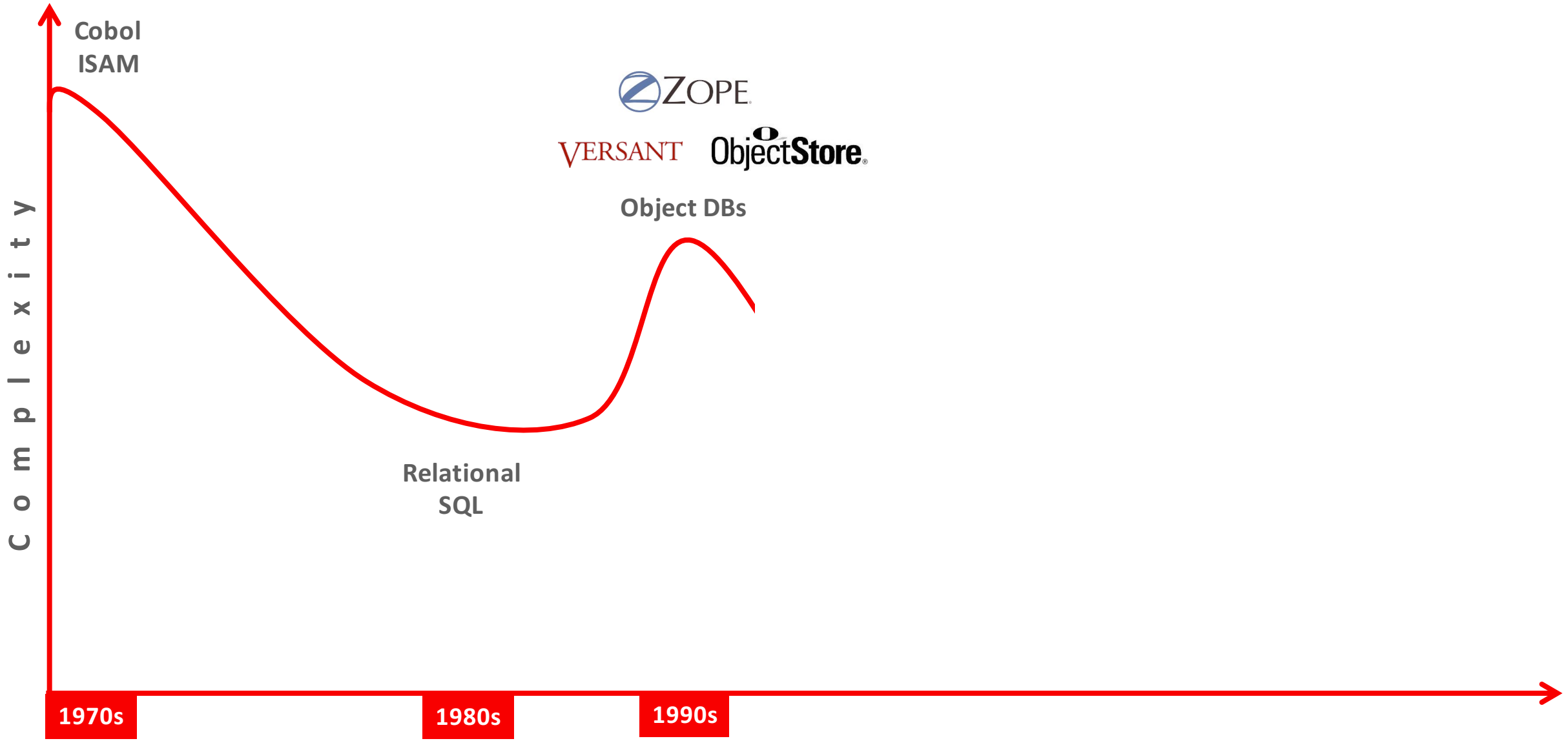


Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Evolution of data management

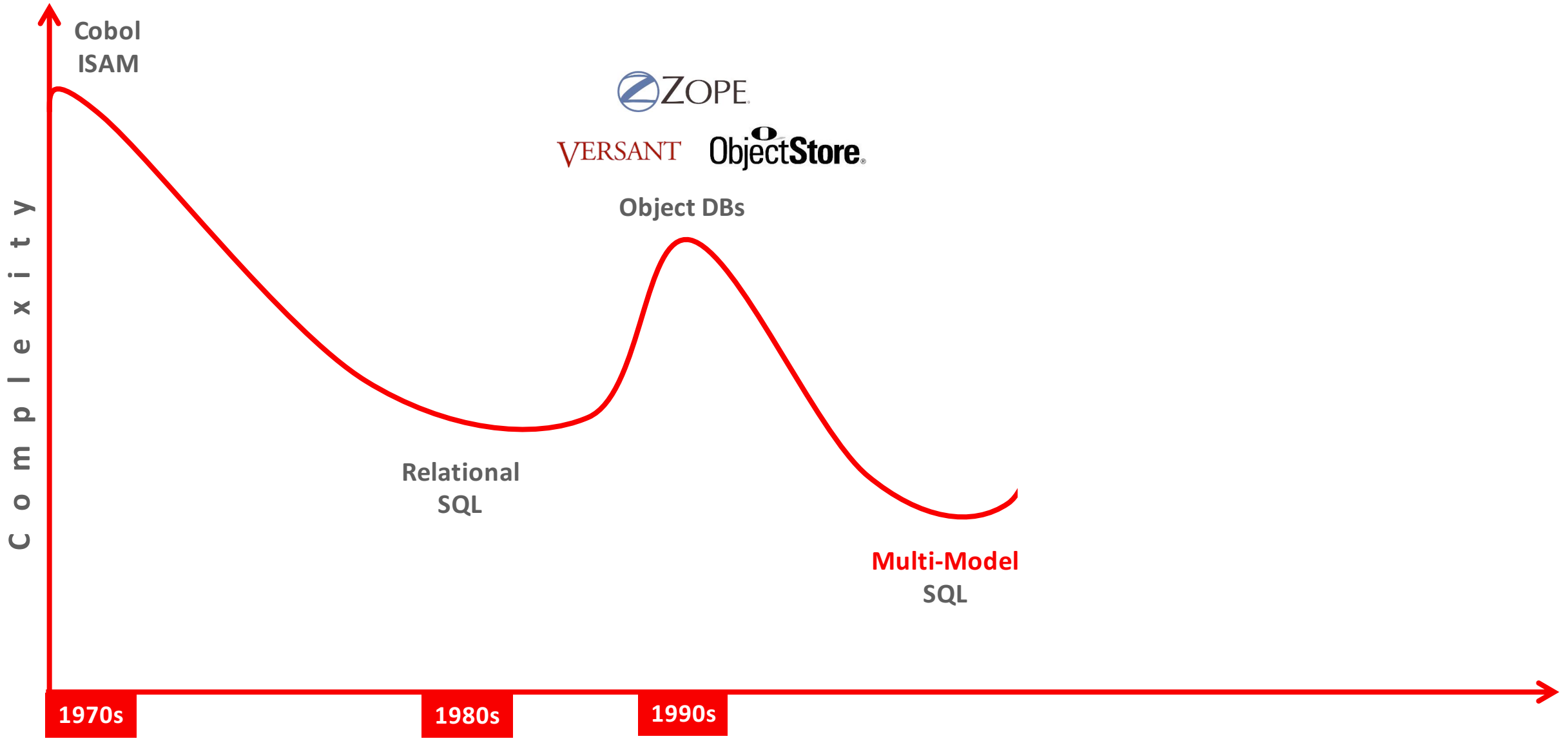
Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Evolution of data management



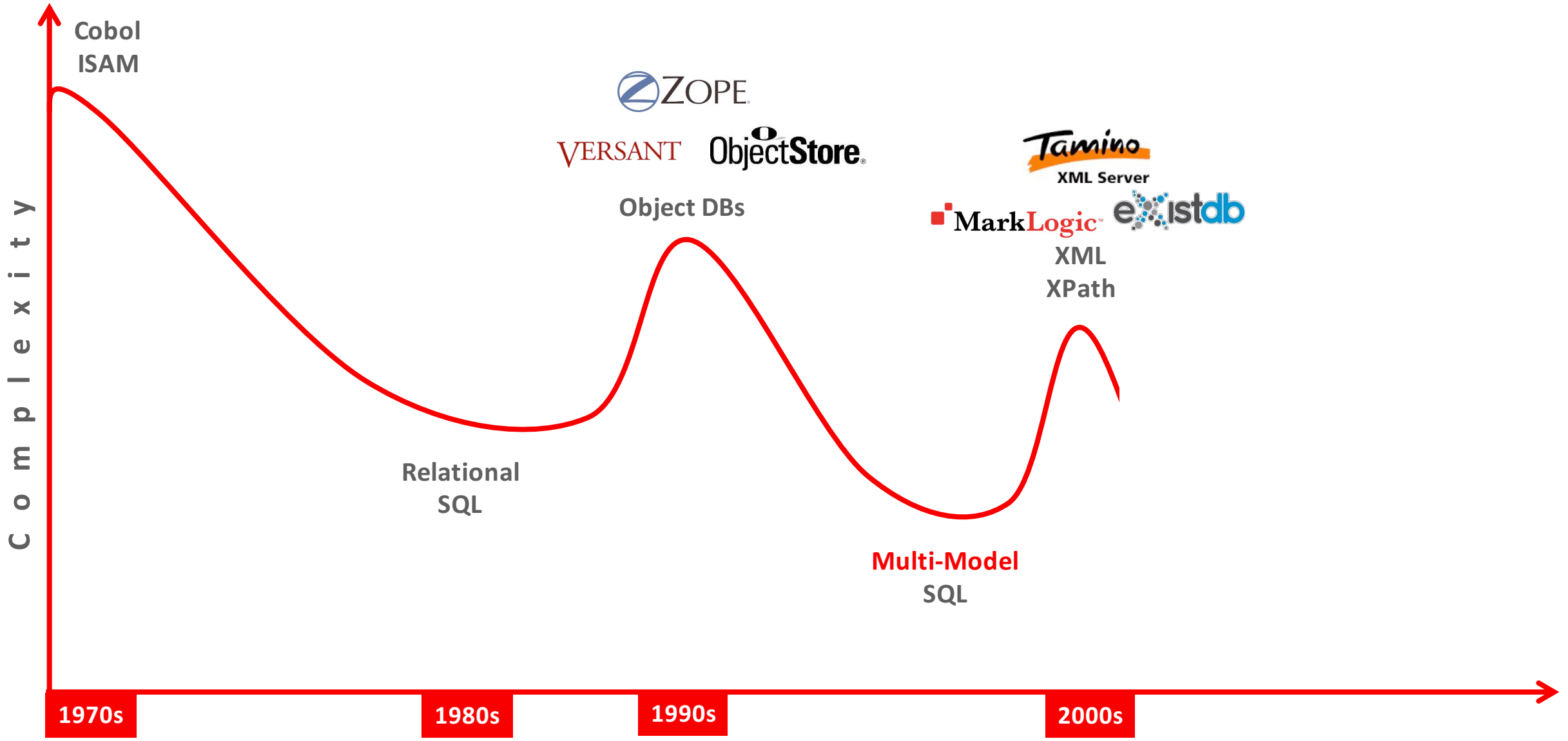
Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Evolution of data management



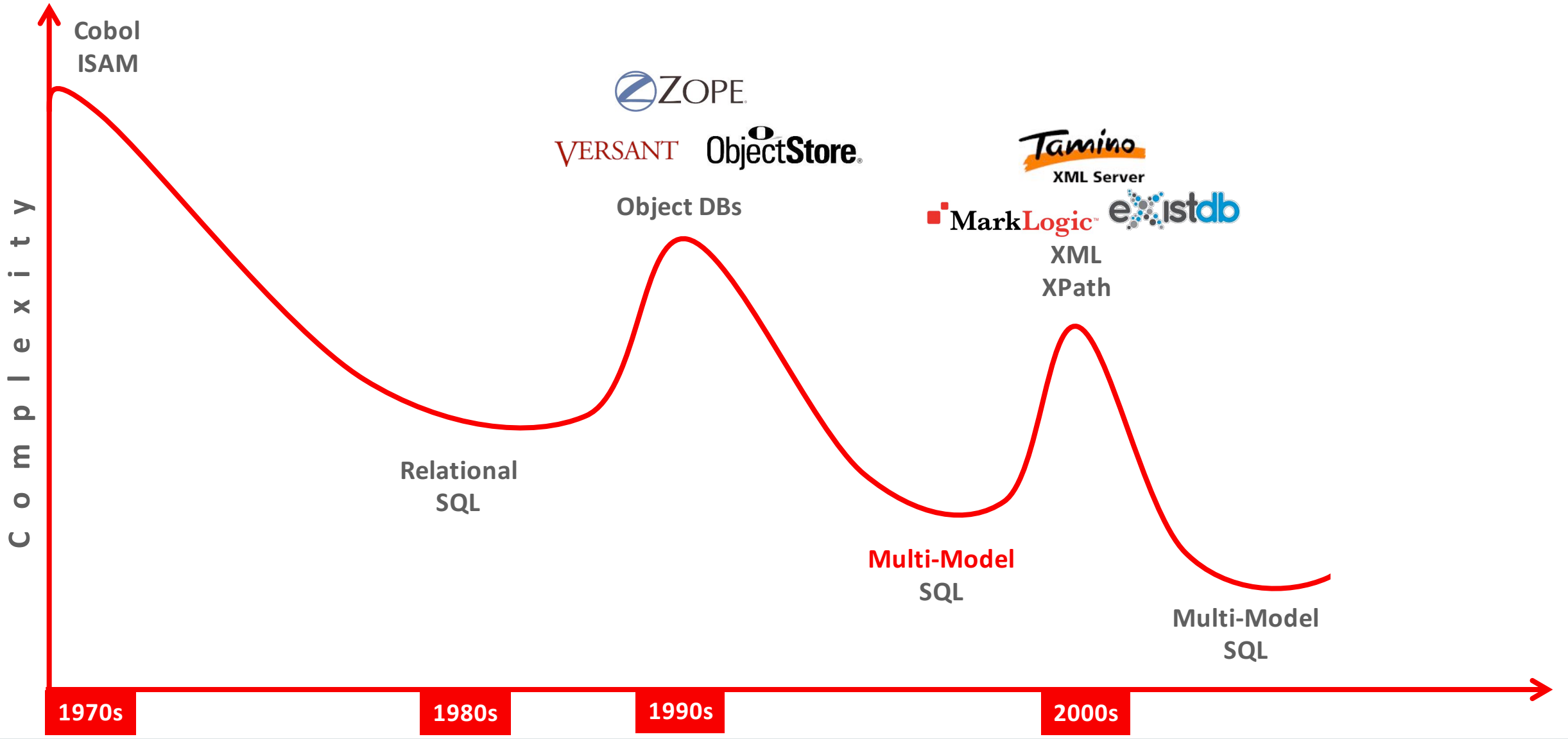
Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Evolution of data management



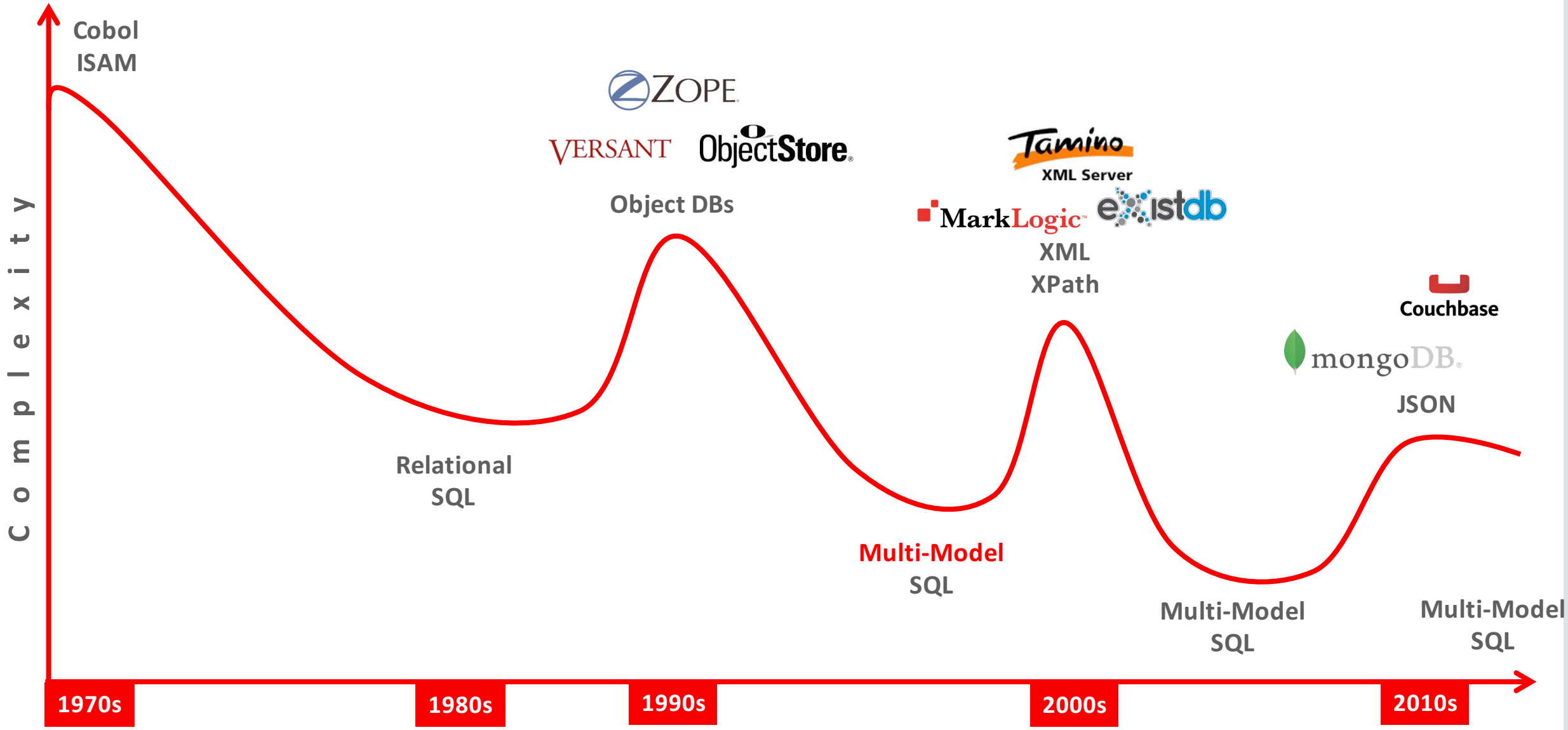
Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Evolution of data management



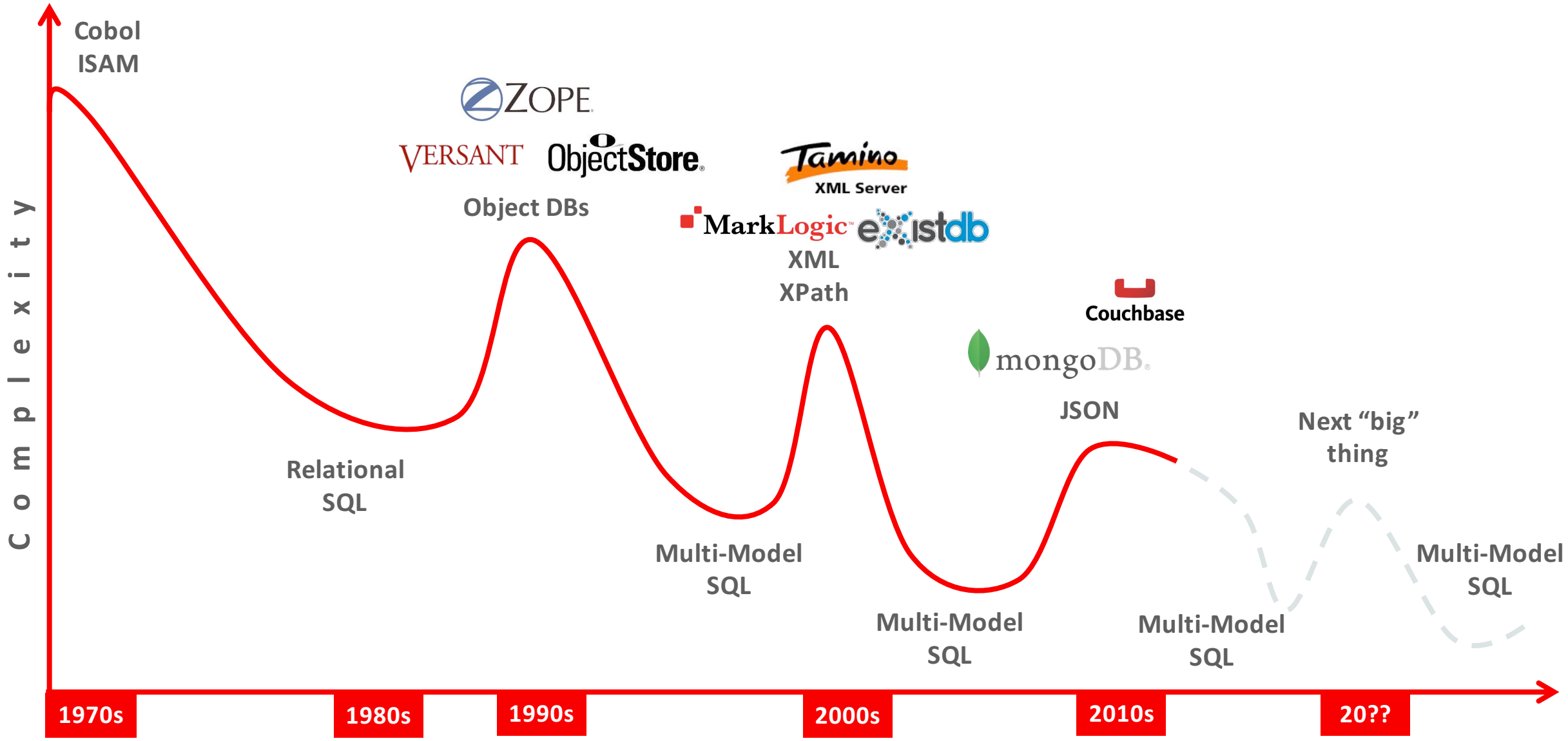
Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Evolution of data management



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Multi-model prevails over time



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Polyglot Persistence Market Trends

- Single-model architectures are most pervasive for ‘edge’ applications
  - New business & workload requirements
- Business applications naturally converge to multi-model architectures
  - Today’s ‘edge’ applications are tomorrow’s mainstream business applications
  - Efficiencies of multi-model architecture override advantages of special-purpose systems over time
- There will always be single-model polyglot architectures
  - Because there are always new ‘edge’ applications
  - Oracle’s single-model architectures:
    - Oracle Berkeley DB, Oracle NoSQL Database, Essbase, Oracle Big Data Spatial and Graph

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Oracle Product Strategy for Polyglot Persistence

Support Both – Customer chooses which one to use

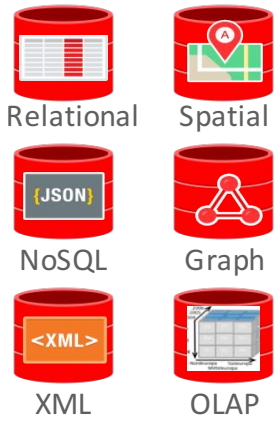
Multi-model

- Oracle Database supports multi-model persistence
  - Relational
  - XML
  - JSON
  - Text
  - Graph & Spatial

- Oracle Database provides integrated access to all database objects

Single-model

- Oracle supports multiple single-model data stores
  - Relational
  - Key/Value
  - XML
  - Spatial
  - Graph
  - OLAP



- Oracle integrates single-model polyglot environments via **Big Data SQL**

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Program Agenda

1. Overview and History
2. Vision
3. Data Management Strategy
4. Oracle and Modern Development
5. Open Source initiatives
6. Developer centric functionalities
7. Q & A

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

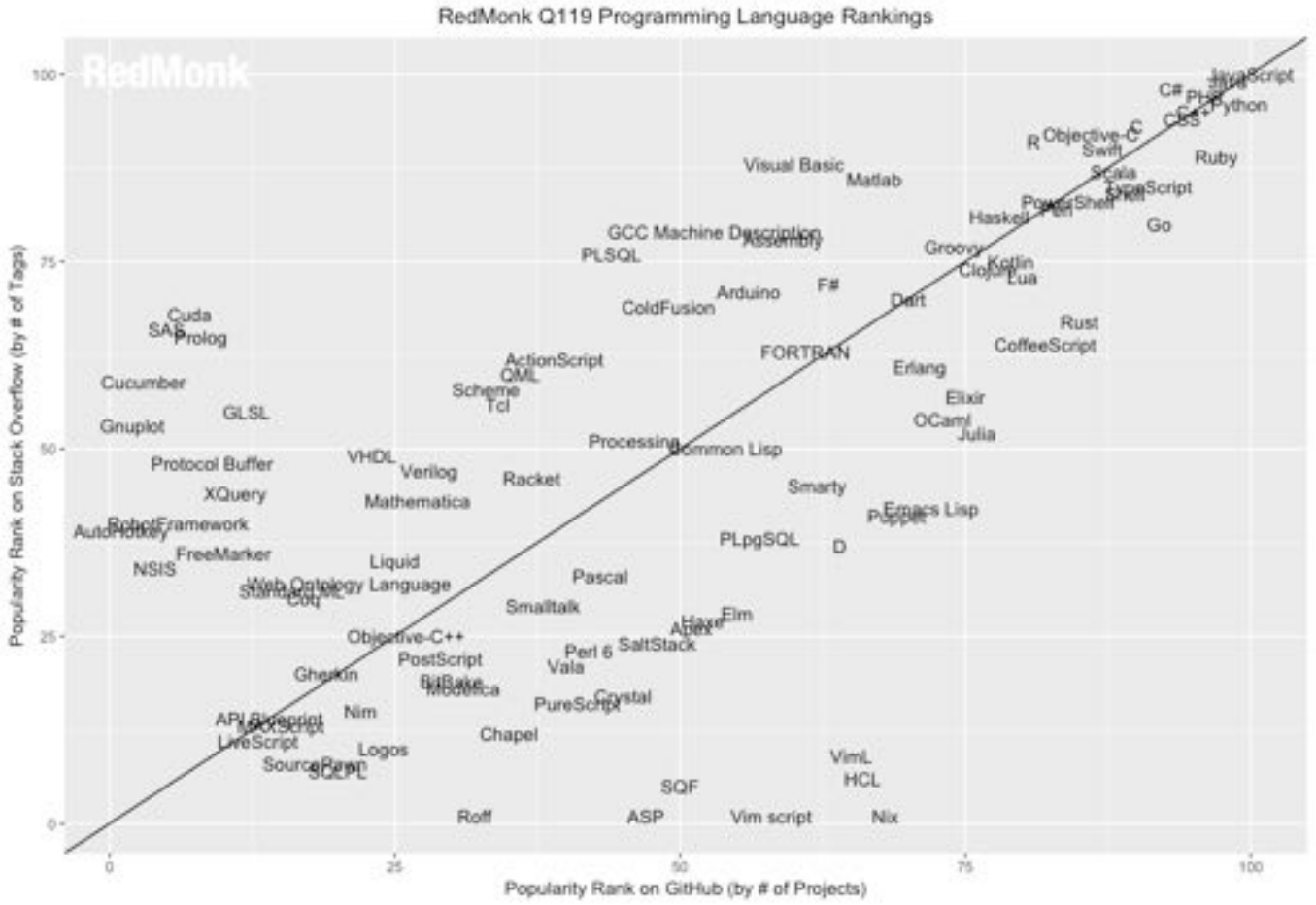
2019 most popular development languages

Stackoverflow

Programming, Scripting, and Markup Languages

All Respondents | Professional Developers

JavaScript	69.8%
HTML	68.5%
CSS	65.1%
SQL	57.0%
Java	45.3%
Bash/Shell	39.8%
Python	38.8%
C#	34.4%
PHP	30.7%
C++	25.4%
C	23.0%
TypeScript	17.4%
Ruby	10.1%
Swift	8.1%



1 JavaScript	4 PHP
2 Java	5 C#
3 Python	6 C++

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Oracle Database for the Developer

Supporting all major development environments and APIs

LANGUAGE	DRIVER
C	OCI, ODPI-C
C++	OCCI
Java	JDBC
.NET	ODP.NET
Node.js	node-oracledb
Python	cx_Oracle
PHP	OCI8, PDO_OCI
R	ROracle
Go	goracle, rana, mattn
Rust	mimir
Ruby	ruby-oci8
Perl	DBD::Oracle

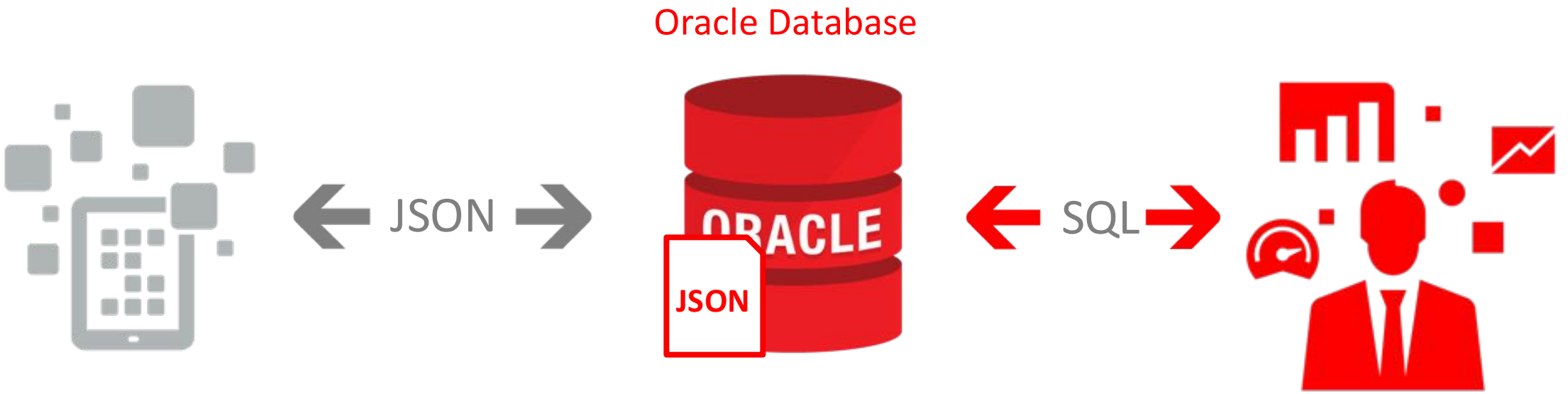
- Oracle provided Drivers
- Open Source Drivers (Oracle contributions)
- Open Source Drivers (Third-party maintainers)

… and ODBC, OLE DB, Pro*C, Pro*COBOL, Pro*Fortran, SQLJ

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

JSON Support in Oracle Database

Powerful SQL Analytics



Data accessed via RESTful service or native API's

Data persisted in database In JSON

Data analyzed via SQL

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. | Public

JSON Support in Oracle Database

Fast Application Development + Powerful SQL Access

Application developers:
Access JSON documents using REST API

```
POST /my_database/my_schema/customers  HTTP/1.0
Content-Type:  application/json
Body:
{
 "firstName":  "John",
 “lastName”:  "Smith",
 "age":  25,
 "address":  {
      "streetAddress":  "21  2nd  Street",
      "city":  "New  York",
      "state":  "NY",
      "postalCode":  "10021“,
      "isBusiness"  :  false   },
  "phoneNumbers":  [
      {"type":  "home",
       "number":  "212  555-1234“  },
      {"type":  "fax",
       "number":  "646  555-4567“  }   ]
}
```

Analytical tools and business users:
Query JSON using SQL

```
SELECT
  c.json_document.firstName,
  c.json_document.lastName,
  c.json_document.address.city
FROM customers  c;

firstName     lastName      address.city
-----------   -----------   --------------
“John”        “Smith”       “New  York”
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. | Public

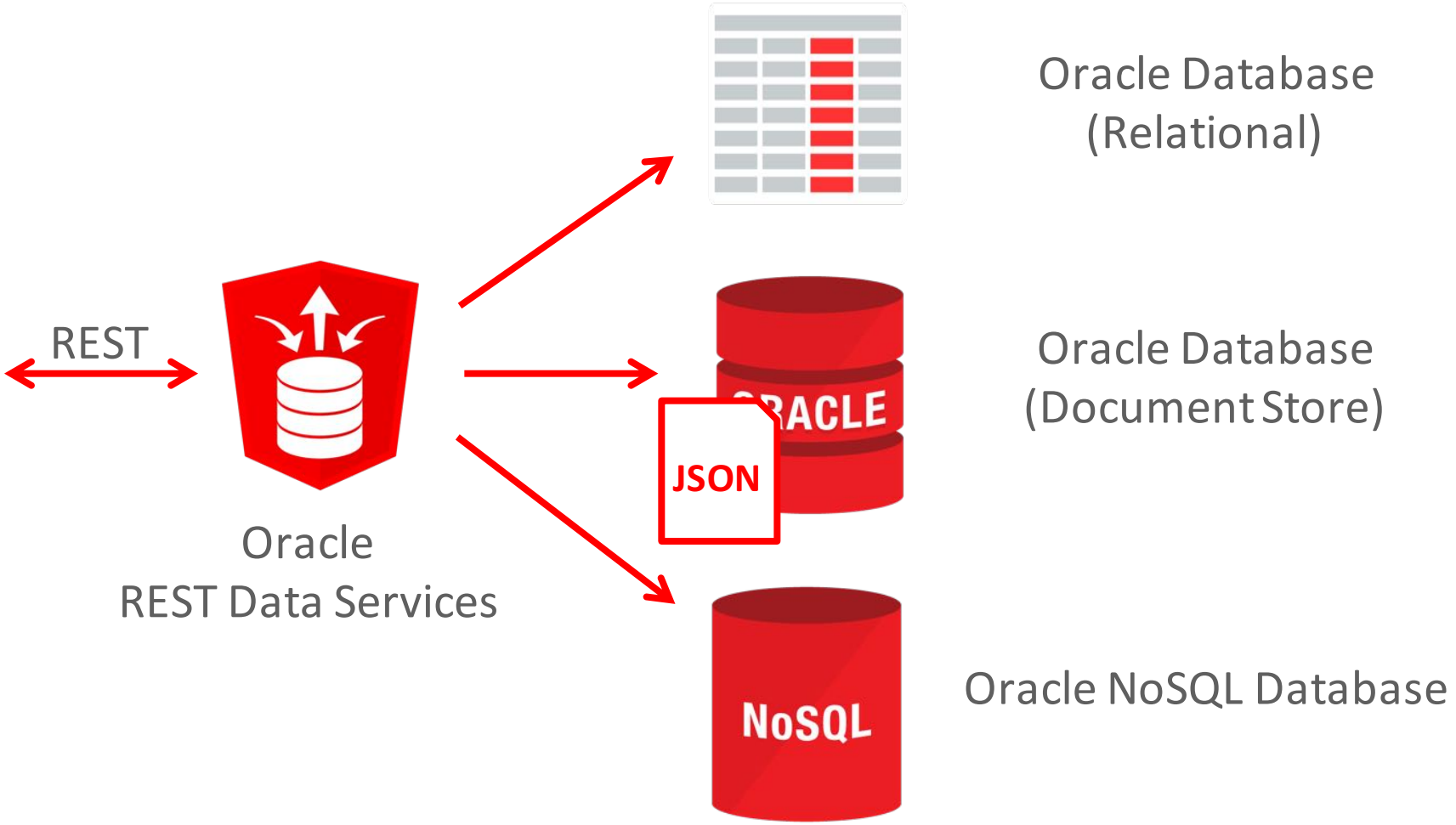
Oracle vs Mongo DB

	Oracle	Mongo DB
Document-store : Store, index, and query JSON documents	✓	✓
Simple Document-Centric API's and REST support	✓	✓
Query by Example (QBE) capability	✓	✓
Joins within Documents, Within Collections and Across Collections	✓	Within Documents Only
Joins with Relational, XML, Spatial and Text Content	✓	Limited Support for Text
Standardized Query Language	✓	
Integration with Industry Leading BI, Analytical and Reporting tools	✓	
Concurrency Control, ACID Transactions, Read Consistency	✓	-
Enterprise Backup/Recovery and Disaster Recovery	✓	
Architected for consolidation and multitenancy	✓	

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Oracle REST Data Services

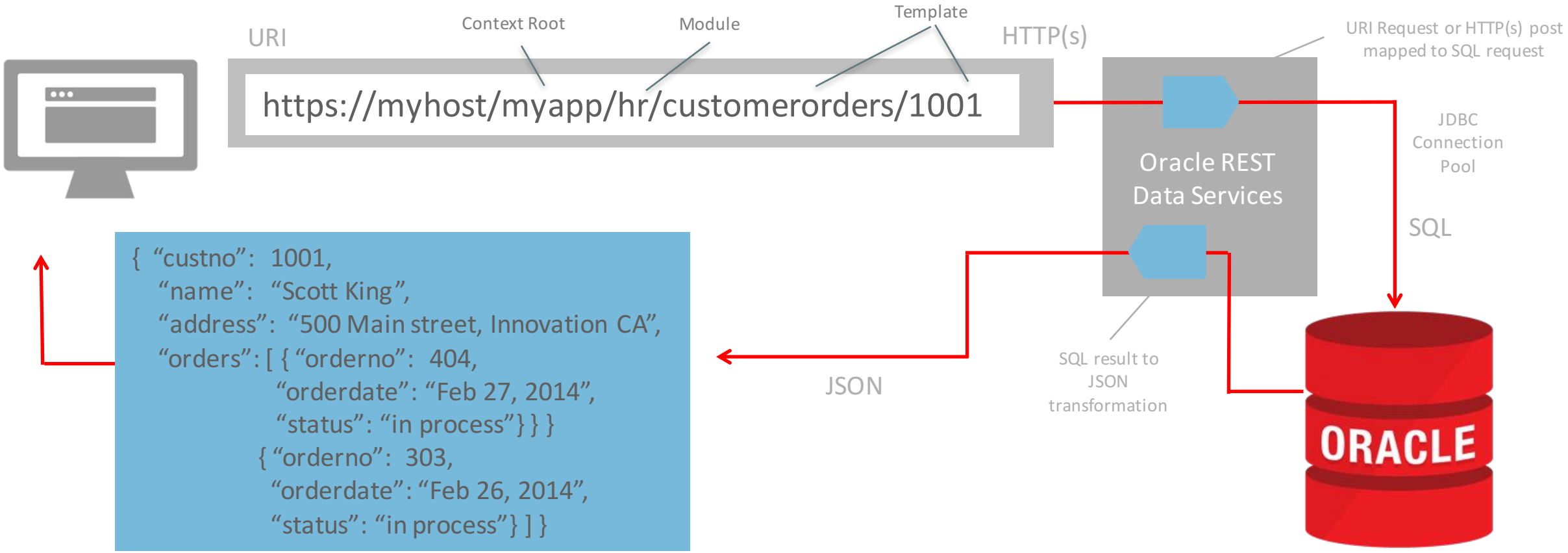
REST-enable your data



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Oracle REST Data Services

HTTP(s) API App-Dev with Relational Tables in Oracle Database



ORDS maps standard URI requests to corresponding relational SQL (not schemaless): e.g. SQL SELECT from customers and orders table.
ORDS also transforms the SQL results into the highly popular JavaScript Object Notation (JSON), other formats include HTML, binary and CSV.
Fully committed to supporting any and all standards required by Fusion / SaaS / FMW; we are actively engaged in the ongoing dialog.

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Oracle REST Data Services

Example: Query returning JSON for customer 1001

http://myhost/myapplication/custorders/simplequery { custno: 1001}

- Map URI request to data access template
- Bind custno (1001) to bindcustno
- Execute select statement below
- Note embedded cursor expression
- Set format to JSON

```
select c.*,
   cursor(
      select *
      from orders o
      where o.custno = c.custno)
     orders
    from customers c
      where c.custno = :bindcustno
```

Data Access Template

```
{ "custno": 1001,
  "name": "Scott King",
  "address": "500 Oracle Parkway, Redwood Shores, CA, 94065",
   "country": "USA",
   "class": "A",
   "orders": [ { "orderno": 303,
                 "orderdate": "Feb 26, 2014",
                 "status": "in process"},
               { "orderno": 202,
                 "orderdate": "Jan 16, 2014",
                 "status": "processed"},
               { "orderno": 101,
                 "orderdate": "Dec 2, 2013",
                 "status": "processed"} ] }
```

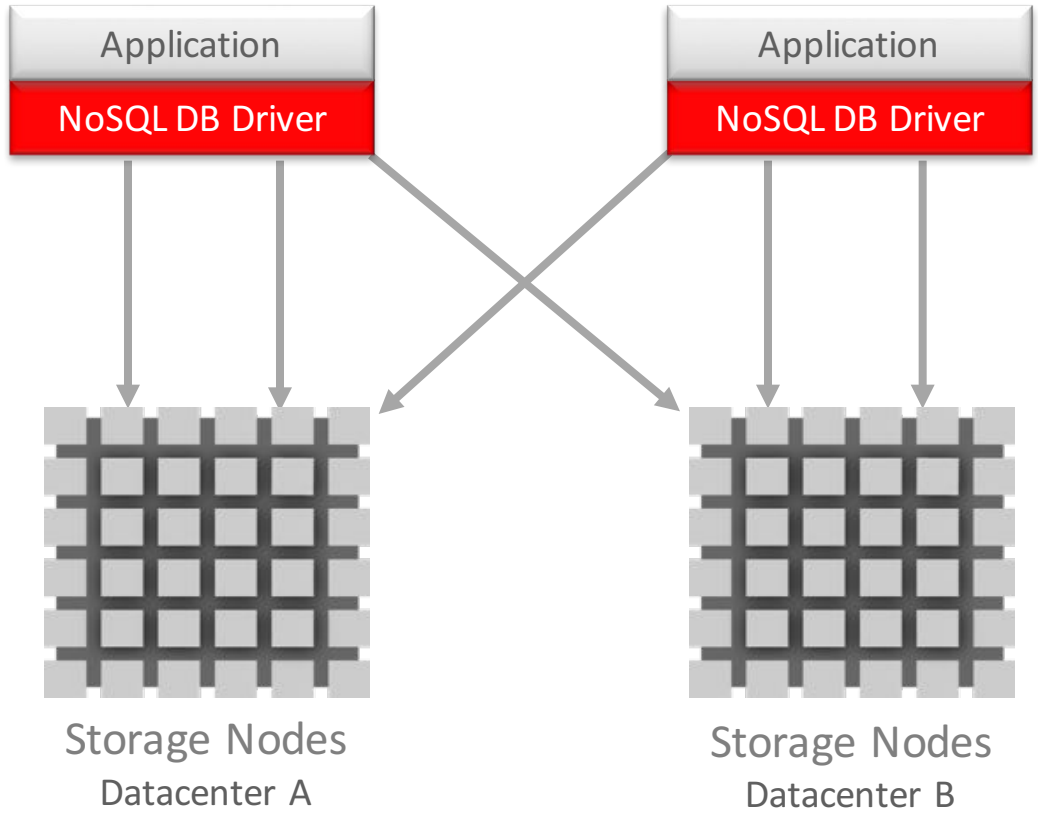
JSON Result

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. |

Oracle's commitment to Single-model Polyglot

Oracle NoSQL Database

- Developer Focus
  - BASE & ACID txns
  - Tables / JSON / Binary
  - C, Java, Python & Node.js APIs
  - Secondary Indexes
- Operations Focus
  - Elastic Configuration
  - Secure Access
  - Data Center Support
  - Online management
- Differentiating Features
  - ACID transactions
  - Online rolling upgrades
  - Streaming large object support
  - **Strong Oracle technology Integration**
  - **Engineered Systems** and Commodity HW



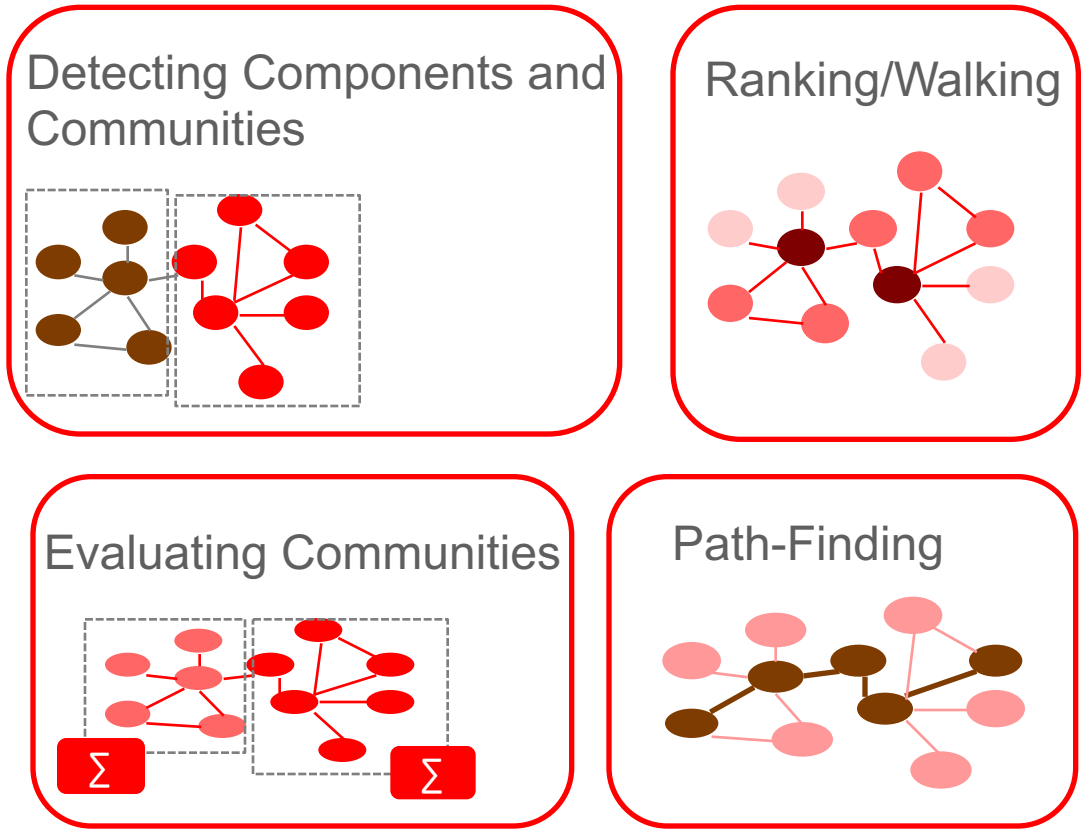
http://www.oracle.com/us/products/database/nosql/

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Oracle's commitment to Single-model Polyglot

Oracle Big Data Spatial & Graph

- Massively-Scalable Graph Database
  - Scales to **trillions** of edges
  - Apache HBase
  - Oracle NoSQL Database
- In-Memory Graph Analytics
  - More than 30 graph analysis algorithms
- Simple, standard interfaces
  - Java
  - Tinkerpop: Blueprints, Gremlin, Rexster
  - Python

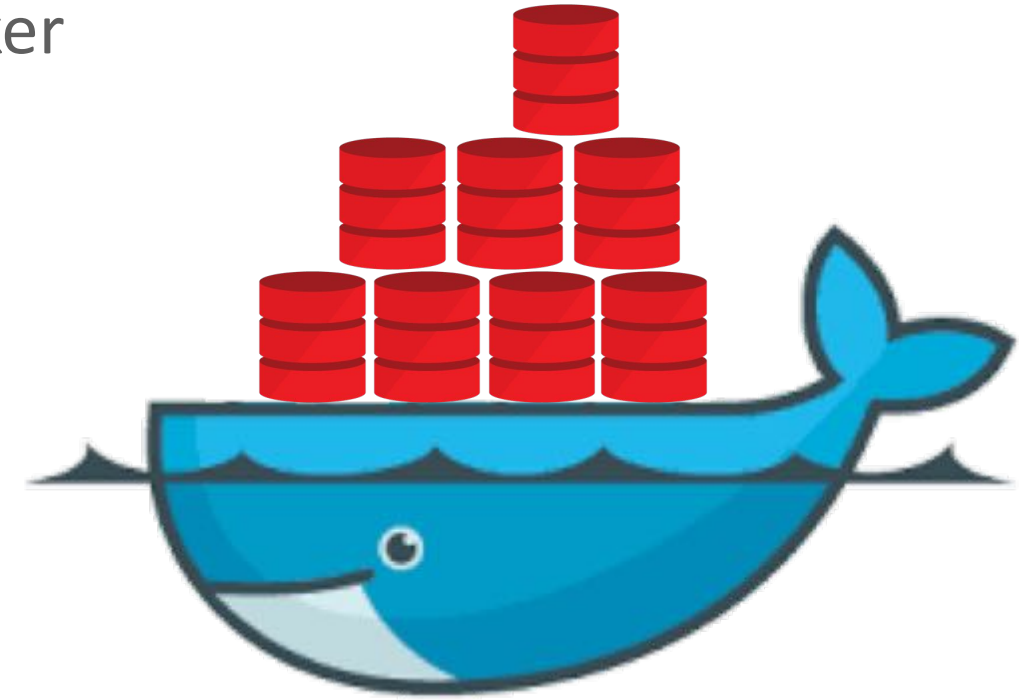


www.oracle.com/database/big-data-spatial-and-graph

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Oracle on Docker

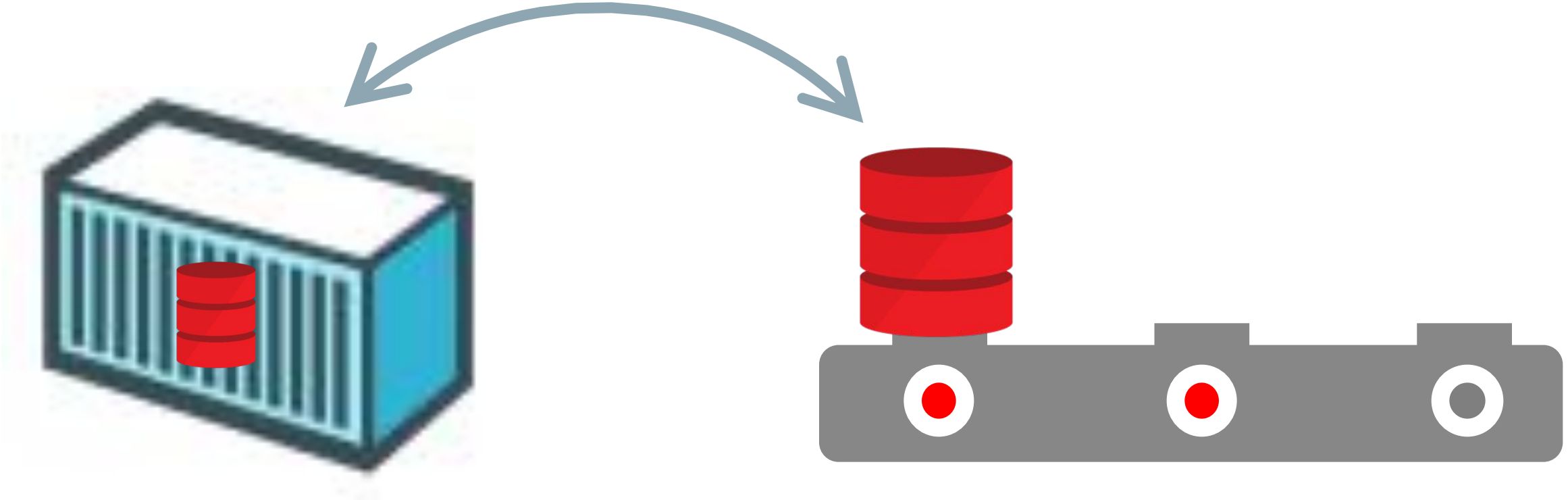
- Oracle Database is fully supported on Docker
  - Oracle Linux 7
  - Red Hat Enterprise Linux 7
- Oracle image on Docker Store
- Docker build files on GitHub



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Oracle on Docker

- Docker container contains single-PDB CDB
- PDB can be plugged, unplugged, etc.
- PDB can move bi-directional



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Docker Store

- Oracle 12.1 & 12.2 images are available on Docker Store Registry
  - https://store.docker.com

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Docker build files available on GitHub

- Repository: https://github.com/oracle/docker-images
- Build files for 18c, 12.2, 12.1, 11.2.0.2 XE

README.md

Docker Images from Oracle

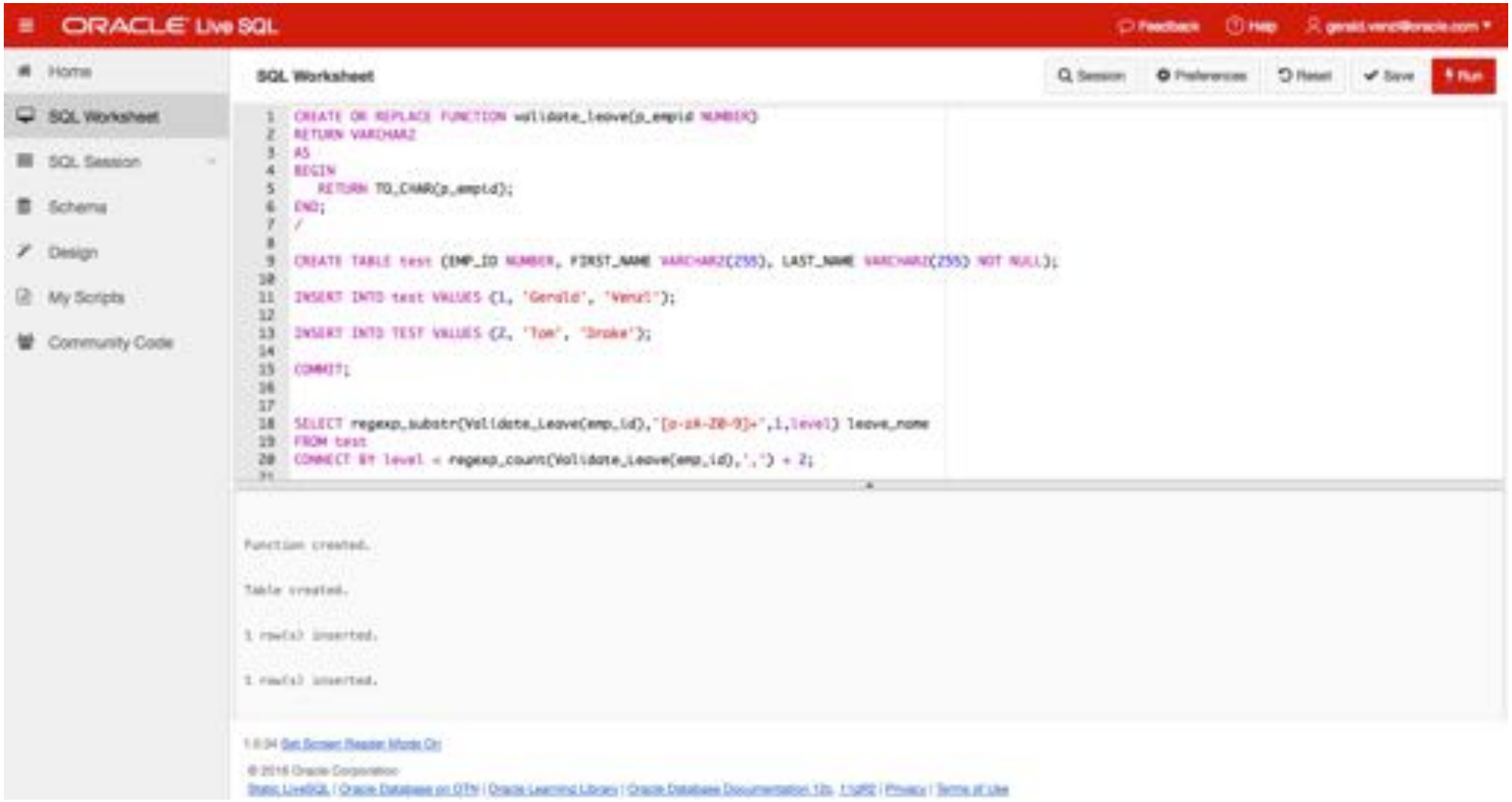
This repository stores Dockerfiles and samples to build Docker images for Oracle products and Open Source projects.

- Oracle Coherence
- Oracle Database
- Oracle Java
- Oracle HTTP Server

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

LiveSQL.oracle.com

The full power of Oracle SQL in your browser



Copyright © 2014 Oracle and/or its affiliates. All rights reserved. |

Program Agenda

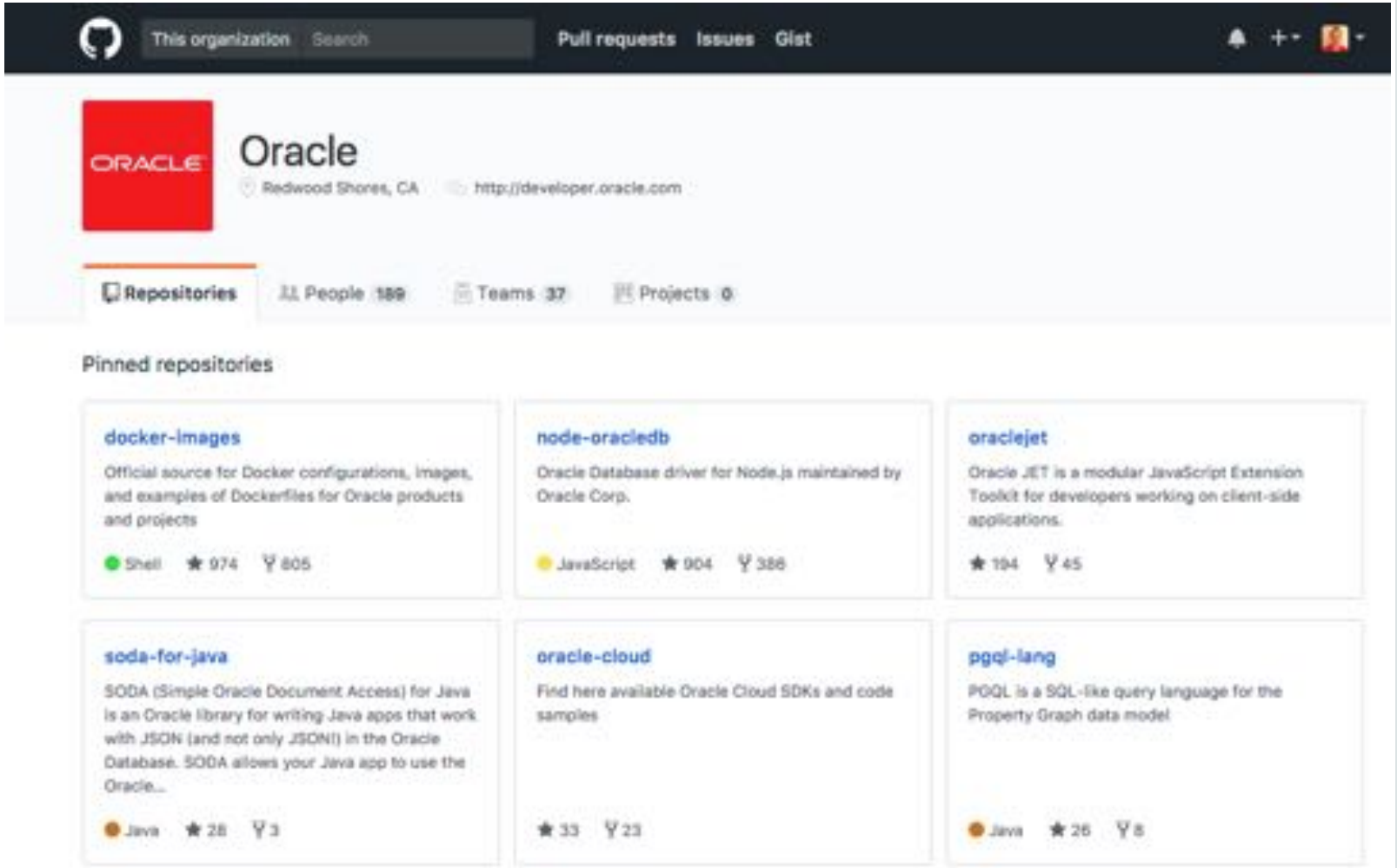
1. Overview and History
2. Vision
3. Data Management Strategy
4. Oracle and Modern Development
5. Open Source initiatives
6. Developer centric functionalities
7. Q & A

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Oracle On GitHub

www.github.com/oracle

- Official Oracle representation on GitHub
- Examples and tools for Docker, Java, SQL, Python, Node.js, PL/SQL
- Repos regularly added
- Main source for Open Source components



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted

Introducing Simple Oracle Document Access (SODA)

- An abstract API definition for
  - Collection Management: Ability to create and drop collections
  - Create, Retrieve, Update and Delete (CRUD) operations on documents
  - List operations on collections
  - Query-by-Example (QBE) for searching collections
  - Utility and control functions
    - Create and Drop Indexes
    - Bulk Insert
- Implementations currently available for JAVA and REST
- Support for NODE.js and other languages forthcoming

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. |

SODA for Java

- SODA implementation for the Java Developer
- Developers can store JSON documents in the Oracle Database without learning JDBC or SQL
- Uses a standard JDBC connection to talk to the database
- Supports transactions
- Enables hybrid application development
  - Mix and Match SODA and JDBC based operations in a single application

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. |

node-oracledb

for Node.js

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. |

node-oracledb

- A simple, stable Oracle Database driver with out-of-the box performance
- Ongoing contributions from Oracle
  - Support for latest Oracle Database features
  - 26 releases since January 2015
- Modular design
  - Underlying, simple DB access layer based on OCI

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. |

node-oracledb

- **Open source** development, release and support under Apache 2.0 license
  - GitHub repository (www.github.com/oracle/node-oracledb)
  - Installable from NPM registry (www.npmjs.com/package/oracledb)
  - Approx. monthly release cycle

Users can contribute under the Oracle Contributor Agreement.

Thanks to all who have contributed code, documentation and ideas

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. |

cx_Oracle

for Python

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. |

cx_Oracle

- A simple, stable Oracle Database driver with out-of-the box performance
- Ongoing contributions from Oracle
  - Support for latest Oracle Database features
- Modular design
  - Underlying, simple DB access layer based on OCI

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. |

cx_Oracle

- **Open source** development, release and support under BSD license
  - GitHub repository (https://github.com/oracle/python-cx_Oracle)
  - Installable from NPM registry (https://oracle.github.io/python-cx_Oracle/)
  - Approx. monthly release cycle

Users can contribute under the Oracle Contributor Agreement.

Thanks to all who have contributed code, documentation and ideas

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. |

Program Agenda

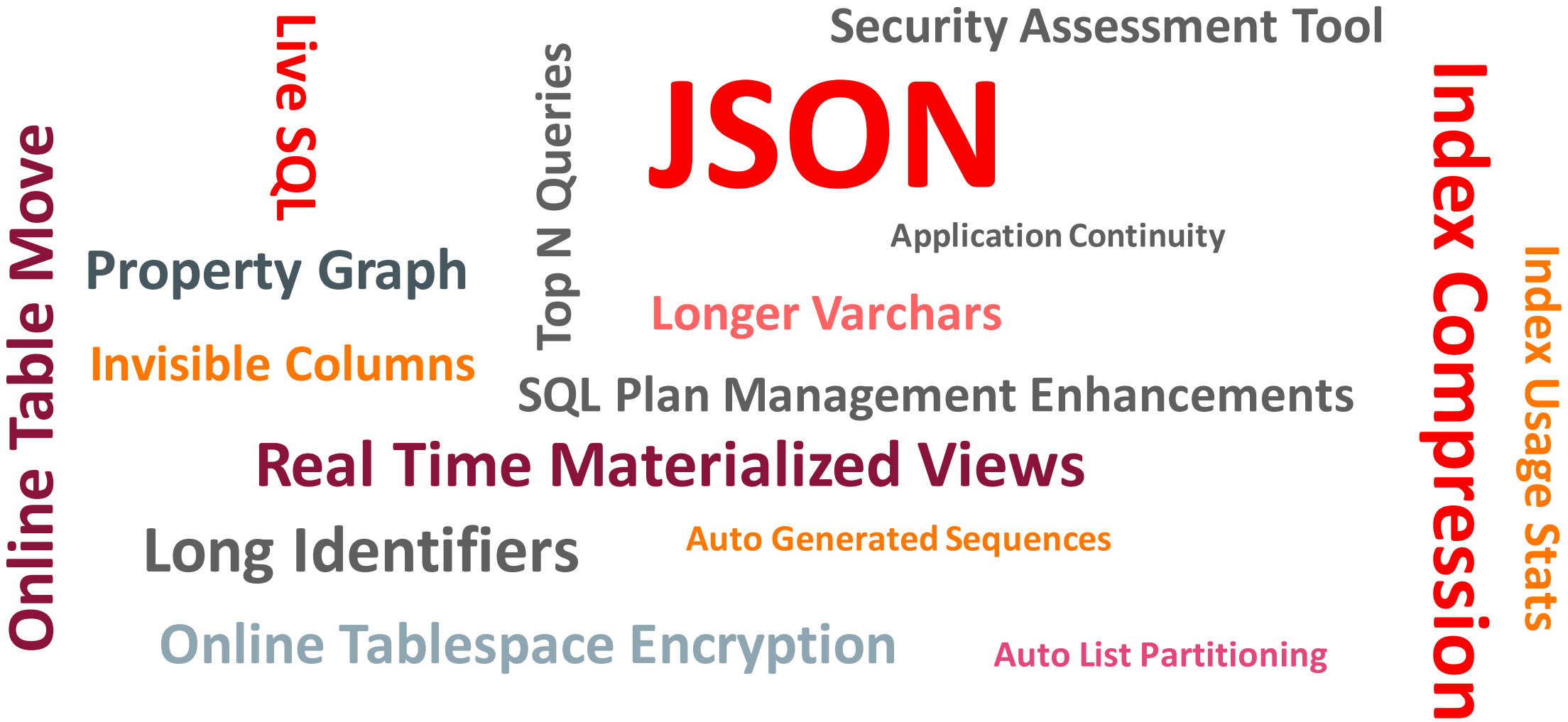
1. Overview and History
2. Vision
3. Data Management Strategy
4. Oracle and Modern Development
5. Open Source initiatives
6. Developer centric functionalities
7. Q & A

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Overall

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. Public

Oracle Database 12c “Under the Radar” Features



Copyright © 2014 Oracle and/or its affiliates. All rights reserved. |

NEW IN 12.2

128-byte identifiers for objects

```
CREATE TABLE VERY_VERY_LONG_TABLE_NAME_IDENTIFIER_THAT_IS_58_BYTES_LONG
(
  VERY_VERY_LONG_TEXT_COLUMN_WITH_DATA_TYPE_VARCHAR2_THAT_IS_72_BYTES_LONG VARCHAR2(25)
);

Table VERY_VERY_LONG_TABLE_NAME_IDENTIFIER_THAT_IS_58_BYTES_LONG created.

INSERT INTO VERY_VERY_LONG_TABLE_NAME_IDENTIFIER_THAT_IS_58_BYTES_LONG
   VALUES ('Hello World!');

1 row inserted.

SELECT * FROM VERY_VERY_LONG_TABLE_NAME_IDENTIFIER_THAT_IS_58_BYTES_LONG;

VERY_VERY_LONG_TEXT_COLUM
-------------------------
Hello World!
```

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

Case-insensitive Database and Column-level Collation

Greatly simplifies migration of case-insensitive functionality of 3rd-party products

```
CREATE TABLE product
( id        NUMBER,
  name      VARCHAR2(50) COLLATE BINARY_CI,
  comments  VARCHAR2(500)
) DEFAULT COLLATION BINARY;

SELECT name, comments FROM product
 WHERE name LIKE '%BASE%' OR
    comments COLLATE BINARY_CI LIKE '%REPORT%';

NAME                          COMMENTS
Oracle Database
Activity-Based Management
Business Intelligence         Replaces Reports
```

_CI = case-insensitive

Inherits BINARY

- Linguistic-sensitive operations, e.g., comparison and sorting, on the column honor the declared collation
- Unspecified column collation is inherited from the default collation property of the parent table or schema
- COLLATE operator can be used to cast an explicit collation anywhere in an expression

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

PL/SQL deprecate pragma

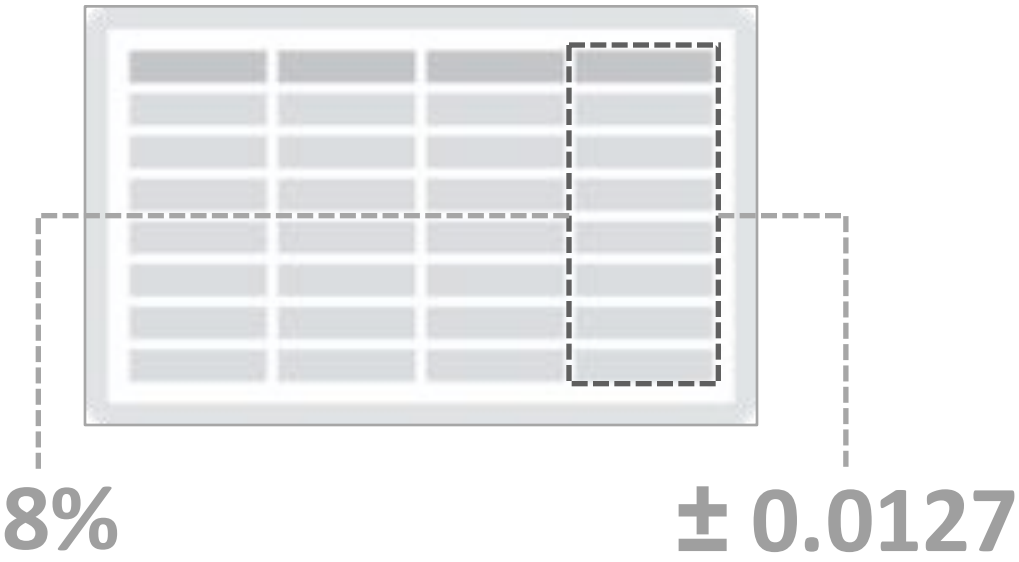
```
CREATE PROCEDURE p AUTHID DEFINER IS
  PRAGMA DEPRECATE (p, 'p is deprecated. You must use p2 instead.');
BEGIN
  DBMS_Output.Put_Line('p');
END p;

PLW-06019: entity P is deprecated

CREATE PROCEDURE q authid Definer is
BEGIN
  p();
  DBMS_Output.Put_Line('q');
END q;

PLW-06020:
reference to a deprecated entity: p is deprecated. You must use p2 instead.
```

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

New in 12.2 Approximate Query Processing

Delivers significantly **faster** analysis for **interactive** and highly **iterative** data exploration



- Approximations for expensive aggregate calculations:
  - **APPROX_COUNT_DISTINCT (12.1)**
  - **APPROX_PERCENTILE**
  - **APPROX_MEDIAN**
  - 6-13X faster, accuracy typically within < 1%
- Use with **ZERO code changes**
  - approx_for_aggregation = TRUE
- Accuracy and error rate provided

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Top-N approximate aggregation

Interactive response times against terabytes of data

NEW IN 18c

- Approximate results for common top-N queries
  - How many approximate page views did the top five blog posts get last week?
  - What were the top 50 customers in each region and their approximate spending?
- Order of magnitude faster processing with high accuracy (error rate < 0.5%)
- New approximate functions APPROX_COUNT(), APPROX_SUM(), APPROX_RANK()

Top 5 blogs with approximate hits

```
SELECT blog_post, APPROX_COUNT(*)
FROM weblog
GROUP BY blog_post
FETCH FIRST 5 ROWS ONLY;
```

Top 50 customers per region with approximate spending

```
SELECT region, customer_name,
       APPROX_RANK(PARTITION BY region
                   ORDER BY APPROX_SUM(sales) DESC) appr_rank,
       APPROX_SUM(sales) appr_sales
FROM sales_transactions
GROUP BY region, customer_name
HAVING APPROX_RANK(..) <=50;
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. Oracle Confidential – Internal

Inline external tables

Transparently access external data

NEW IN 18c

- External table definition provided at runtime
  - Similar to inline view
- No need to pre-create external tables that are used one time only
  - Increased developer productivity

```
CREATE TABLE sales_xt
    (prod_id number, … )
     TYPE ORACLE_LOADER
     …
     LOCATION 'new_sales_kw13')
     REJECT LIMIT UNLIMITED );

INSERT INTO sales SELECT * FROM
sales_xt;

DROP TABLE sales_xt;
```

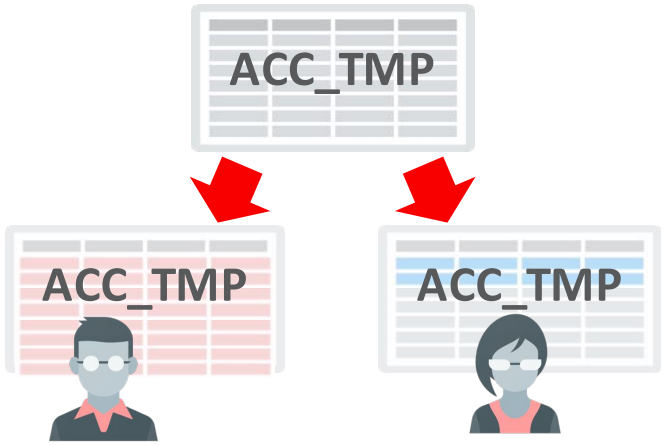
```
INSERT INTO sales
SELECT sales_xt.*
FROM EXTERNAL(
      (prod_id number, … )
       TYPE ORACLE_LOADER
       …
       LOCATION 'new_sales_kw13')
       REJECT LIMIT UNLIMITED );
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. Oracle Confidential – Internal

Private temporary tables

transient tables useful for reporting applications

NEW IN 18c



Global temporary tables

- Persistent, shared (global) table definition
- Temporary, private (session-based) data content
  - Data physically exists for a transaction or session
  - Session-private statistics

Private temporary tables (18c)

- Temporary, private (session-based) table definition
  - Private table name and shape
- Temporary, private (session-based) data content
  - Session or transaction duration

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. |

NEW IN 12.1

Oracle Database 12 Temporal Support

Transaction Time Temporal (Flashback Data Archive)

- Tracks transactional changes to a table over its lifetime
- Typically used for compliance and auditing
- Enables the users to see the data as it was at a point in time in the past

Valid Time Temporal

- Enables user to model & query data for “real world validity”
- Typically used for insurance policies, financial markets, trade data & future changes
- Users can model concepts such as the “Life time of an insurance policy”

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. |

Valid Time Temporal
Example

NEW IN 12.1

```
CREATE TABLE customers(
  custid NUMBER,
  custname VARCHAR2(30),
  custaddr1 VARCHAR2(50),
  custaddr2 VARCHAR2(50),
  custcity VARCHAR2(50),
  coststate VARCHAR2(2),
  custzip VARCHAR2(20),
  start_time TIMESTAMP,
  end_time TIMESTAMP,
PERIOD FOR cust_valid_time (start_time, end_time));
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Valid Time Temporal
Example

NEW IN 12.1

custid	custname	custaddr1	custaddr2	custcity	coststate	custzip	start_time	end_time
1	Acme Inc	123 Any Street	Suite 17	Anytown	CA	99999	01-JAN-15	

```
INSERT INTO CUSTOMERS VALUES(1,'Acme Inc.','123 Any
Street','Suite 17','Anytown','AS','99999', TO_TIMESTAMP('01-JAN-
15') ,NULL);
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Valid Time Temporal
Example

NEW IN 12.1

custid	custname	custaddr1	custaddr2	custcity	coststate	custzip	start_time	end_time
1	Acme Inc	123 Any Street	Suite 17	Anytown	CA	99999	01-JAN-15	31-MAY-15

```
UPDATE customers
SET end_time = TO_TIMESTAMP('31-MAY-15')
WHERE custid = 1;
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Valid Time Temporal
Example

NEW IN 12.1

custid	custname	custaddr1	custaddr2	custcity	coststate	custzip	start_time	end_time
1	Acme Inc	123 Any Street	Suite 17	Anytown	CA	99999	01-JAN-15	31-MAY-15
1	Acme Inc	456 Another Street		Anytown	CA	99998	01-JUN-15	

```
INSERT INTO CUSTOMERS VALUES(1,'Acme Inc.',’456 Another Street',
NULL,'Anytown','AS','99998', TO_TIMESTAMP('01-JUN-15’) ,NULL);
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Valid Time Temporal
Example

NEW IN 12.1

custid	custname	custaddr1	custaddr2	custcity	coststate	custzip	start_time	end_time
1	Acme Inc	123 Any Street	Suite 17	Anytown	CA	99999	01-JAN-15	31-MAY-15
1	Acme Inc	456 Another Street		Anytown	CA	99998	01-JUN-15	

```
SELECT custaddr1, custaddr2, custcity, coststate, custzip
FROM customers WHERE custid = 1;
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Valid Time Temporal
Example

NEW IN 12.1

custid	custname	custaddr1	custaddr2	custcity	coststate	custzip	start_time	end_time
1	Acme Inc	123 Any Street	Suite 17	Anytown	CA	99999	01-JAN-15	31-MAY-15
1	Acme Inc	456 Another Street		Anytown	CA	99998	01-JUN-15	

```
EXEC DBMS_FLASHBACK_ARCHIVE.ENABLE_AT_VALID_TIME('CURRENT');

SELECT custid, start_time, end_time
FROM customers WHERE custid=1;
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Valid Time Temporal
Example

NEW IN 12.1

custid	custname	custaddr1	custaddr2	custcity	coststate	custzip	start_time	end_time
1	Acme Inc	123 Any Street	Suite 17	Anytown	CA	99999	01-JAN-15	31-MAY-15
1	Acme Inc	456 Another Street		Anytown	CA	99998	01-JUN-15	

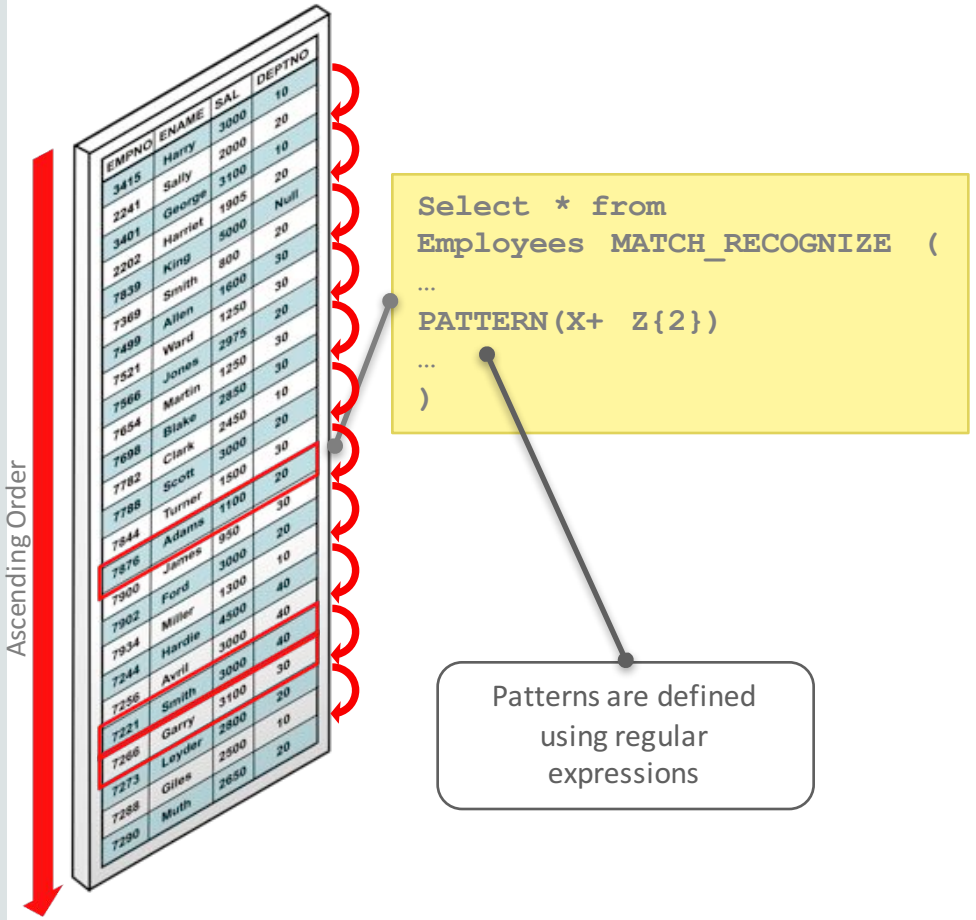
```
SELECT custid, start_time, end_time
FROM customers
AS OF PERIOD FOR cust_valid_time TO_TIMESTAMP('03-JUN-15');
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

SQL Pattern Matching

Simplified Analysis of Data

NEW IN 12.1



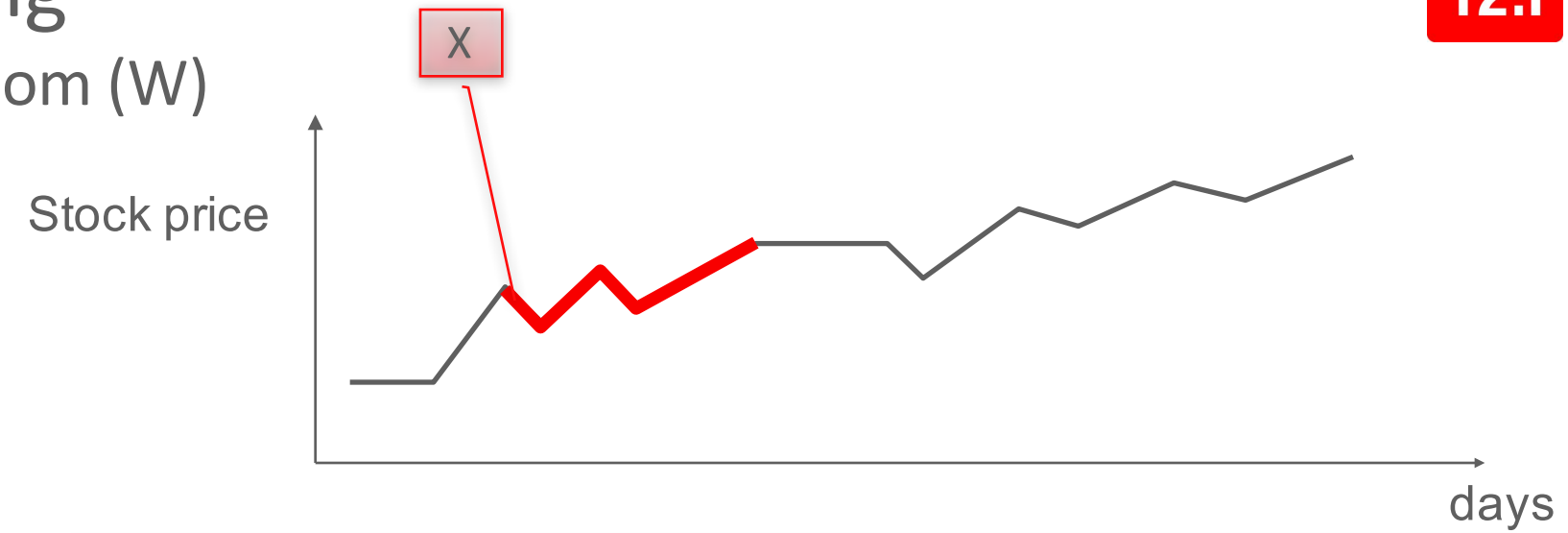
- Scalable discovery of business event sequences
  - Clickstream logs: sessionization, search behaviour
  - Financial transactions: fraud detection, double bottom (“W”) stock analysis
  - Telco: dropped calls
  - Medical sensors: automated medical observations and detections

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

SQL Pattern Matching

Example: Find Double Bottom (W)

NEW IN 12.1



- Find double bottom (W) patterns and report:
  - Beginning and ending date of the pattern
  - Average Price Increase in the second ascent
  - Modify the search to find only patterns that lasted less than a week

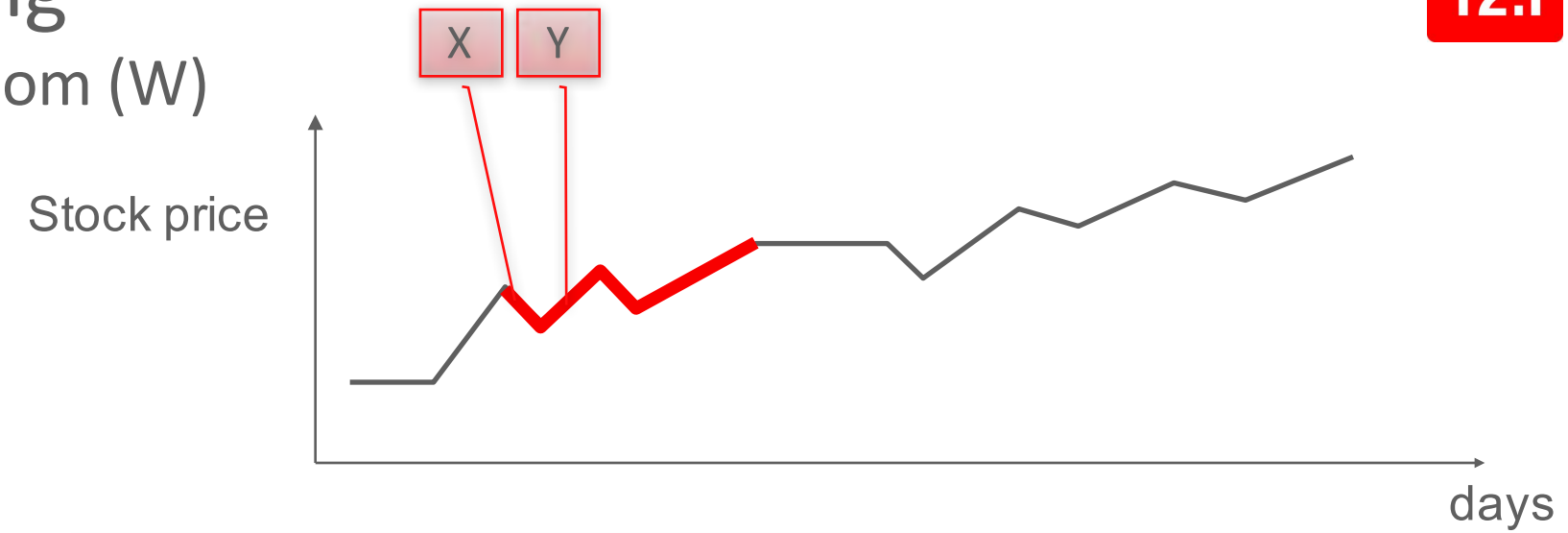
```
PATTERN (X+ Y+ W+ Z+)
DEFINE X AS (price < PREV(price))
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

SQL Pattern Matching

Example: Find Double Bottom (W)

NEW IN 12.1



- Find double bottom (W) patterns and report:
  - Beginning and ending date of the pattern
  - Average Price Increase in the second ascent
  - Modify the search to find only patterns that lasted less than a week

```
PATTERN (X+ Y+ W+ Z+)
DEFINE  X AS (price < PREV(price))
        Y AS (price > PREV(price))
```

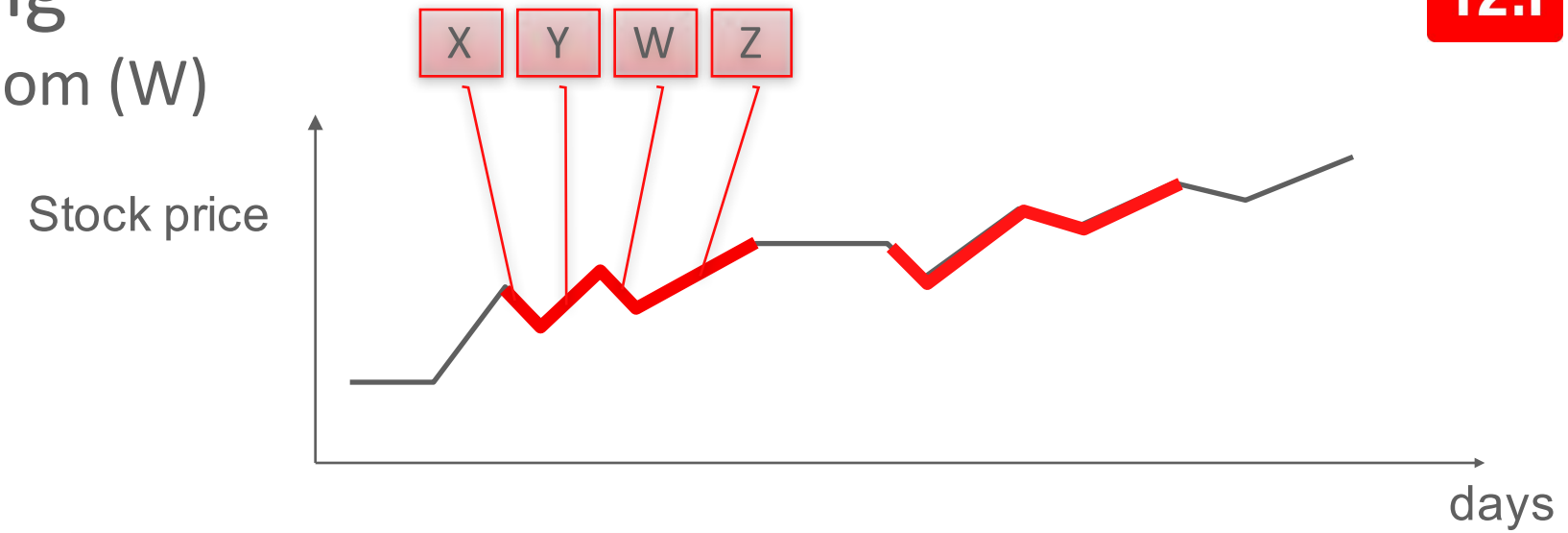
Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

SQL Pattern Matching

Example: Find Double Bottom (W)

NEW IN 12.1

- Find double bottom (W) patterns and report:
  - Beginning and ending date of the pattern
  - Average Price Increase in the second ascent
  - Modify the search to find only patterns that lasted less than a week




```
SELECT first_x, last_z
FROM ticker MATCH_RECOGNIZE (
       PARTITION BY name ORDER BY time
       MEASURES FIRST(x.time) AS first_x
                LAST(z.time)  AS last_z
       ONE ROW PER MATCH
       PATTERN (X+ Y+ W+ Z+)
       DEFINE X AS (price < PREV(price))
              Y AS (price > PREV(price))
              W AS (price < PREV(price))
              Z AS (price > PREV(price))
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

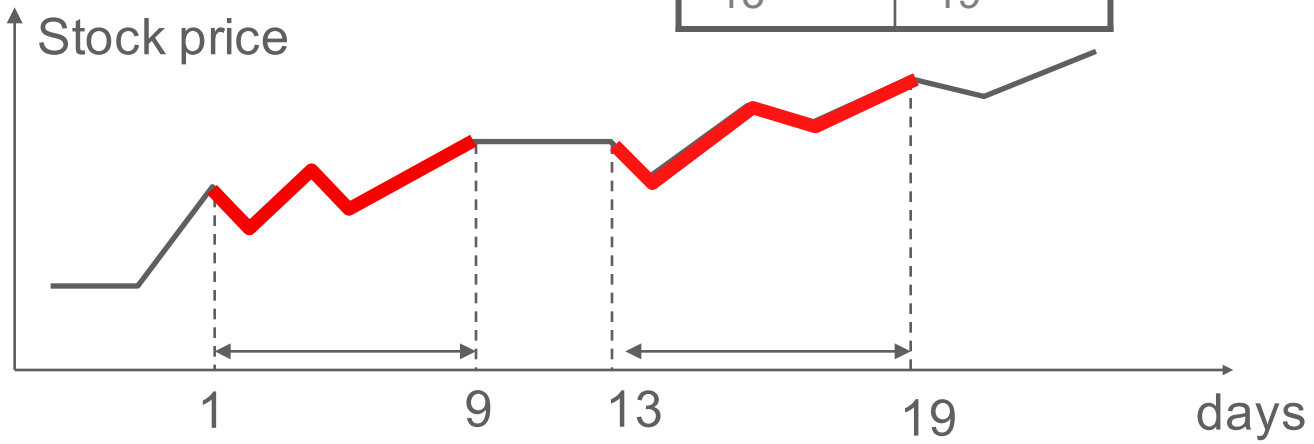
SQL Pattern Matching

Example: Find Double Bottom (W)

NEW IN 12.1

- Find double bottom (W) patterns and report:
  - Beginning and ending date of the pattern
  - Average Price Increase in the second ascent
  - Modify the search to find only patterns that lasted less than a week

First_x	Last_z
1	9
13	19




```
SELECT first_x, last_z
FROM ticker MATCH_RECOGNIZE (
       PARTITION BY name ORDER BY time
       MEASURES FIRST(x.time) AS first_x,
                LAST(z.time)  AS last_z
       ONE ROW PER MATCH
       PATTERN (X+ Y+ W+ Z+)
       DEFINE X AS (price < PREV(price)),
              Y AS (price > PREV(price)),
              W AS (price < PREV(price)),
              Z AS (price > PREV(price)))
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

SQL Pattern Matching

More power to directly applied to your data

NEW IN 12.1

```
            if (q.isEmpty() || eq(q, prev)) {
                state = "F";
                return state;
            }

            return state;
        }

        private boolean eq(String a, String b) {
            if (a.isEmpty() || b.isEmpty()) {
                return false;
            }
            return a.equals(b);
        }

        private boolean gt(String a, String b) {
            if (a.isEmpty() || b.isEmpty()) {
                return false;
            }
            return Double.parseDouble(a) > Double.parseDouble(b);
        }

        private boolean lt(String a, String b) {
            if (a.isEmpty() || b.isEmpty()) {
                return false;
            }
            return Double.parseDouble(a) < Double.parseDouble(b);
        }

        public String getState() {
            return this.state;
        }
    }
    BagFactory bagFactory = BagFactory.getInstance();

    @Override
    public Tuple exec(Tuple input) throws IOException {

        long c = 0;
```

250+ Lines of Java

```
SELECT first_x, last_z
FROM ticker MATCH_RECOGNIZE (
        PARTITION BY name ORDER BY time
        MEASURES FIRST(x.time) AS first_x,
                 LAST(z.time)  AS last_z
        ONE ROW PER MATCH
        PATTERN (X+ Y+ W+ Z+)
        DEFINE X AS (price < PREV(price)),
               Y AS (price > PREV(price)),
               W AS (price < PREV(price)),
               Z AS (price > PREV(price)  AND
                     z.time - FIRST(x.time) <= 7 ))
```

12 Lines of SQL

Less code, easier to maintain, faster to write

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

PL/SQL in SQL

PL/SQL functions embedded in “with” clause

NEW IN 12.1

```
WITH
 FUNCTION get_domain(url VARCHAR2) RETURN VARCHAR2 IS
   pos BINARY_INTEGER;
   len BINARY_INTEGER;
 BEGIN
   pos := INSTR(url, 'www.');
   len := INSTR(SUBSTR(url, pos + 4), '.') - 1;
   RETURN SUBSTR(url, pos + 4, len);
 END;
SELECT
 DISTINCT get_domain(catalog_url)
FROM
 orders;
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

IDENTITY

Auto increment for Oracle

NEW IN 12.1

- Create a table with an id column that is always populated

```
CREATE TABLE t1
(id NUMBER GENERATED AS IDENTITY,
 first_name VARCHAR2(30)
);
```

- Create a table with an id column that is populated if not provided

```
CREATE TABLE t2
(id NUMBER GENERATED BY DEFAULT AS IDENTITY
              (START WITH 100 INCREMENT BY 10),
 first_name varchar2(30)
);
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

32k VARCHAR2/NVARCHAR2

NEW IN 12.1

Longer strings to store

- Enable 32k VARCHAR2 support

```
ALTER SYSTEM set MAX_STRING_SIZE = EXTENDED scope = SPFILE;
```

- Create a table with 32k VARCHAR2

```
CREATE TABLE Applicants
(id           NUMBER GENERATED AS IDENTITY,
 first_name   VARCHAR2(30),
 last_name    VARCHAR2(30),
 application  DATE,
 CV           VARCHAR2(32767)
);
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Row Limit

SQL Standard for row limiting

NEW IN 12.1

- Select only the first 5 rows

```
SELECT employee_id, last_name
  FROM employees
  ORDER BY employee_id
  FETCH FIRST 5 ROWS ONLY;
```

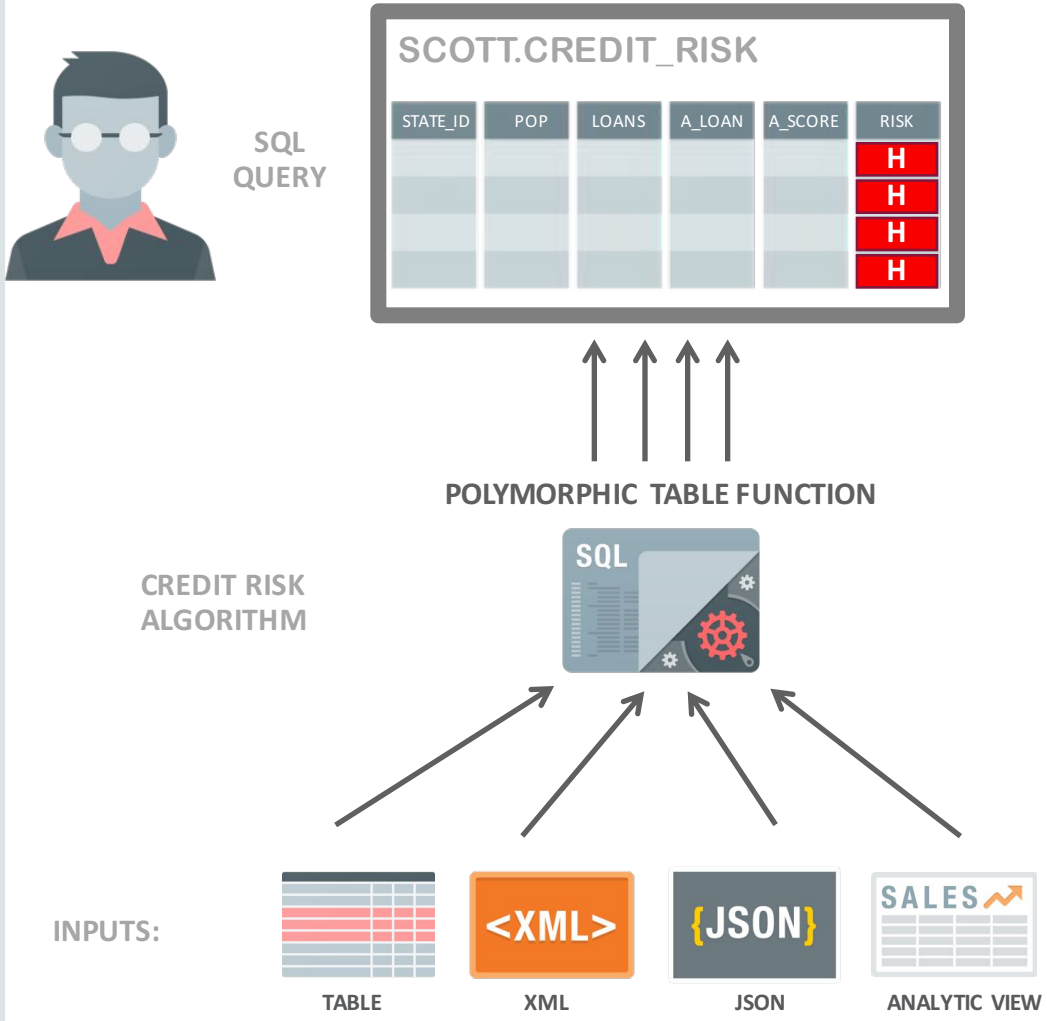
- Select only the first 5% of rows including rows that “tie”

```
SELECT employee_id, last_name, salary
  FROM employees
  ORDER BY salary
  FETCH FIRST 5 PERCENT ROWS WITH TIES;
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Polymorphic Tables: Self-Describing, Fully Dynamic SQL

NEW IN 18c



- Part of ANSI 2016
- Encapsulate **sophisticated algorithms**
  - Hides implementation of algorithms
  - Leverage powerful, dynamic capabilities of SQL
  - Pass in any table-columns for processing
  - Returns SQL rowset (table, JSON, XML doc etc.)
    - E.g. return credit score and associated risk level

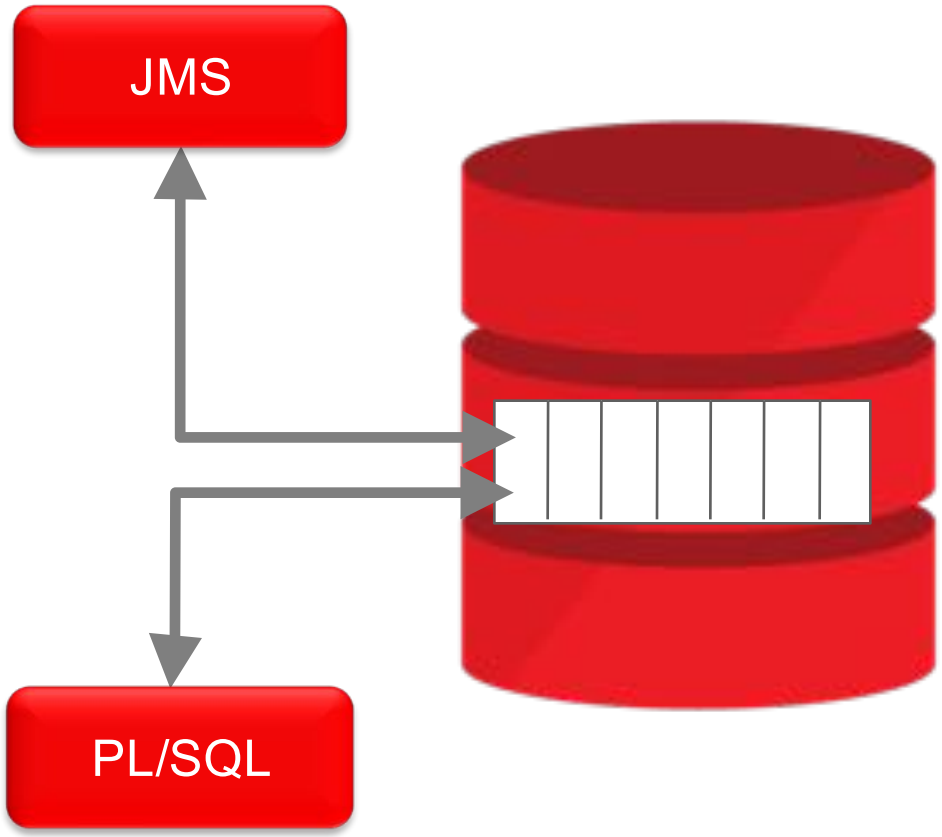
```
SELECT state_id, . . ., AVG(credit_score), risk
FROM   CREDIT_RISK(
           tab  => scott.customers,
           cols => columns(dob, zip,loan_default),
           outs => columns(credit_score, risk_level))
WHERE  risk_level = 'High'
GROUP BY state_id;
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal

Oracle Advanced Queuing (AQ)

Messaging and Notification in the Database

- JMS support
- PL/SQL, OCI, JDBC, .NET support
- Integrated with the Database
- Messaging Gateway



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

AQ-JMS Sharded Queues

NEW IN 12.1

New with 12.1.0.2

- A single logical queue with many “shards”
  - A “shard” is a way of obtaining higher concurrency and throughput via horizontal partitioning.
- Automatic management of session affinity to shards
- Automatic management of table partitions to avoid contention
- Automatic management of partition instance affinity
- Integrated with the database to optimize performance

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

AQ-JMS Sharded Queues

NEW IN 12.1

Key benefits

- Higher throughput
- Less system resource consumption
- Many enqueuers and dequeuers across multiple RAC instances
- Large number of subscribers

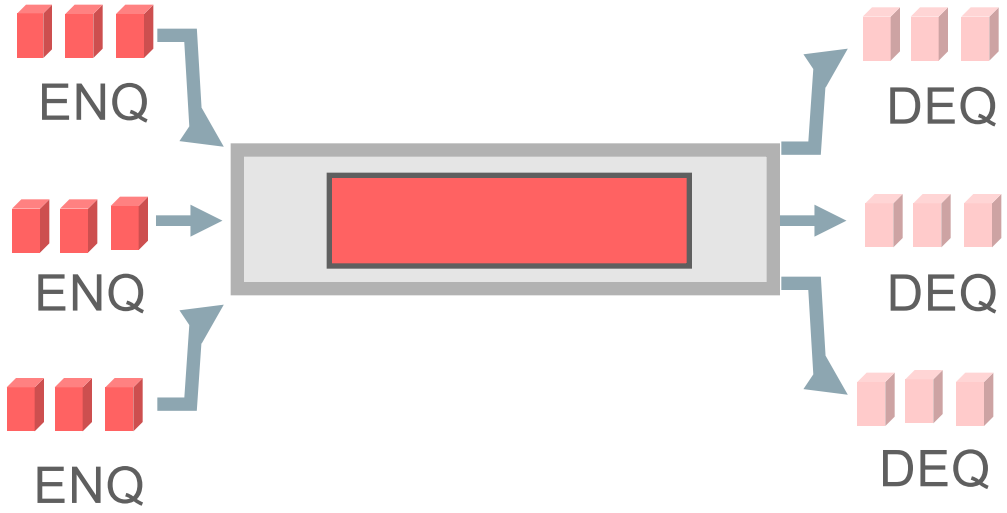
Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

AQ Sharded Queues

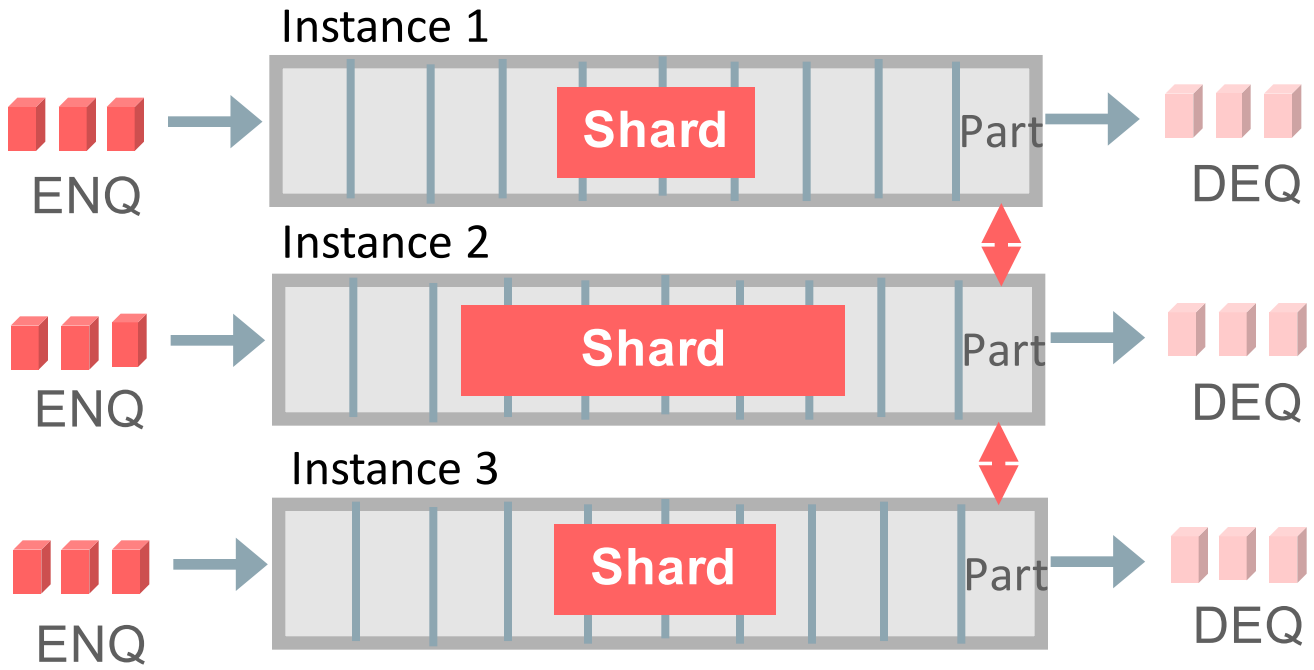
Architecture for Scalability and Performance

NEW IN 12.1

Single logical queue model



Physical Queue: partitions mapped to shards with instance affinity on RAC

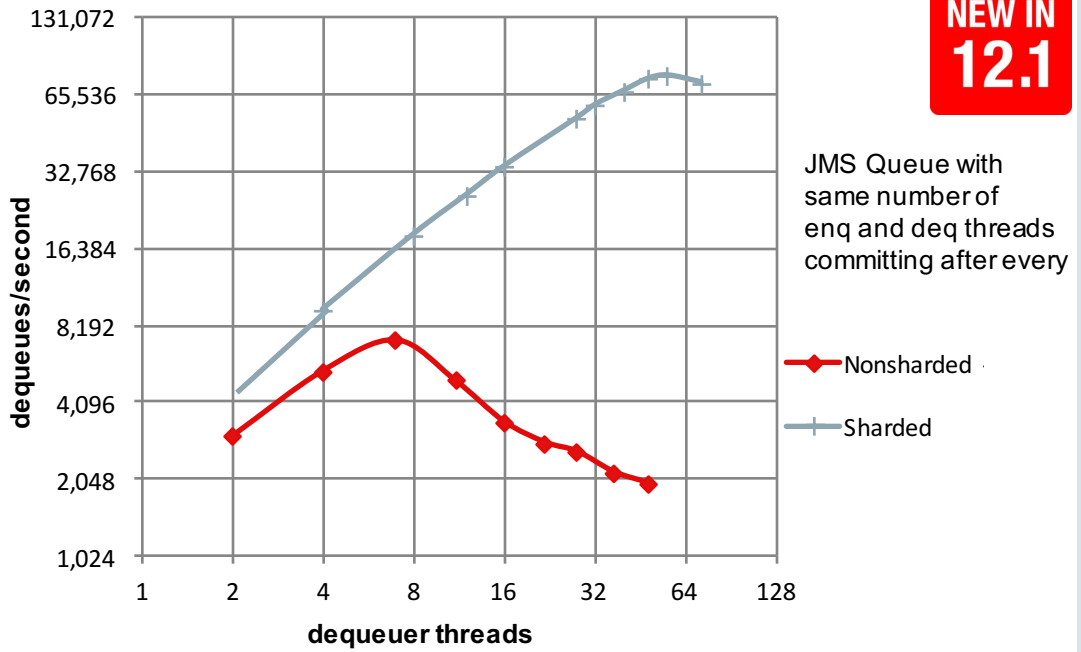


Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

AQ-JMS Sharded Queues

Key benefits

NEW IN 12.1



- Higher throughput
- Less system resource consumption
- Large number of subscribers
- Event-based listener with fewer database connections
- Many concurrent enqueuers and dequeuers across multiple RAC instances
- Backwards Compatible for Standard JMS based applications
  - just recreate the AQ in the database

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

NEW IN 12.2

Analytic Views

- Moves business logic (Aggregations, Hierarchies, Calculations) back into database
- Simple SQL for complex analytic queries
  - no joins or GROUP-BY clauses necessary
- Works on top of pre-existing tables or views
  - no persistent storage
- Built-in data visualization via APEX

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted

NEW IN 12.2

Analytic Views

A new type of view in the Oracle Database

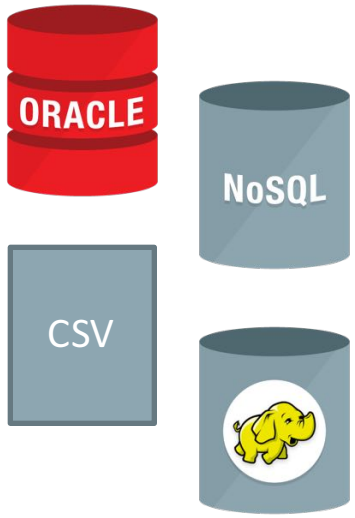
- A new type of view in the Oracle Database
  - Business model and calculation rules are embedded within the Analytic View
- Analytic Views as easily queried with simple SQL and MDX
  - With a smart Analytic View, SQL generation is easy
  - MDX Provider (OLE DB for OLAP) supports Excel PivotTable connections
- Access data from tables, views, external tables and Big Data SQL
  - Use Analytic Views to organize and present a wide variety of data

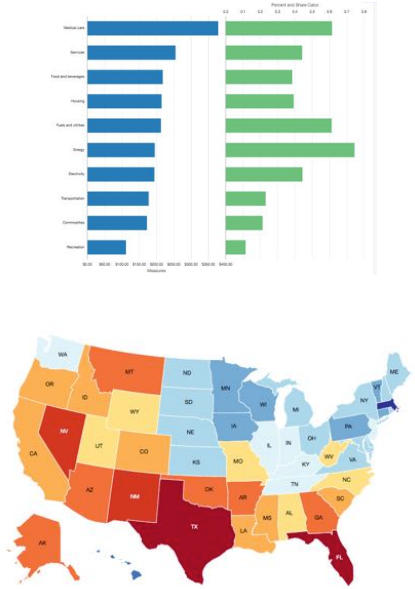
Copyright © 2016 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal and Highly Restricted

NEW IN 12.2

Analytic Views

Easier Access To Your Data





Your Data

Organized & Enhanced

Simple SQL

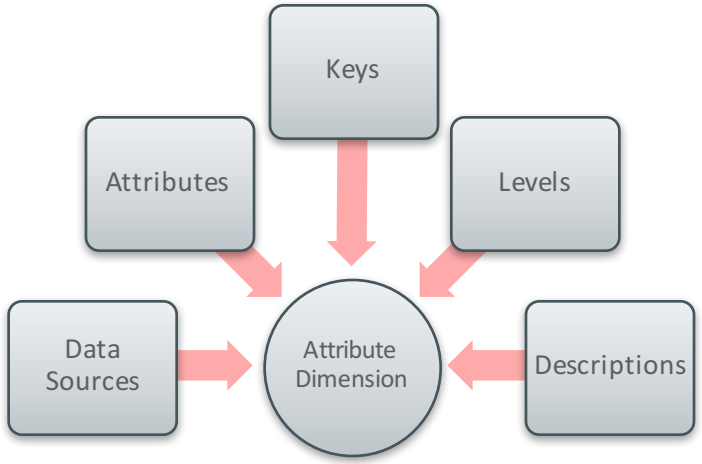
Your Applications

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted

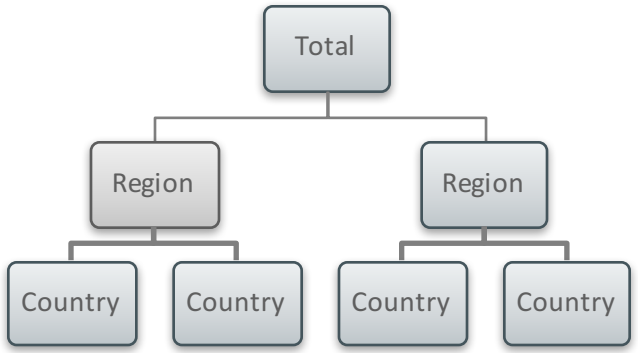
NEW IN 12.2

Three New Database Objects

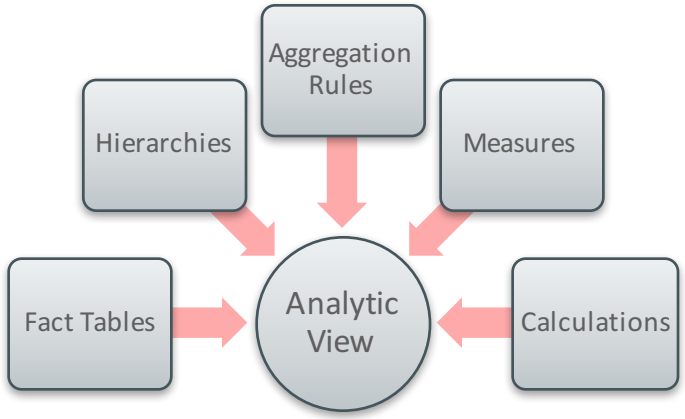
- Attribute Dimensions
  - Map to data objects with dimension / attribute data
  - Identify the roles of columns



- Hierarchies
  - Organizes levels into aggregation and drill paths
  - A new type of view that can be queried with SQL



- Analytic Views
  - Maps to data objects with fact / measure data
  - A new type of view that an can be queries with SQL and MDX



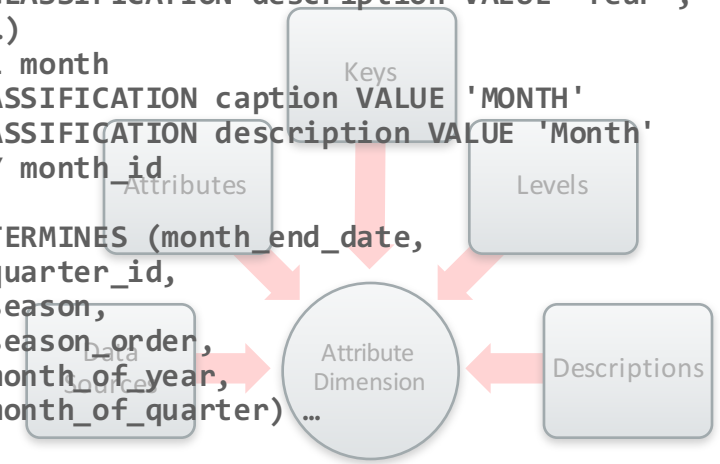
Copyright © 2016 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal and Highly Restricted

NEW IN 12.2

Three New Database Objects

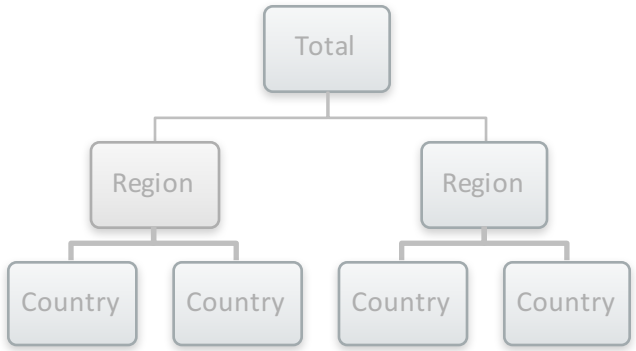
- Attribute Dimensions

```
CREATE OR REPLACE ATTRIBUTE
DIMENSION time_attr_dim
USING time_dim
ATTRIBUTES
 (year_id
   CLASSIFICATION caption VALUE 'YEAR_ID'
   CLASSIFICATION description VALUE 'YEAR ID',
  year_name
    CLASSIFICATION caption VALUE 'YEAR_NAME'
    CLASSIFICATION description VALUE 'Year',
    …)
LEVEL month
  CLASSIFICATION caption VALUE 'MONTH'
  CLASSIFICATION description VALUE 'Month'
  KEY month_id
  …
  DETERMINES (month_end_date,
    quarter_id,
    season,
    season_order,
    month_of_year,
    month_of_quarter) …
```



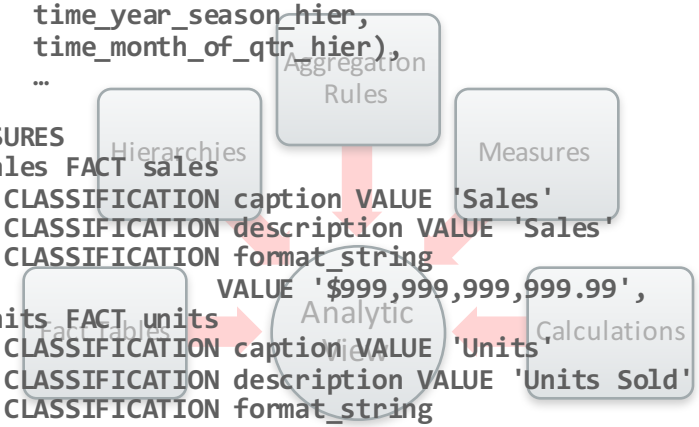
- Hierarchies

```
CREATE OR REPLACE HIERARCHY time_hier
  CLASSIFICATION caption VALUE 'CALENDAR'
  CLASSIFICATION description VALUE'CALENDAR'
USING time_attr_dim
 (month CHILD OF
  quarter CHILD OF
  year)
```



- Analytic Views

```
CREATE OR REPLACE ANALYTIC VIEW sales_av
  CLASSIFICATION caption VALUE 'Sales AV'
  CLASSIFICATION description VALUE
                              'Sales Analytic View'
  CLASSIFICATION created_by VALUE 'George Jones'
USING sales_fact
DIMENSION BY
  (time_attr_dim
    KEY month_id REFERENCES month_id
    HIERARCHIES (
      time_hier DEFAULT,
      time_season_hier,
      time_year_season_hier,
      time_month_of_qtr_hier),
      …
   )
MEASURES
 (sales FACT sales
    CLASSIFICATION caption VALUE 'Sales'
    CLASSIFICATION description VALUE 'Sales'
    CLASSIFICATION format_string
                    VALUE '$999,999,999,999.99',
  units FACT units
    CLASSIFICATION caption VALUE 'Units'
    CLASSIFICATION description VALUE 'Units Sold'
    CLASSIFICATION format_string
                    VALUE '999,999,999,999',
…
```



Copyright © 2016 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal and Highly Restricted

NEW IN 12.2

Analytic Views

Organize and Enhance Data

- Transforms data into a business model and presentation layer in the database
  - Data is organized for easy access and navigation
  - Data is easily extended with interesting calculations and aggregations
  - Data is easily queried with simple SQL
- Easily defined with SQL
  - Complete applications defined with just a few SQL statements
  - Supported by SQL Developer

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted

NEW IN 12.2

Analytic Views

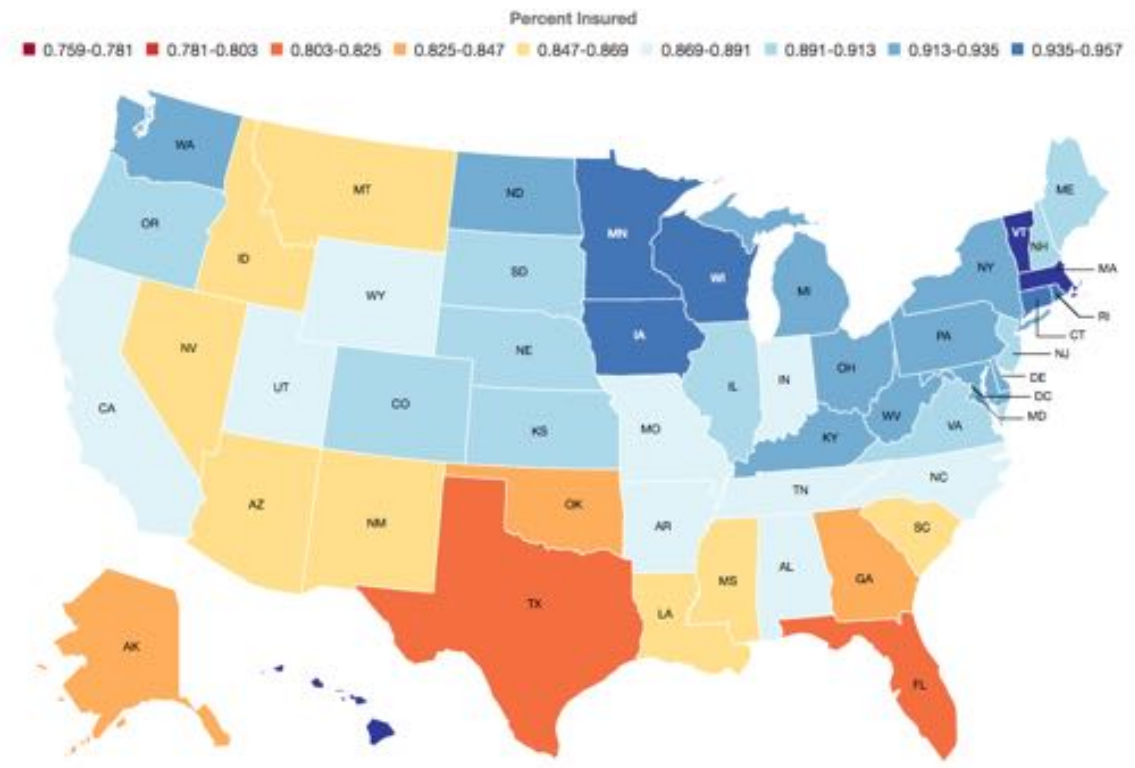
Better for Everyone

- For the data warehouse architect and developer
  - Easily extend star schema with aggregate data and calculations
- For the application developer
  - Simplifies metadata management and SQL generation
- For the business user
  - Built-in, browser-based data visualization via APEX application

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted

NEW IN 12.2

Analytic Views



Health Insurance Coverage Rates by State, 2014

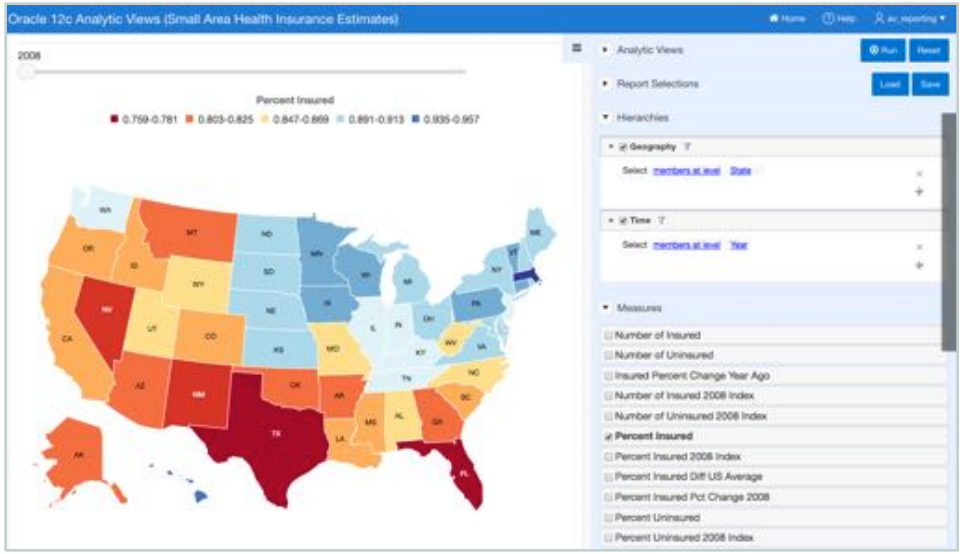
- How would you build this application?
  - Analysis of health insurance coverage rates in the United States
  - Coverage rates by time, counties and states
  - Geographic comparisons
  - Measure improvement over time
  - Interactive data visualization tools for end users

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted

NEW IN 12.2

Analytic Views

- This application can be built with 5 SQL statements
  - Create 2 hierarchies (4 SQL statements)
  - Create 1 analytic view (1 SQL statement)
- Instantly accessible via APEX based data visualizer
- Entirely in the Database



Simple SQL

Analytic View

Data
Tables, Views, etc.

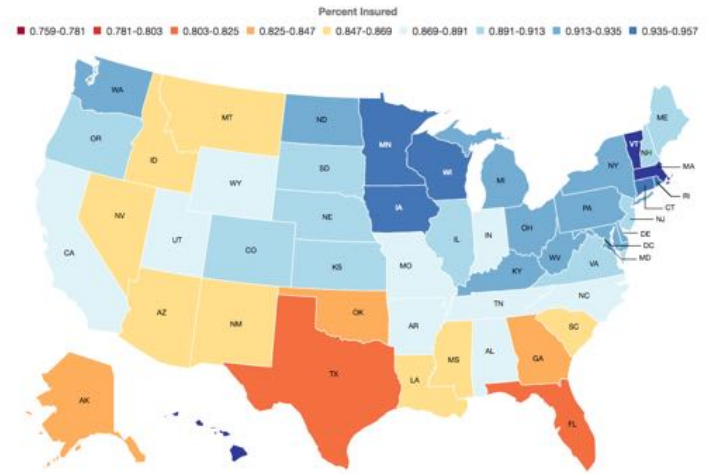
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted

NEW IN 12.2

Analytic Views

Simple SQL

```
SELECT time_hier.member_name AS TIME,
geog_hier.member_name       AS GEOGRAPHY,
  pct_insured
FROM insured_av HIERARCHIES(time_hier,geog_hier)
WHERE time_hier.level_name = 'YEAR'
AND   geog_hier.level_name = 'STATE')
ORDER BY time_hier.hier_order ,
  geog_hier.hier_order;
```



Fact data is selected from analytic views using SQL

Analytic views are views on top of a star schema.
No storage structures

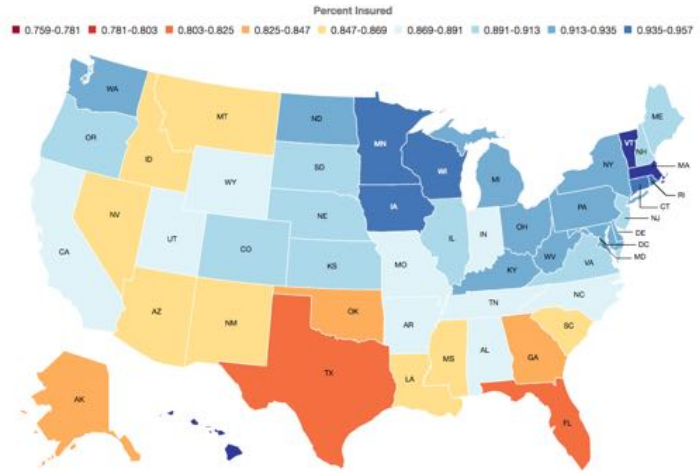
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted

NEW IN 12.2

Analytic Views

Simple SQL

```
SELECT time_hier.member_name AS TIME,
geog_hier.member_name       AS GEOGRAPHY,
  pct_insured
FROM insured_av HIERARCHIES(time_hier,geog_hier)
WHERE time_hier.level_name = 'YEAR'
AND   geog_hier.level_name = 'STATE'
ORDER BY time_hier.hier_order ,
  geog_hier.hier_order;
```



The HIERARCHIES clause specifies the dimensions and hierarchies for this query

No JOIN or GROUP BY clauses in analytic view queries

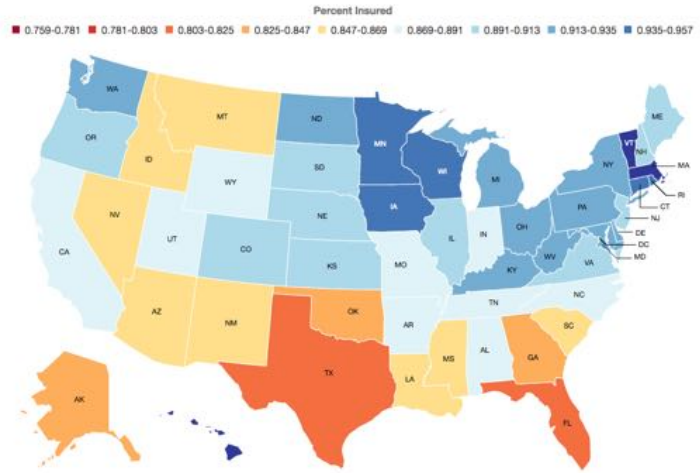
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted

NEW IN 12.2

Analytic Views

Simple SQL

```
SELECT time_hier.member_name AS TIME,
geog_hier.member_name        AS GEOGRAPHY,
  pct_insured
FROM insured_av HIERARCHIES(time_hier,geog_hier)
WHERE time_hier.level_name = 'YEAR'
AND   geog_hier.level_name = 'STATE'
ORDER BY time_hier.hier_order ,
  geog_hier.hier_order;
```



Standardized columns such as ‘member_name’ are selected from the hierarchies

A typical star query would instead select a column such as ‘time.year’

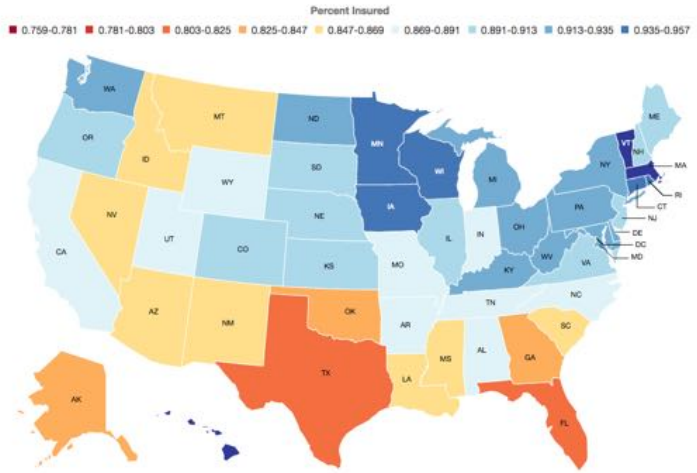
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted

NEW IN 12.2

Analytic Views

Simple SQL

```
SELECT time_hier.member_name AS TIME,
geog_hier.member_name        AS GEOGRAPHY,
  pct_insured
FROM insured_av HIERARCHIES(time_hier,geog_hier)
WHERE time_hier.level_name = 'YEAR'
AND   geog_hier.level_name = 'STATE'
ORDER BY time_hier.hier_order ,
  geog_hier.hier_order;
```



Levels of aggregation are specified in the WHERE clause

When filtering on the level ‘State’ for the time hierarchy, the member named will include California, New York, etc

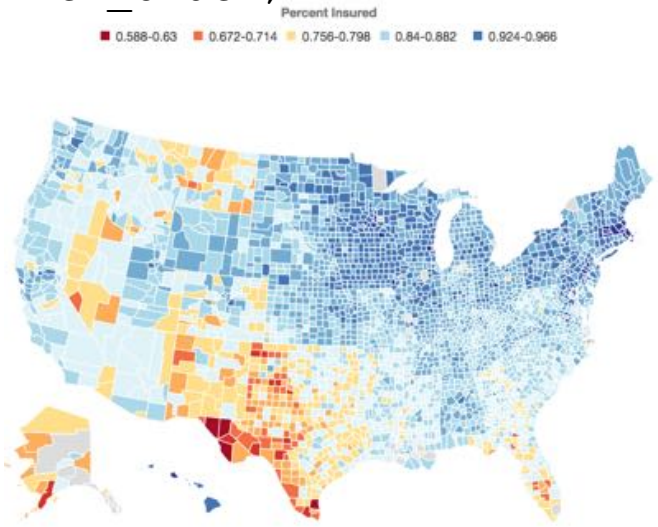
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted

NEW IN 12.2

Analytic Views

Simple SQL

```
SELECT time_hier.member_name AS TIME,
geog_hier.member_name        AS GEOGRAPHY,
  pct_insured
FROM insured_av HIERARCHIES(time_hier,geog_hier)
WHERE time_hier.level_name = 'YEAR'
AND   geog_hier.level_name = 'COUNTY'
ORDER BY time_hier.hier_order ,
  geog_hier.hier_order;
```



To drill, just update the WHERE clause. Everything else remains the same.

The calculations automatically use new hierarchy levels.

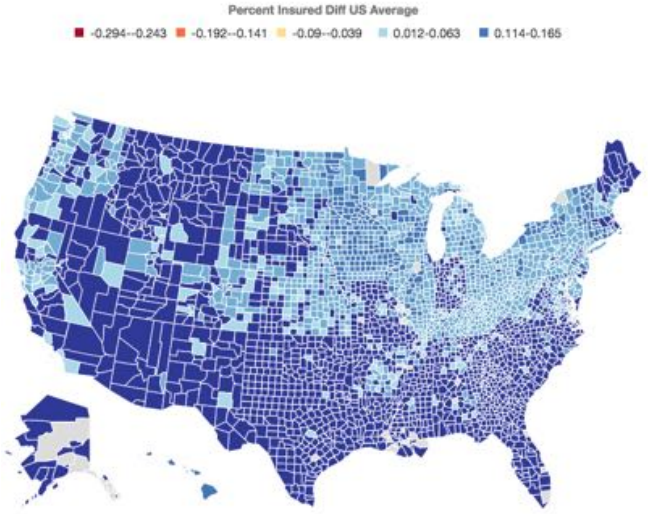
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted

NEW IN 12.2

Analytic Views

Simple SQL

```
SELECT time_hier.member_name AS TIME,
geog_hier.member_name        AS GEOGRAPHY,
  pct_insured_diff_us_avg
FROM insured_av HIERARCHIES(time_hier,geog_hier)
WHERE time_hier.level_name = 'YEAR'
AND   geog_hier.level_name = 'COUNTY'
ORDER BY time_hier.hier_order ,
  geog_hier.hier_order;
```



To select a calculation, just select columns

Calculations are express in the analytic view so they can just be selected in the query

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted

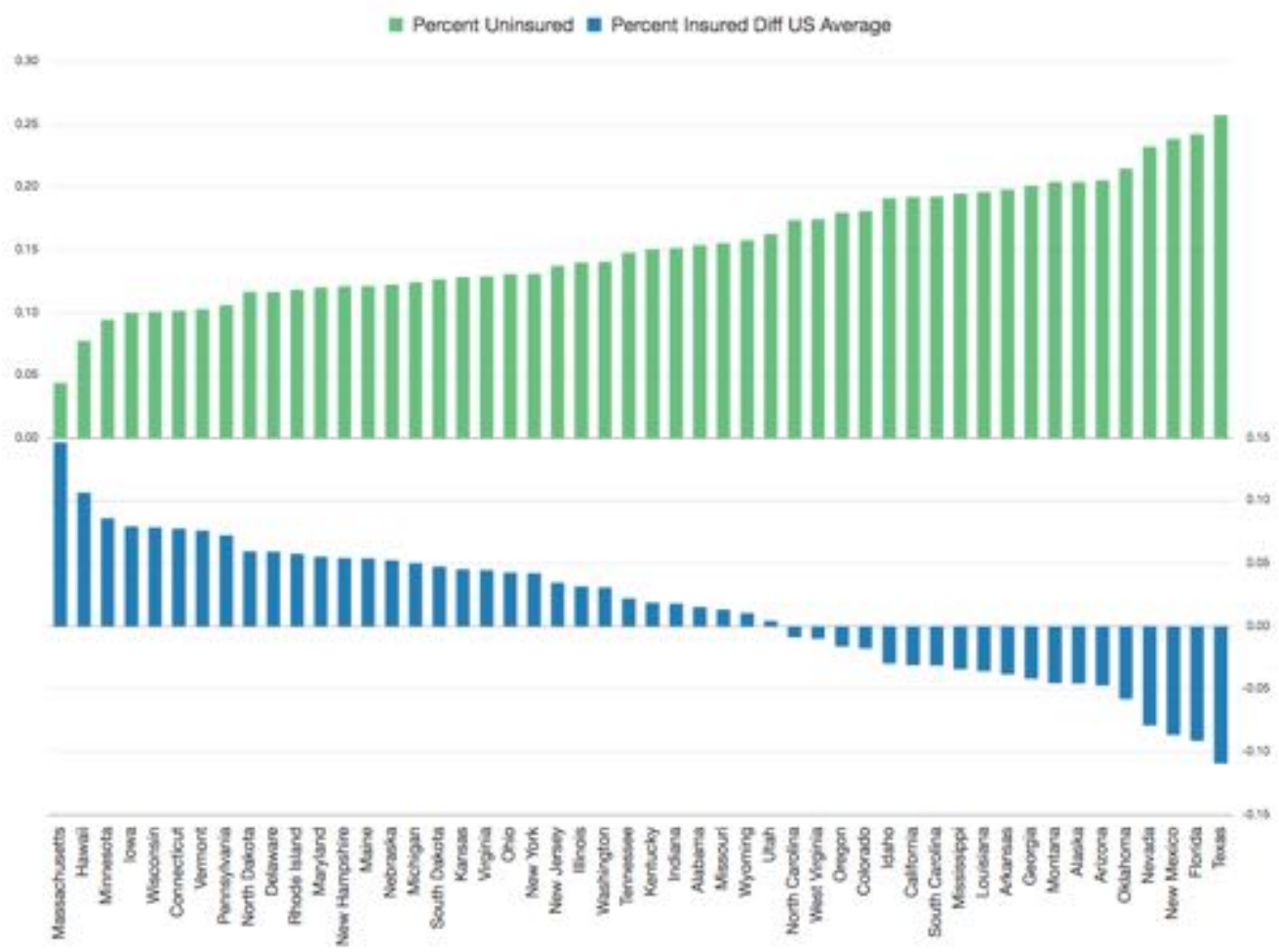
NEW IN 12.2

Analytic Views

Embedded Calculations

- Easily create new measures
  - Simplified syntax based on business model
  - Includes dimensional and hierarchical functions

Add Percent Uninsured Difference from US Average with a single line of code




```
SHARE_OF(pct_uninsured HIERARCHY
  geog_hier
  MEMBER country ['USA']) - 1)
```

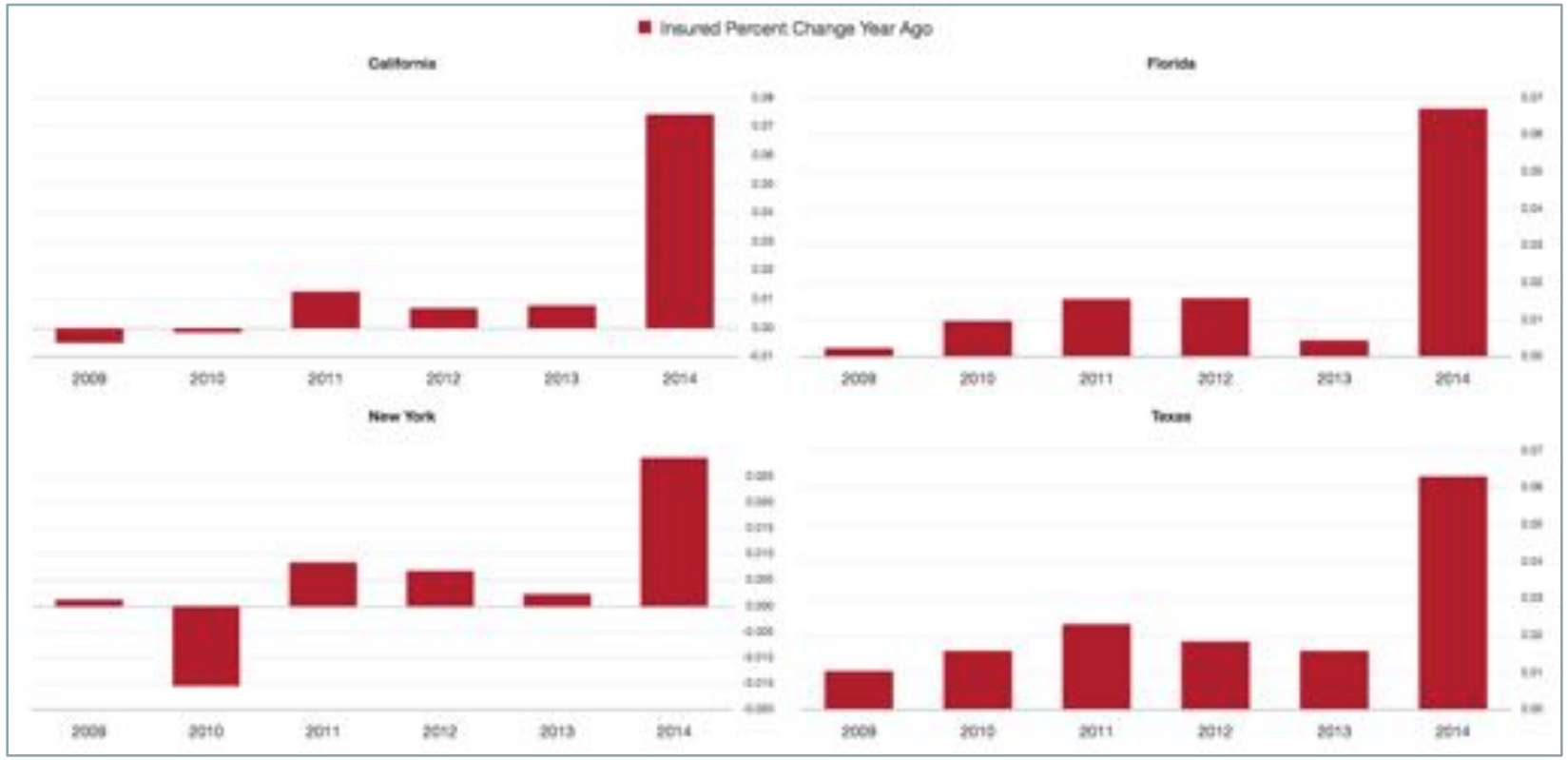
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. Oracle Confidential – Internal

NEW IN 12.2

Analytic Views

Embedded Calculations

Add time series calculations with a single line of code




```
LAG_DIFF_PERCENT(pct_insured)
 OVER (HIERARCHY time_hier
 OFFSET 1 ACROSS ANCESTOR AT LEVEL year)
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. Oracle Confidential – Internal

NEW IN 12.2

“Standard” and Analytic Views

	“Standard” View	Analytic View
Data Sources (FROM)	Yes	Yes
Joins	Yes	Yes
Business Model-Based Calculations	No	Yes
Automatic Hierarchical Columns	No	Yes
Automatic Multi-Level Aggregation	No	Yes
Automatic Filter Expansion	No	Yes
Automatic Outer Join	No	Yes
Automatic Order of Calculation	No	Yes
Presentation Metadata	No	Yes

Copyright © 2016 Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal and Highly Restricted

Machine Learning and Advanced Analytics

NEW IN 18c

- Machine Learning Algorithms available since 9i Release 2 (2002)
- New algorithms in 18
  - Random Forests for Classification
  - Neural Networks for both classification and regression
  - Explicit Semantic Analysis ML algorithm extended to support classification
  - Time Series via Exponential Smoothing
  - CUR decomposition-based algorithm for attribute and row importance
- New ability to export ML models to C and Java for applications deployment

Copyright © 2016 Oracle and/or its affiliates. All rights reserved.

Property Graph Improvements

NEW IN 18c

- PGQL – Property Graph Query Language
  - SQL-like declarative language to query in-memory and in-database Property Graph
  - Supports graph pattern matching and recursive path queries
  - Proposing as ISO standard; Language is also licensed under Apache https://github.com/oracle/pgql-lang
- In-memory virtual columns for RDF Graph
  - Up to 100X faster queries performance
- RDF Graph networks now support list-hash composite partitioning
  - 5 to 10 times query performance improvement

PGQL

```
PATH connects_to := (from) <- (connector) -> (to)
SELECT y.name
WHERE (x:Device) -/:connects_to*/-> (y:Device),
      x.name = 'Regulator, HVMV_Sub_RegB'),
      x != y
```

SQL

```
WITH temp(device_id, device_name) AS (
  -- Anchor member:
  SELECT device_id, name
  FROM   Devices
  WHERE  name = 'Regulator, HVMV_Sub_RegB‘
UNION ALL
  -- Recursive member:
  SELECT Devices.device_id, Devices.name
  FROM   temp, Devices, Connections conn1,
         Connections conn2, Connectors
  WHERE  temp.device_id = conn1.to_device_id
    AND  conn1.from_connector_id =
Connectors.connector_id
    AND  Connectors.connector_id =
conn2.from_connector_id
    AND  conn2.to_device_id = Devices.device_id
    AND  temp.device_id != Devices.device_id)
CYCLE device_id SET cycle TO 1 DEFAULT 0
SELECT DISTINCT device_name
FROM temp
WHERE cycle = 0
  AND device_name = 'Regulator, HVMV_Sub_RegB'
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved. |

JSON Support

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.1

JSON Queries using SQL

- Simple Queries

```
SELECT j.PO_DOCUMENT
  FROM J_PURCHASEORDER j
    WHERE j.PO_DOCUMENT.PONumber = 1600;
```

- Advanced queries using JSON path expressions

```
SELECT JSON_VALUE(PO_DOCUMENT,
'$.LineItems[0].Part.UnitPrice' returning NUMBER(5,3))
  FROM J_PURCHASEORDER p
    WHERE JSON_VALUE(PO_DOCUMENT, '$.PONumber' RETURNING
      NUMBER(10)) = 1600;
```

  - Complies with SQL:2016 syntax

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

NEW IN 12.2

JSON integration with PL/SQL

- New PL/SQL objects enable fine grained manipulation of JSON content
  - **JSON_OBJECT_T** : for working with JSON objects
  - **JSON_ARRAY_T** : for working with JSON Arrays
  - JSON_OBJECT_T and JSON_ARRAY_T are subtypes of JSON_ELEMENT_T
- These objects provide a set of methods for manipulating JSON
- Piecewise updates of JSON documents now supported in PL/SQL

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

JSON integration with PL/SQL

```
WITH FUNCTION updateTax(JSON_DOC in VARCHAR2 ) RETURN VARCHAR2 IS
     jo JSON_OBJECT_T;
     price NUMBER;
     taxRate NUMBER;
BEGIN
     jo := JSON_OBJECT_T(JSON_DOC);
     taxRate := jo.get_Number('taxRate');
     price := jo.get_Number('total');
     jo.put('totalIncludingTax', price * (1+taxRate));
     RETURN jo.to_string();
END;
ORDERS AS (
     SELECT '{"taxRate":0.175,"total":10.00}' JSON_DOCUMENT
       FROM dual
)
SELECT JSON_DOCUMENT, updateTax(JSON_DOCUMENT)
  FROM ORDERS;

JSON_DOCUMENT                    UPDATETAX(JSON_DOCUMENT)
-------------------------------- ----------------------------------------------------
{"taxRate":0.175,"total":10.00} {"taxRate":0.175,"total":10.00,"totalIncludingTax":11.75}
```

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

Data Guide: Understanding your JSON documents

- Metadata discovery: discovers the structure of collection of JSON documents
  - Optional: deep analysis of JSON for List of Values, ranges, sizing etc.
- Automatically Generates
  - Virtual columns
  - Relational views
    - De-normalized relational views for arrays
  - Reports/Synopsis of JSON structure

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

Data Guide: Automatic Schema Inference

Table containing JSON documents

```
SQL> desc MOVIE_TICKETS
NAME                          TYPE
----------------------------  -----------
BOOKING_ID                    RAW(16)
BOOKING_TIME                  TIMESTAMP(6)
BOOKING_DETAILS               VARCHAR2(4000)
```

```
{
   "Theater":"AMC  15",
   "Movie":"Jurrasic  World  3D",
   "Time“:2015-11-26T18:45:00",
   "Tickets":{
      "Adults":2
    }
}
```

JSON DataGuide

```
DBMS_JSON.AddVC(
   'MOVIE_TICKETS',
   'BOOKING_DETAILS');
```

Table enhanced with virtual columns

```
SQL> desc MOVIE_TICKETS
NAME                          TYPE
----------------------------  -----------
BOOKING_ID                    RAW(16)
BOOKING_TIME                  TIMESTAMP(6)
BOOKING_DETAILS               VARCHAR2(4000)
BOOKING_DETAILS$Movie         VARCHAR2(16)
BOOKING_DETAILS$Theater       VARCHAR2(16)
BOOKING_DETAILS$Adults        NUMBER
BOOKING_DETAILS$Time          VARCHAR2(32)
```

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

JSON Search Index : A universal index for JSON content

```
CREATE SEARCH INDEX JSON_SEARCH_INDEX
  ON J_PURCHASEORDER (PO_DOCUMENT) FOR JSON;
```

- Supports searching on JSON using key, path and value
- Supports range searches on numeric values
- Supports full text searches:
  - Full boolean search capabilities (and, or, and not)
  - Phrase search, proximity search and "within field" searches.
  - Inexact queries: fuzzy match, soundex and name search.
  - Automatic linguistic stemming for 32 languages
  - A full, integrated ISO thesaurus framework

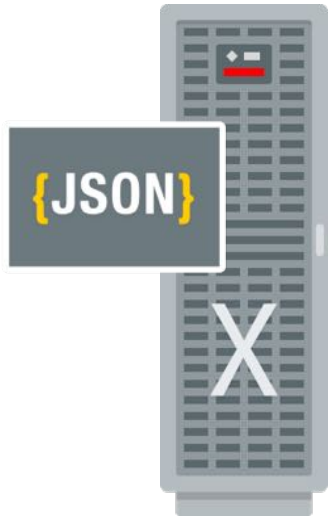
Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

Query Optimizations for JSON

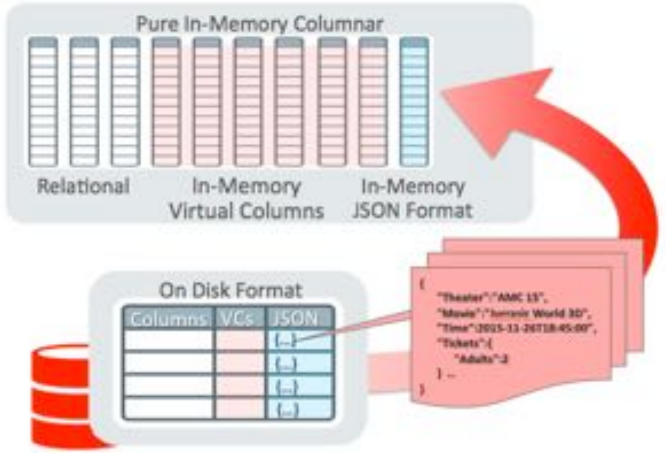
Exadata Smart Scans

- Exadata Smart Scans execute portions of SQL queries on Exadata storage cells
- JSON query operations 'pushed down' to Exadata storage cells
  - Massively parallel processing of JSON documents



In-Memory Columnar Store

- Virtual columns, included those generated using JSON Data Guide loaded into In-Memory Virtual Columns
- JSON documents loaded using a highly optimized In-Memory binary format
- Query operations on JSON content automatically directed to In-Memory



Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

Native JSON Generation

```
SQL> SELECT JSON_OBJECT('Id' is EMPLOYEE_ID, 'FirstName' is FIRST_NAME,
 2                      'LastName' is LAST_NAME) JSON
 3      FROM HR.EMPLOYEES
 4        WHERE EMPLOYEE_ID = 100;

JSON
--------------------------------------------------------------------------------
{ "Id" : 100 , "FirstName" : "Steven" , "LastName" : "King" }

SQL>
```

- JSON generation functions available:
  - JSON_OBJECT / JSON_OBJECTAGG
  - JSON_ARRAY / JSON _ARRAYAGG

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | Public

JSON Enhancements

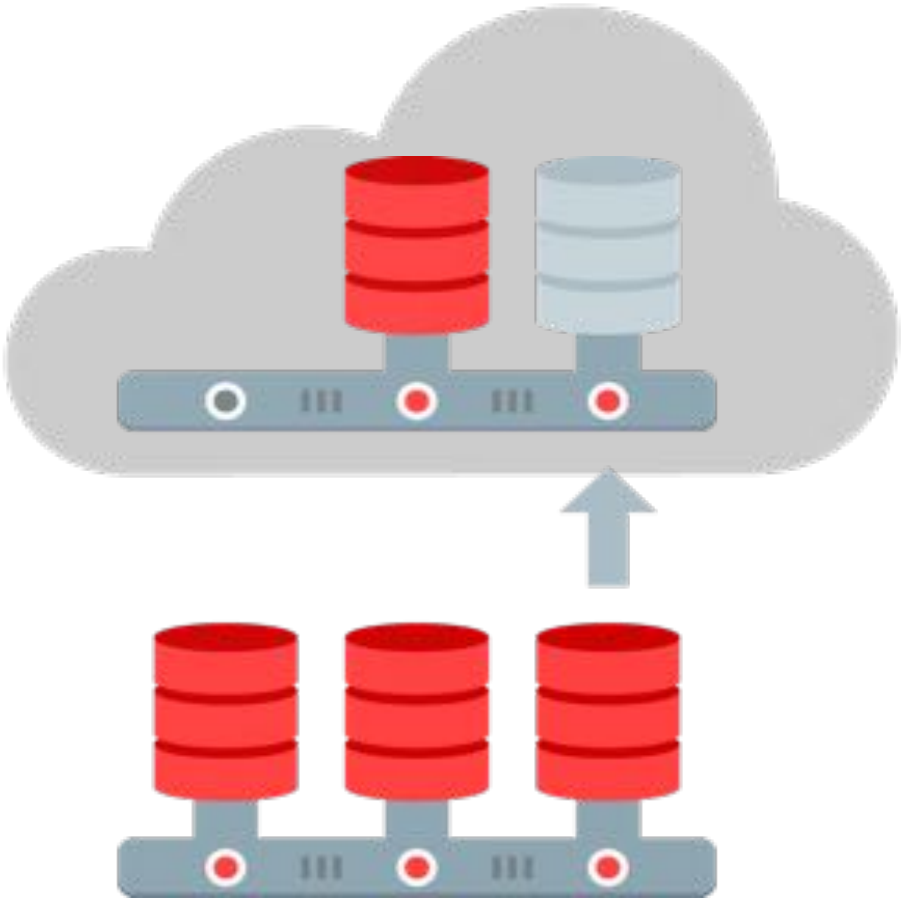
NEW IN 18c

Simpler development of JSON-centric applications using Oracle Database

- Generate large JSON documents from relational data
  - JSON generation extended to supports LOB's
- New SODA (Simple Oracle Document Access) drivers
  - OCI and PL/SQL now added, in additional to JSON and REST
  - Simple, non-relational ('nosql-like') API for accessing JSON data
- New TREAT (<expression> AS JSON) operator
  - Dynamically declare operands to be handled as JSON data, enabling more seamless JSON optimizations
- Extended key length for JSON search indexes
  - Raise the key length from 64 bytes to 255 bytes; enables faster search queries for JSON objects containing long key names.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal

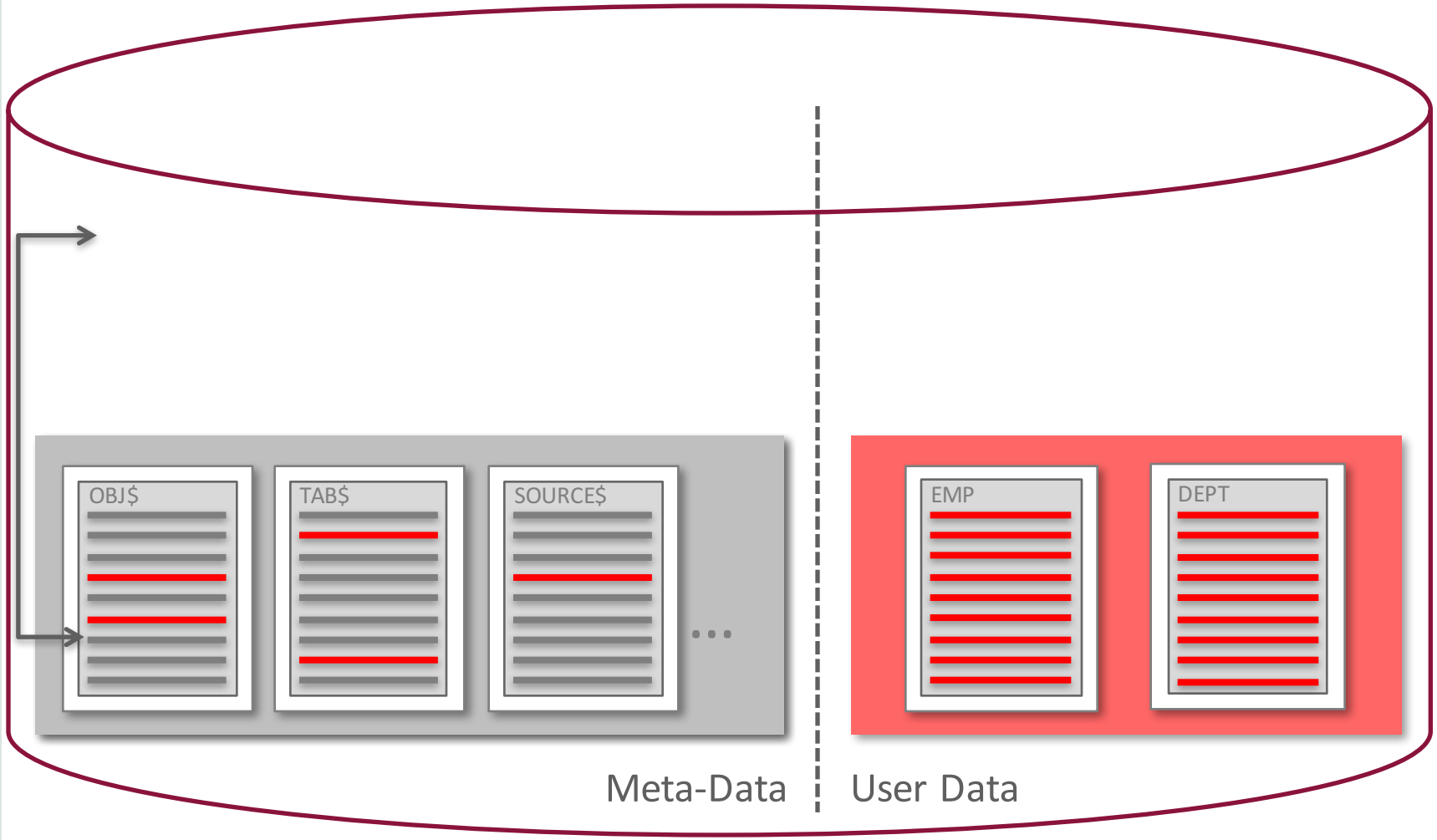
Multitenant



Copyright © 2014 Oracle and/or its affiliates. All rights reserved. Public

Oracle Data and User Data

Before 12.1: Oracle and user data intermingle over time



- New database contains Oracle meta-data only
- Populate database with user data
  - Oracle and customer meta-data intermingled
  - Portability challenge!
- Multitenant fix: *Horizontally-partitioned data dictionary*
  - Only Oracle-supplied meta-data remains in root

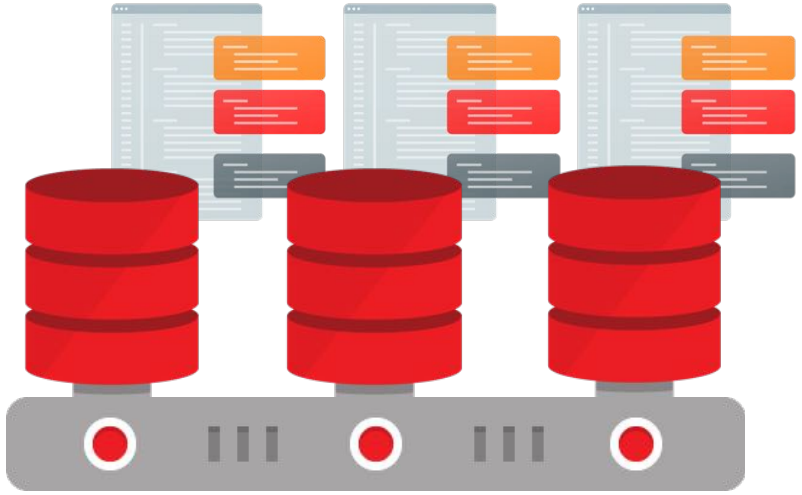
Copyright © 2014 Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

Application Containers

Programs replicated across PDBs

the brooklyn bean V2



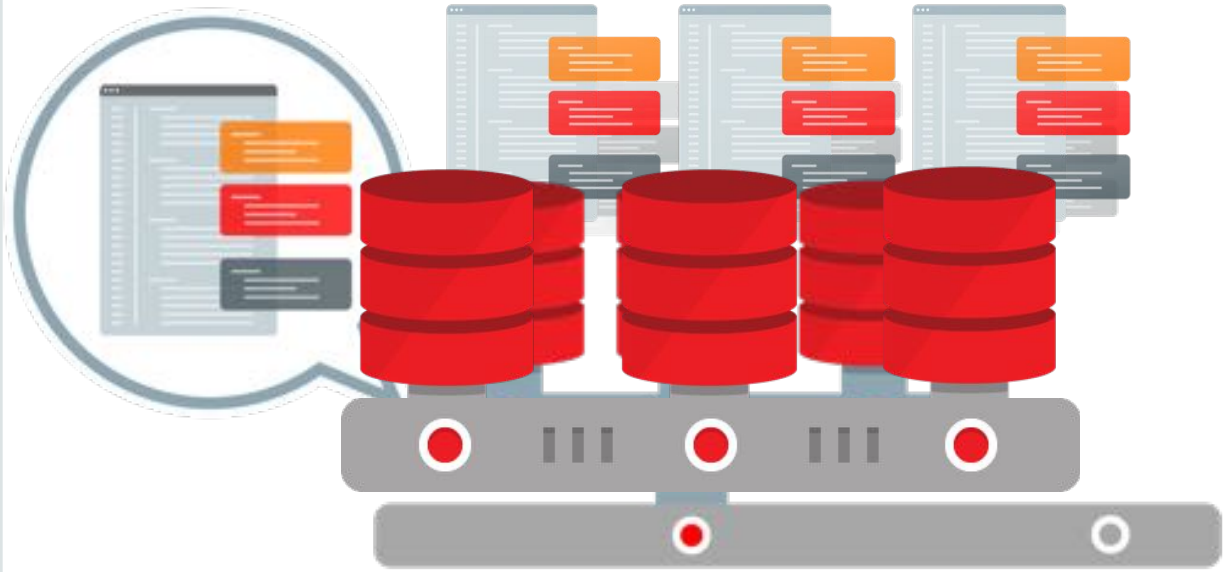
Copyright © 2014 Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

Application Containers

Root container for your applications

- Application Container comprises
  - Application Root (Master)
  - Application PDBs (for each Tenant)
  - Application Seed (for provisioning)
- PDBs share application objects
  - Code, metadata and data
- Further simplifies management
  - Apply updates to application container
  - Sync tenant PDBs from central master
- Suitable for all applications
  - SaaS, franchise, divisional, etc.

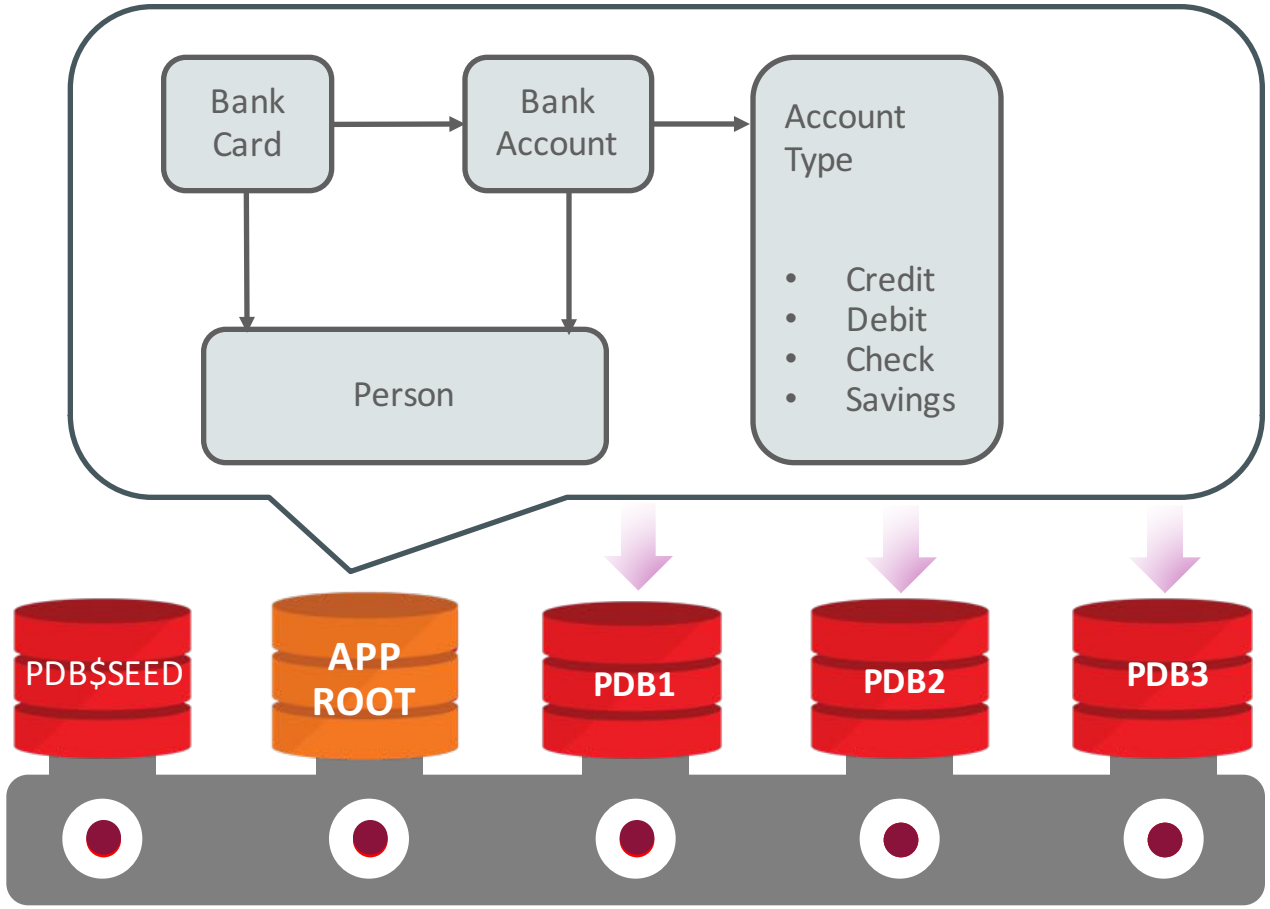


Copyright © 2014 Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

Application Containers

Share & propagate across multiple PDBs

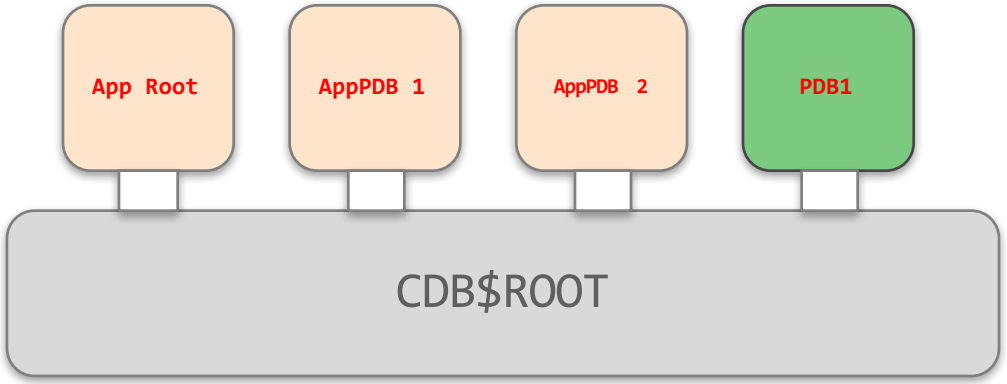


Copyright © 2014 Oracle and/or its affiliates. All rights reserved. | Public

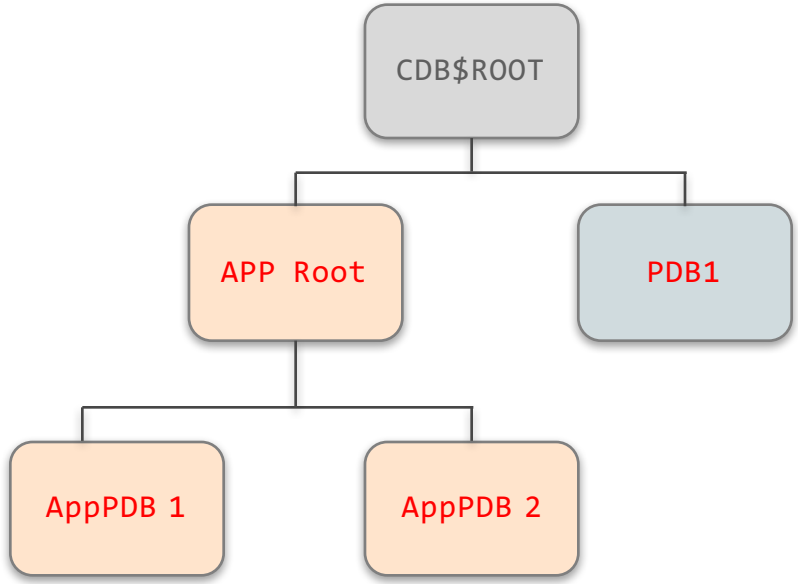
NEW IN 12.2

What is an Application Container ?

- An Application container is a collection of PDBs consisting of Application Root and all Application PDBs associated with it



Physical Representation



Logical Representation

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

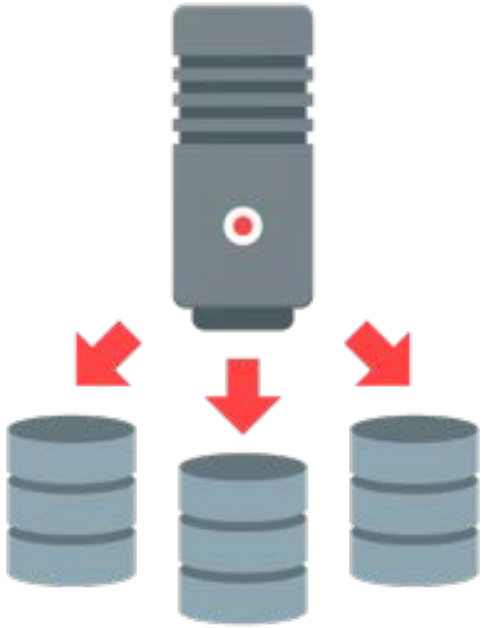
Application Containers
The future of Database Application Development

- Application Root PDB for defining application master
  - Metadata and common data shared across tenant PDBs
- Install one copy of your application
- Instant provisioning of an Application PDB/Tenant (with a seed PDB)
- Container Data views for reporting across PDBs (CONTAINERS clause based)
- Supports in-place simple patching
- Supports Unplug/Plug upgrade across Application Root

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. | Public

NEW IN
12.2

Sharding

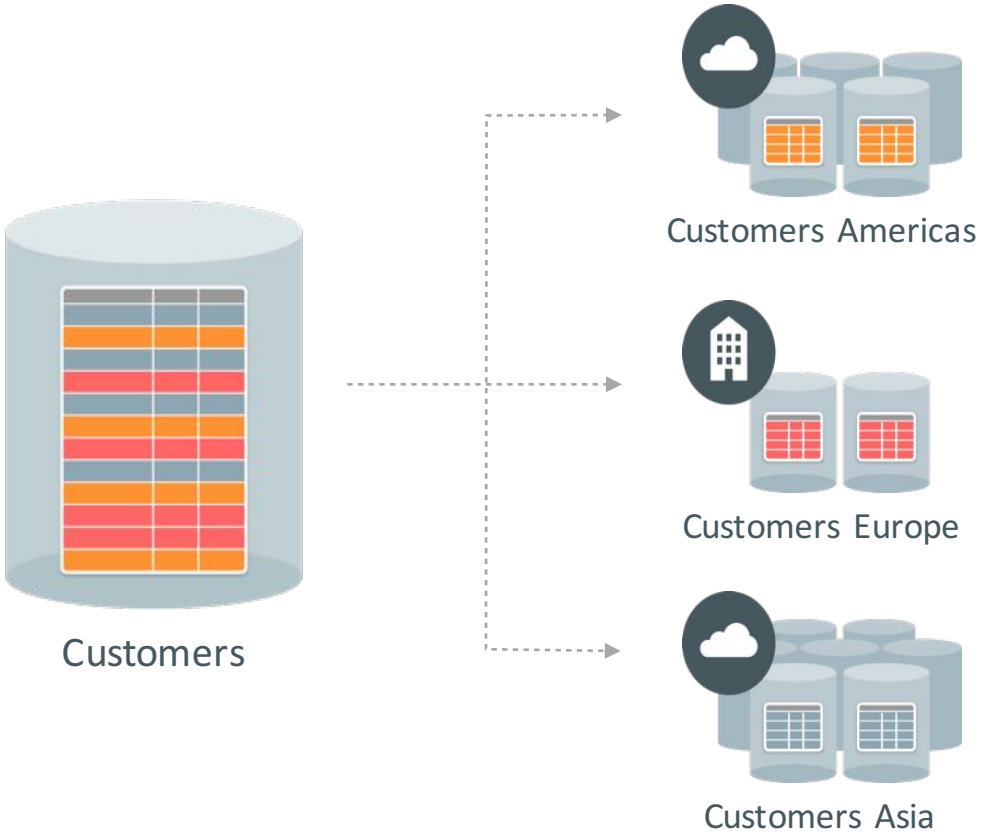


Copyright © 2015, Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

Oracle Database Sharding

Oracle Database for web-scale applications



One giant database partitioned into many small databases (shards)

- RAC and Data Guard meet needs of over 99% of applications while preserving application transparency
- Some Global-Scale OLTP applications prefer to shard massive databases into a farm of smaller databases
  - Avoid scalability or availability edge cases of a single large database
  - Willing to customize data model and applications to enable transactions to be automatically routed to the right shard
- Native SQL for sharding tables across up to 1000 Shards
  - Routing of SQL based on shard key, and cross shard queries
  - Online addition and reorganization of shards
  - Linear scalability of data, workload, users with isolation

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

Application Suitability for Sharding

OLTP Applications with the Following Characteristics

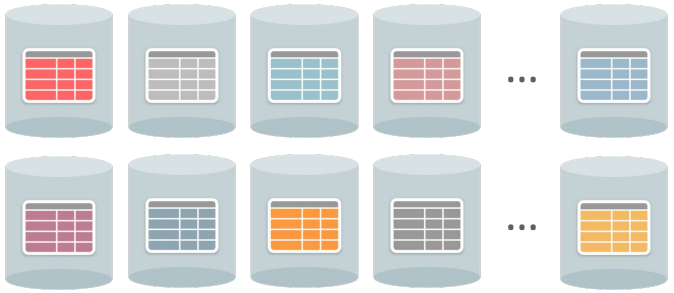
- Applications for massive scale
  - E.g. e-commerce, mobile, social etc.
- Applications must be shard-aware
- Primary usage pattern
  - Single-shard operations based on shard key , e.g. customer_id, account_id etc.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. | Public

Oracle Sharding Automated Distribution

Enhanced SQL syntax for Sharding

```
…

CREATE SHARDED TABLE Customers
( CustId    VARCHAR2(60) NOT NULL,
  FirstName VARCHAR2(60),
  LastName  VARCHAR2(60),
  …
    PRIMARY KEY(CustId),
)
PARTITION BY CONSISTENT HASH (CustId)
 …
```



- SQL syntax for creating sharded tables
  - Not proprietary APIs as with NoSQL
- Creation of a sharded table automatically partitions data across shards
  - Transparent resharding as data grows
- Choice of sharding methods:
  - System managed - consistent hash
  - User defined - range, list
  - Composite - range-hash, list-hash
- Common reference data (e.g. Price List) is automatically duplicated on all shards
- Supports shard placement in specific geographies to satisfy government data privacy

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

Sharded Schema

Customers

Customer	Name
123	Mary
456	John
999	Peter

Orders

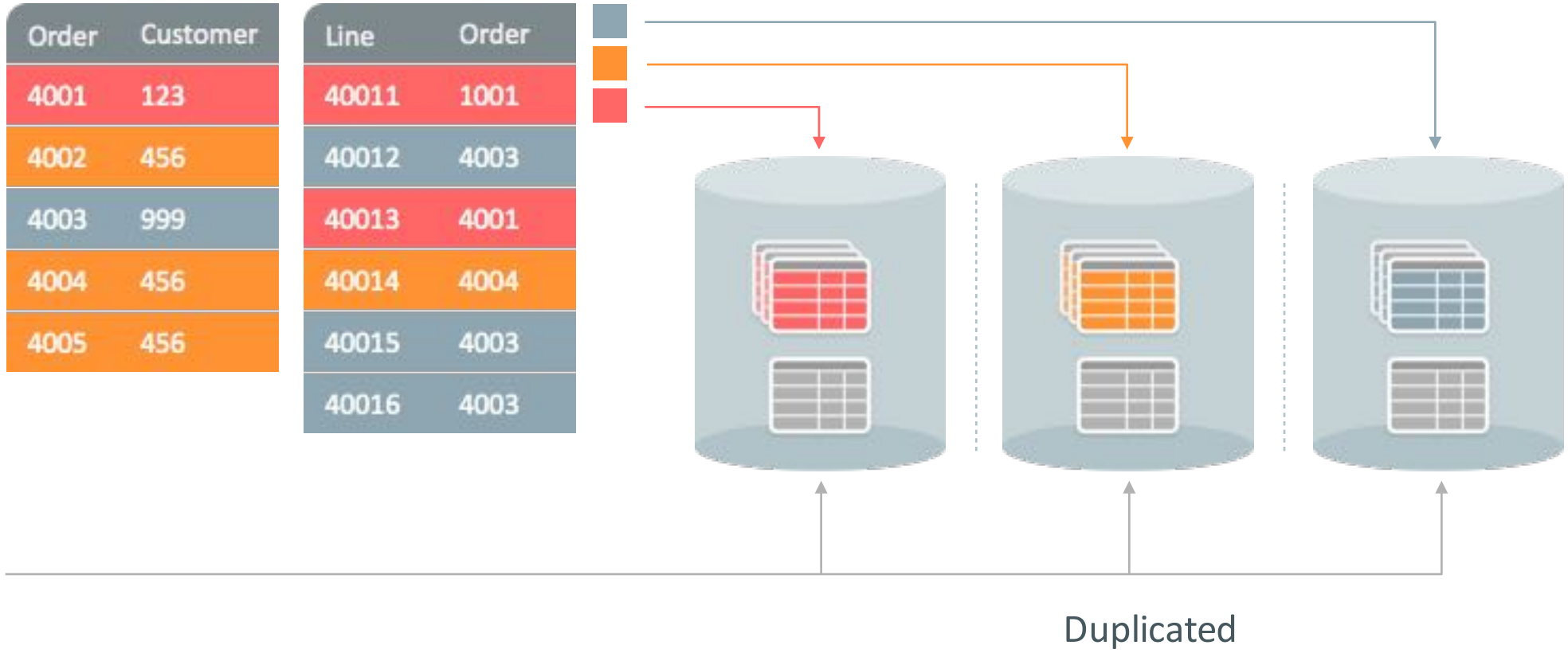
Order	Customer
4001	123
4002	456
4003	999
4004	456
4005	456

Line Items

Line	Order
40011	1001
40012	4003
40013	4001
40014	4004
40015	4003
40016	4003

Products

SKU	Product
100	Coil
101	Piston
102	Belt



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. | Public

Sharded Table Family – Enhanced SQL DDL Syntax

NEW IN 12.2

```
CREATE SHARDED TABLE Customers
( CustNo NUMBER NOT NULL,
  Name VARCHAR2(50),
 ….
  Class VARCHAR2(3),
  CONSTRAINT RootPK PRIMARY
KEY(CustNo)
)
PARTITION BY CONSISTENT HASH (CustNo)
PARTITIONS AUTO
TABLESPACE SET ts1 ;
```

```
CREATE LOOKUP TABLE Products(
SKU NUMBER(4) PRIMARY KEY,
Product VARCHAR2(20),
Price NUMBER(6,2))
)
TABLESPACE dupl ;
```

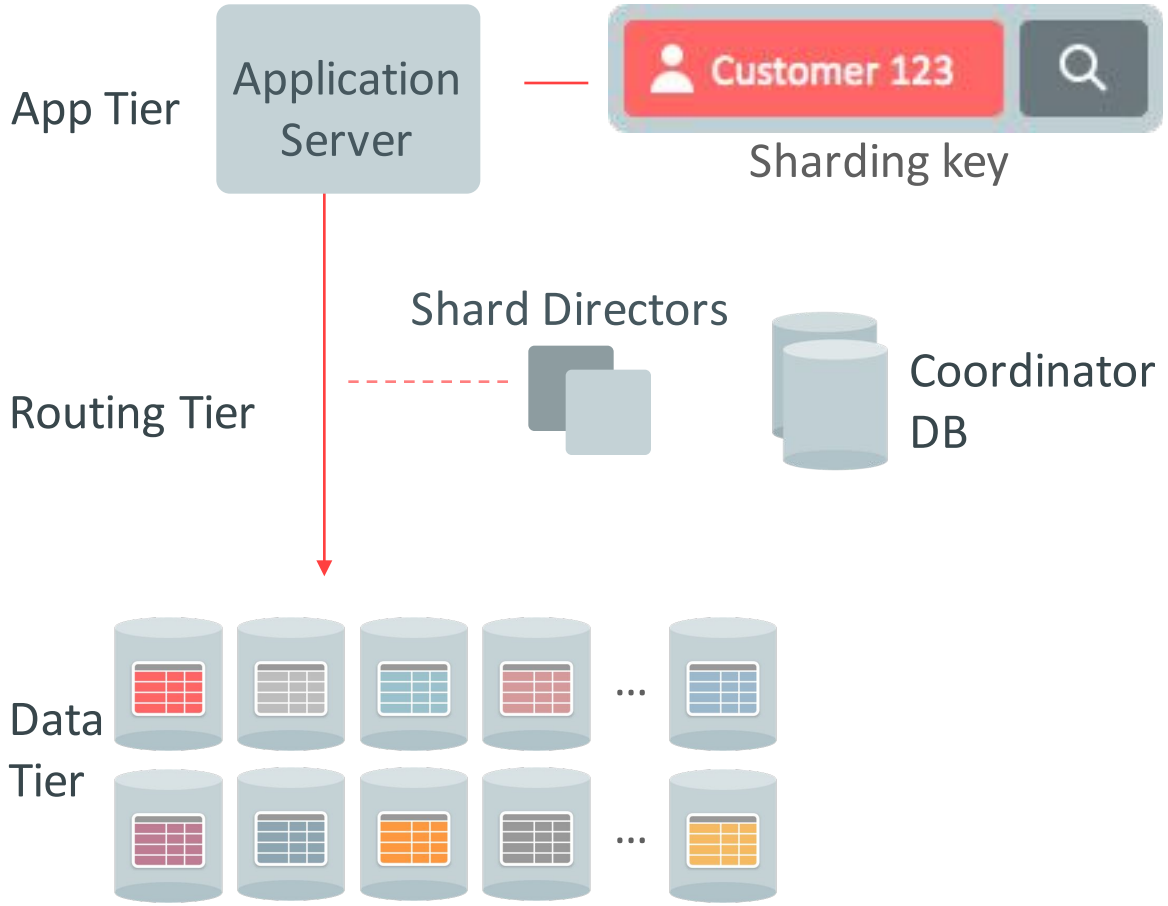
```
CREATE SHARDED TABLE Orders
( OrderNo NUMBER(5),
  CustNo NUMBER(3),
  OrderDate DATE ,
  …
  CONSTRAINT CustFK FOREIGN KEY
(CustNo)
    REFERENCES Customers(CustNo)
)
PARTITION BY REFERENCE (CustFK) ;
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

Routing Support on Client for Highest Speed

- Clients pass sharding key (e.g. Customer ID) to Connection pool, connection is routed to the right shard
- Fast: caching key ranges on client ensures that most accesses go directly to the shard
- Scalable: easily scales with more clients and shards
- Supports UCP, OCI, ODP.NET, and JDBC

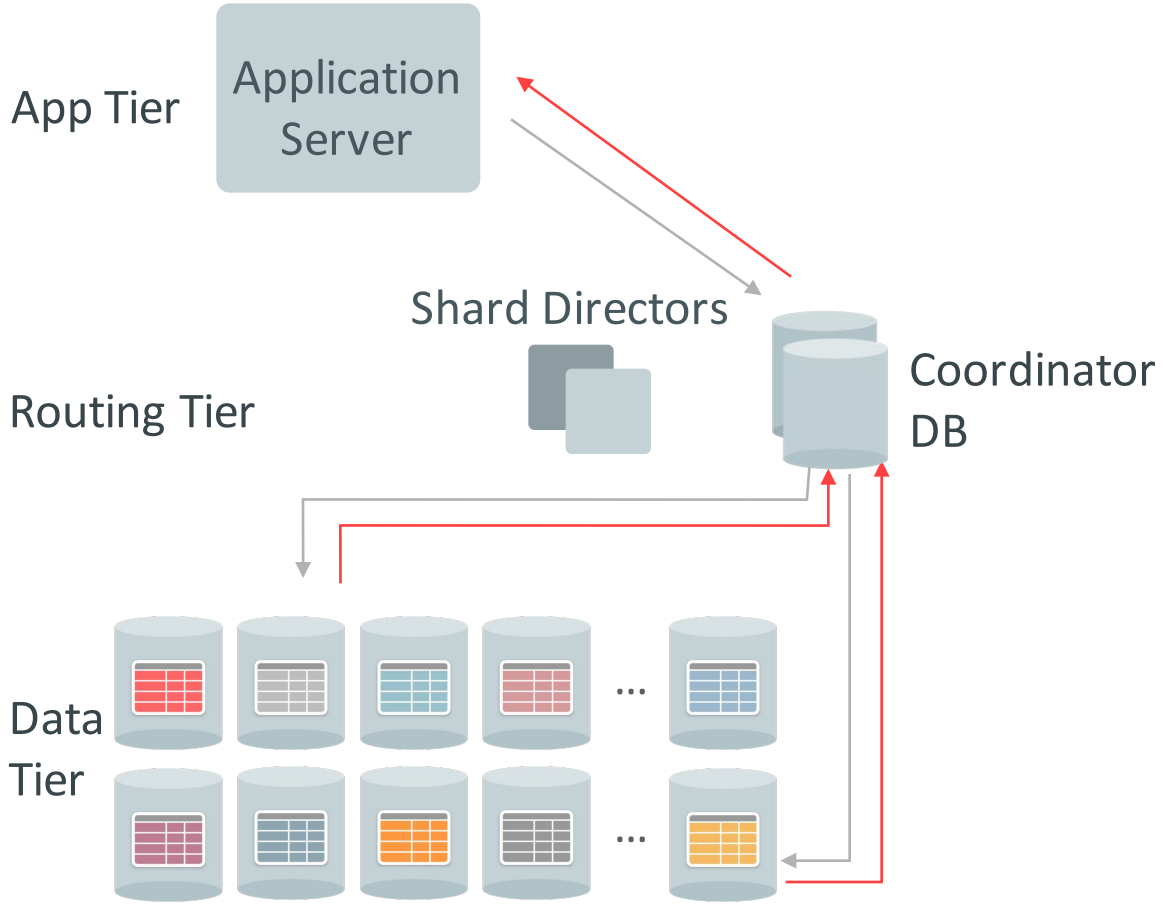


Copyright © 2015, Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 12.2

Non-Shard Key Access & Cross-Shard Queries

- If client does not pass shard key to Connection pool, the connection is made to the coordinator database
- Coordinator parses SQL and will proxy/route request to one or more shards
  - Supports shard pruning and scatter-gather
- For developer convenience and not for high performance
- Supports many but not all Queries
- No Update support



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. | Public

NEW IN 19c

EZConnect Improvements

- Simplification of Easy Connect syntax
- Easy Connect adaptor will now accept a list of name value pairs
  - For example: SDU, RETRY_COUNT, CONNECT_TIMEOUT, etc.)
- Will now enable multiple hosts/ports in the connect string
  - Typically used in load-balancing client connections.

```
$> sqlplus soe/soe@(DESCRIPTION=
(ADDRESS_LIST=
(LOAD_BALANCE=ON) (ADDRESS=(PROTOCOL=tcp)(HOST=salesserver1)(PORT=1522))
(ADDRESS=(PROTOCOL=tcp)(HOST=salesserver2)(PORT=1522))
(ADDRESS=(PROTOCOL=tcp)(HOST=salesserver3)(PORT=1522)))
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

NEW IN 19c

EZConnect Improvements

- Simplification of Easy Connect syntax
- Easy Connect adaptor will now accept a list of name value pairs
  - For example: SDU, RETRY_COUNT, CONNECT_TIMEOUT, etc.)
- Will now enable multiple hosts/ports in the connect string
  - Typically used in load-balancing client connections.

```
$> sqlplus soe/soe@//salesserver1,salesserver2,salesserver3:1522/sales.us.example.com
```

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

NEW IN 19c

EZConnect Improvements

- Simplification of Easy Connect syntax
- Easy Connect adaptor will now accept a list of name value pairs
  - For example: SDU, RETRY_COUNT, CONNECT_TIMEOUT, etc.)
- Will now enable multiple hosts/ports in the connect string
  - Typically used in load-balancing client connections.

```
$> sqlplus soe/soe@(DESCRIPTION=
(retry_count=3) (connect_timeout=60)(transport_connect_timeout=30)
(ADDRESS=(PROTOCOL=tcp)(HOST=salesserver1)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

NEW IN 19c

EZConnect Improvements

- Simplification of Easy Connect syntax
- Easy Connect adaptor will now accept a list of name value pairs
  - For example: SDU, RETRY_COUNT, CONNECT_TIMEOUT, etc.)
- Will now enable multiple hosts/ports in the connect string
  - Typically used in load-balancing client connections.

```
$> sqlplus soe/soe@//salesserver1:1521/sales.us.example.com?connect_timeout=60&
transport_connect_timeout=30&retry_count=3
```

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Program Agenda

1. Overview and History
2. Vision
3. Data Management Strategy
4. Oracle and Modern Development
5. Open Source initiatives
6. Developer centric functionalities
7. Q & A

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Integrated Cloud

Applications & Platform Services

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |


ORACLE®