



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Building Rule-Based OLTP Systems Using Oracle RDF

Dr. Philip Cannata and Nigel Jacobs

(<https://analyticsanddatasummit.org/techcastdays>)

Let's Put the Application Complexity in the Database!

(Business Rules, Data Partitioning, Application Provisioning)





Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Building Rule-Based OLTP Systems Using Oracle RDF

Links to the git repositories referenced in this presentation are in the Slack channel

MS-RDBMS
docs
oci
DEPT_VIEW.sql
EMP_RULE.sql
EMP_VIEW.sql
README.md

<https://github.com/CannataUT/RDF-OLTP>

Current Repository

Future Repository

<https://github.com/AnalyticsandDataOracleUserCommunity/Analytics/RDF-OLTP>

Questions can be addressed to: phil.cannata@oracle.com and nigel.jacobs@oracle.com

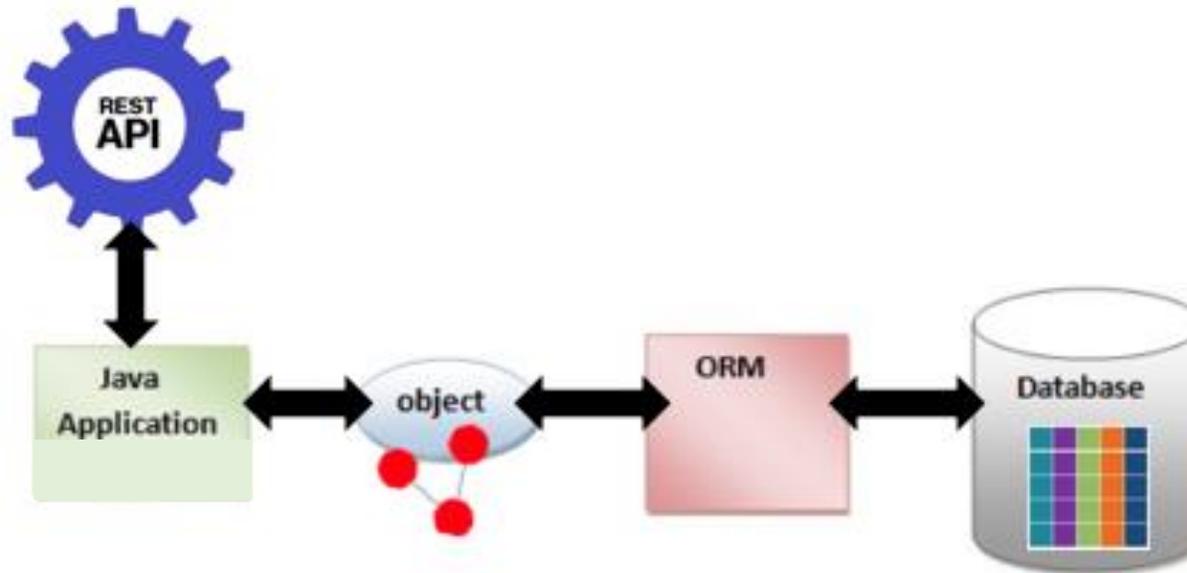


Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

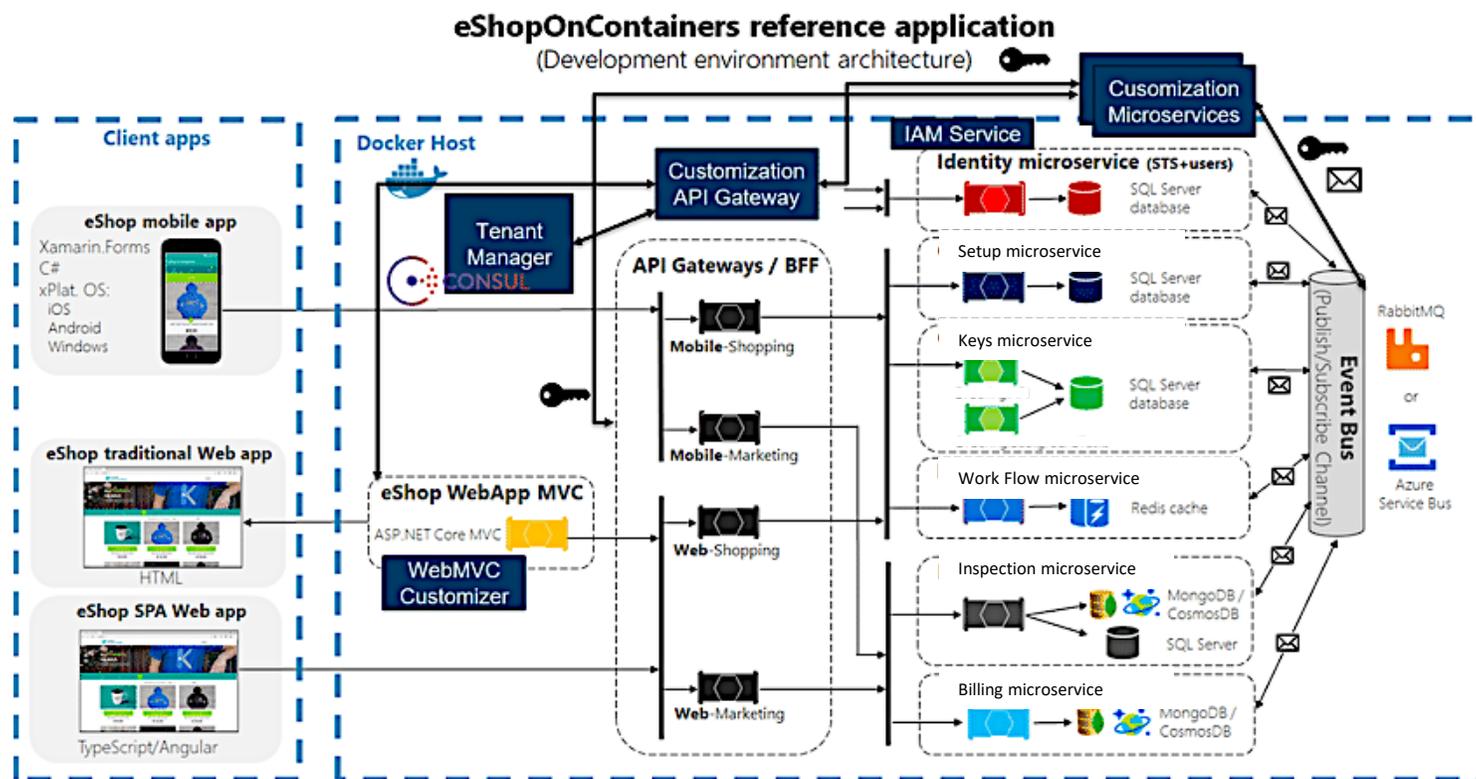
www.analyticsanddatasummit.org/techcastdays

Conventional Way to Think about Building OLTP (SaaS) Systems Today





This is Typically Cast as a Set of Microservices



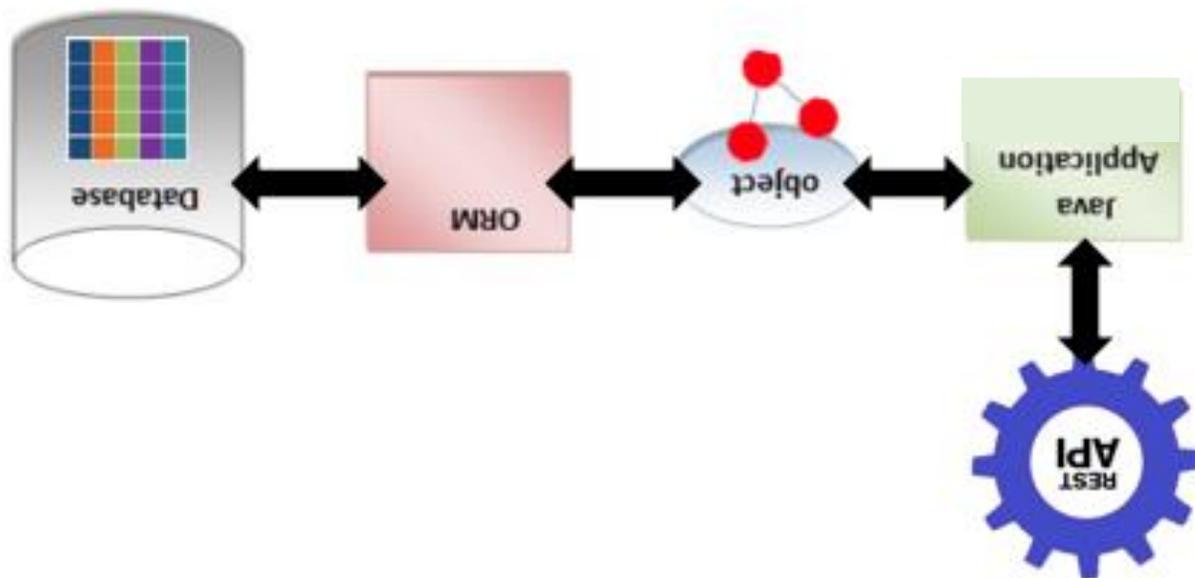


Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Let's Turn This Way of Thinking Upside Down



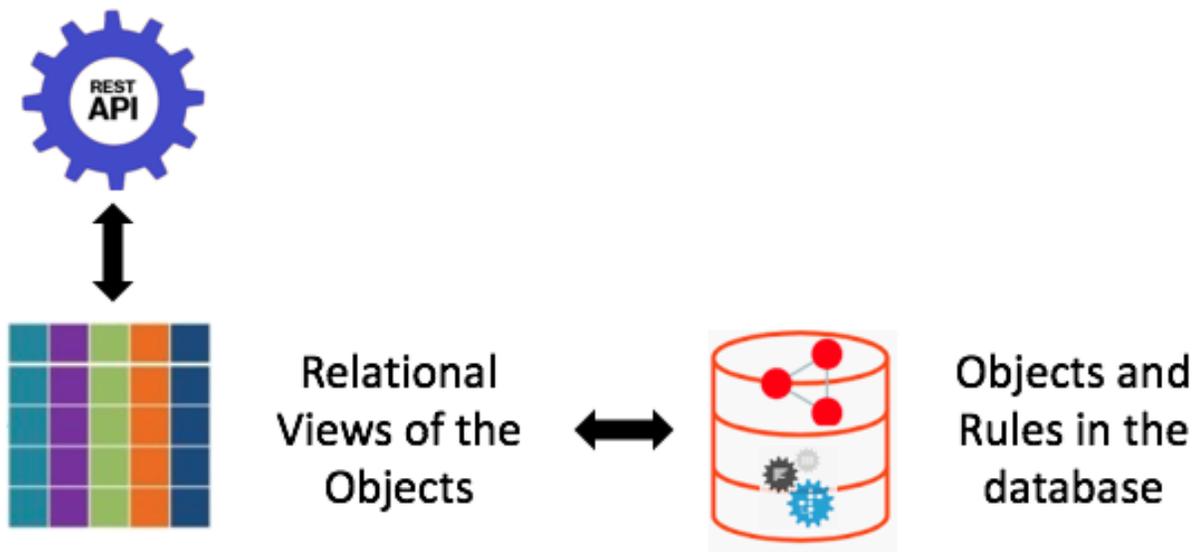


Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

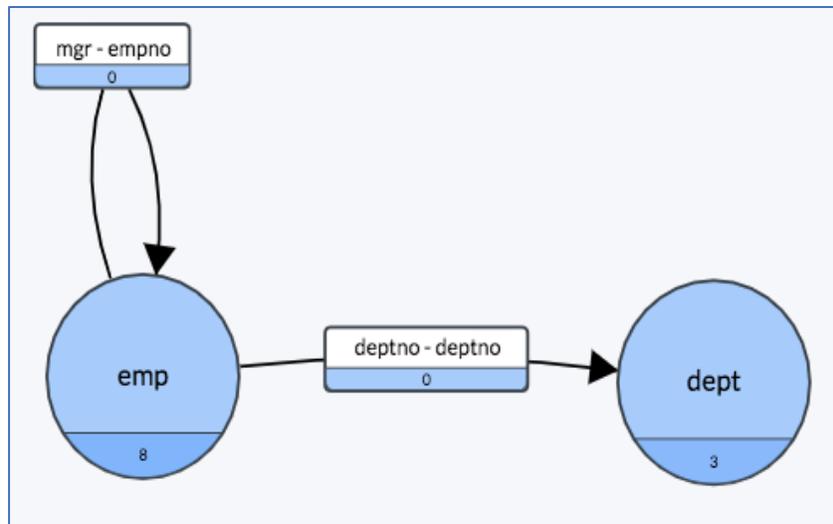
Building Rule-Based OLTP (SaaS) Systems Using Oracle RDF ([RDF-OLTP](#))





This Approach Will be Demonstrated Using a Simple Example Based on the Familiar Emp/Dept Schema

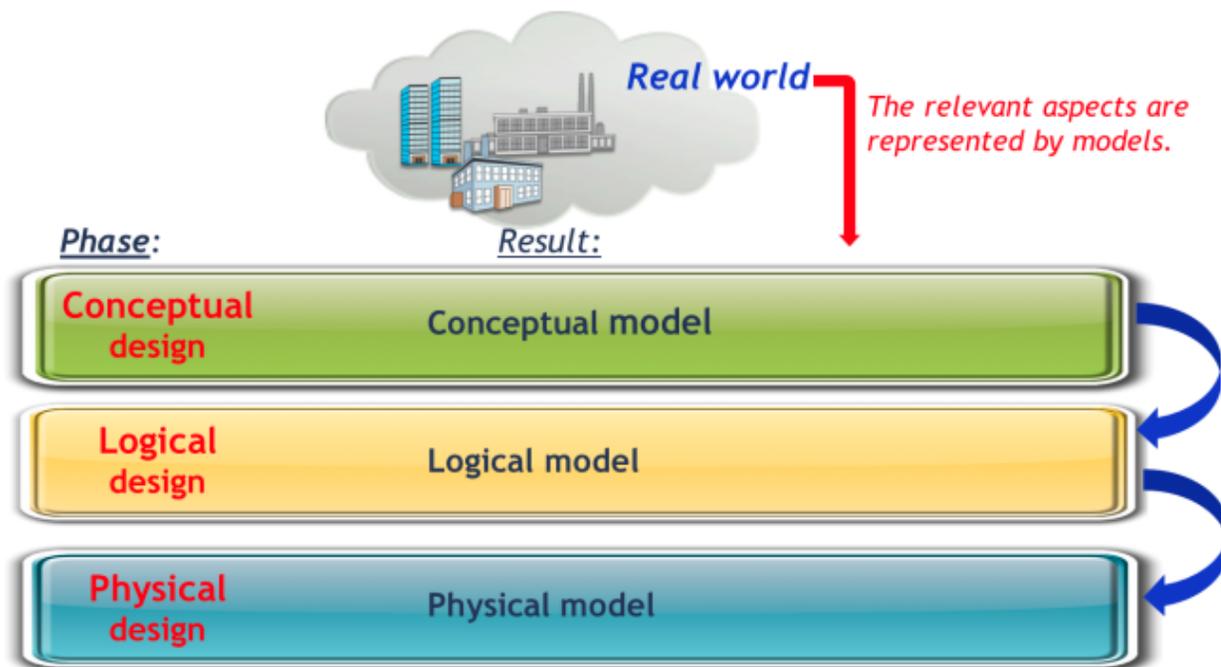
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800	(null)	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	(null)	10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000	(null)	20
7839	KING	PRESIDENT	(null)	17-NOV-81	5000	(null)	10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	(null)	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100	(null)	20
7900	JAMES	CLERK	7698	03-DEC-81	950	(null)	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	(null)	20
7934	MILLER	CLERK	7782	23-JAN-82	1300	(null)	10



DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



Traditionally, the Database for this Example Would be Built as Follows:



Each time something changes, you get to do this all over again. 😞

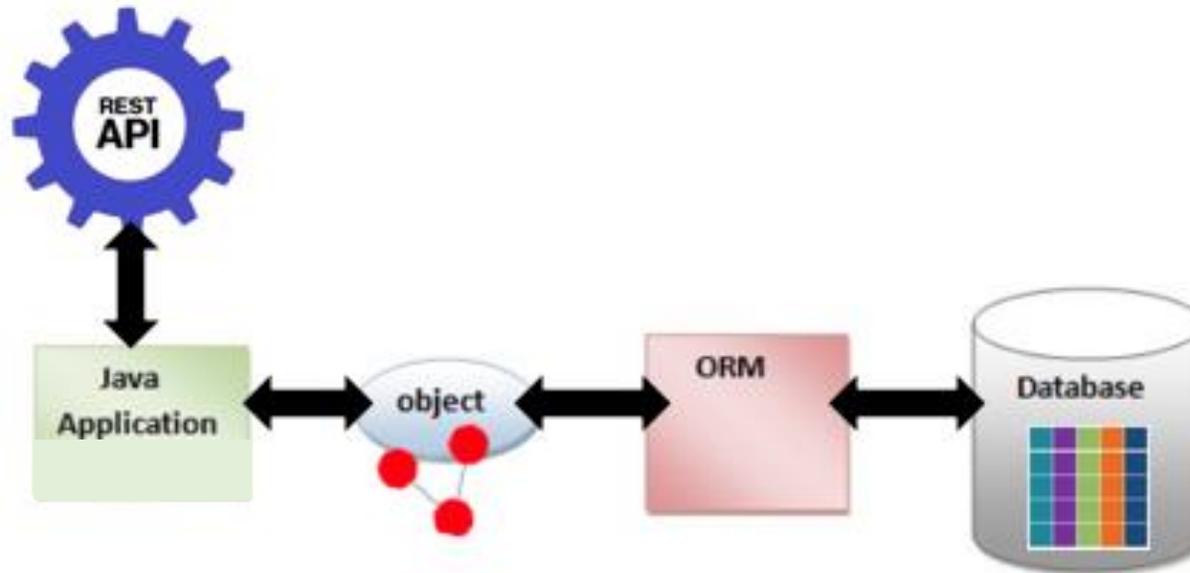


Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Then You Would Build Your Application as a Bunch of Java Code On Top of the Database





Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

With RDF-OLTP, You do the Following to Build the Database:



I.e., You don't Actually Build the Database because RDF is Schemaless

If something changes, you just change the View. No change is needed in the database. 😊



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

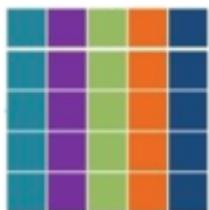
www.analyticsanddatasummit.org/techcastdays

Instead, You Build Relational Views on the Conceptual Model

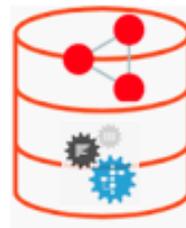
Then Build INSERT, UPDATE, and DELETE Triggers on These Views

(In Principle, the Views and Triggers could be generated based on a simple declarative set of specifications based on the Conceptual Model, i.e., this is a Low Code Model of Development)

The RDF Database is Created When You Insert Data Into It



Relational
Views of the
Objects



Objects and
Rules in the
database



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Then You Build Application in the Database as a Set of Rules



Here's an Example Application:

“If all of the employees in a department have taken a training course, the department is eligible for an award”

EMPNO	ENAME	JOB	SALARY	HIREDATE	SALARY_1	COMM	TRAINING	DNAME	LOCATION	AWARD
7934	MILLER	CLERK	1300	23-JAN-1982	1300	(null)	Yes	ACCOUNTING	NEW YORK	Department 10 is eligible for an award
7839	KING	PRESIDENT	5000	17-NOV-1981	5000	(null)	Yes	ACCOUNTING	NEW YORK	Department 10 is eligible for an award
7782	CLARK	MANAGER	2450	9-JUN-1981	2450	(null)	Yes	ACCOUNTING	NEW YORK	Department 10 is eligible for an award
7902	FORD	ANALYST	3000	3-DEC-1981	3000	(null)	(null)	RESEARCH	DALLAS	(null)
7788	SCOTT	ANALYST	3000	09-DEC-1982	3000	(null)	(null)	RESEARCH	DALLAS	(null)
7876	ADAMS	CLERK	1100	12-JAN-1983	1100	(null)	(null)	RESEARCH	DALLAS	(null)
7369	SMITH	CLERK	800	17-DEC-1980	800	(null)	(null)	RESEARCH	DALLAS	(null)
7566	JONES	MANAGER	2975	2-APR-1981	2975	(null)	(null)	RESEARCH	DALLAS	(null)
7900	JAMES	CLERK	950	3-DEC-1981	950	(null)	(null)	SALES	CHICAGO	(null)
7654	MARTIN	SALESMAN	1250	28-SEP-1981	1250	1400	(null)	SALES	CHICAGO	(null)
7844	TURNER	SALESMAN	1500	8-SEP-1981	1500	(null)	(null)	SALES	CHICAGO	(null)
7521	WARD	SALESMAN	1250	22-FEB-1981	1250	500	(null)	SALES	CHICAGO	(null)
7499	ALLEN	SALESMAN	1600	20-FEB-1981	1600	(null)	(null)	SALES	CHICAGO	(null)
7698	BLAKE	MANAGER	2850	1-MAY-1981	2850	(null)	(null)	SALES	CHICAGO	(null)



Here's the Application Code in the Database as a Rule:

```
INSERT { ?dept_uri saas:award ?eligible . }
```

```
WHERE
```

```
{ # Get the dept for each emp.
```

```
?emp_uri a saas:Emp .
```

```
?emp_uri saas:empDeptEVA ?dept_uri .
```

```
?dept_uri saas:deptno ?deptno .
```

```
MINUS # Remove depts that have at lease one emp who is not trained.
```

```
{ SELECT ?dept_uri
```

```
WHERE
```

```
{
```

```
?emp_uri a saas:Emp .
```

```
?emp_uri saas:empDeptEVA ?dept_uri .
```

```
OPTIONAL { ?emp_uri saas:training ?training. }
```

```
FILTER(?training = "")
```

```
}
```

```
}
```

```
# Construct the value to be inserted into the dept's award attribute.
```

```
BIND(CONCAT("Department ", ?deptno, " is eligible for an award") AS ?eligible)
```

```
}
```

Get a list of all departments that have employees.

Take a Department out of the List if an employee has not taken the training.

Create the desired insert value.



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Oracle's Implementation of RDF Has Some Unique Features That Allows These Rules Based Applications to be Built on an Enterprise Level

- Tight integration of SQL/PLSQL and RDF/SPARQL
- Integrates seamlessly with ORDS (Oracle REST Data Service)
- Built in Data Partitioning
- PDBs
- All OCI Services are available to the application, e.g., telemetry, events, health services . . .
- No "Provisioning" is needed other than compiling views in the PDB
- Very little DBA Assistance is needed



There is a Current OCI SaaS Customer That is Using All of This to Deploy a Very Successful Multitenant, Production Application.



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Here's the Same Application as a Microservice:



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

A 'Traditional' Microservice Approach to EmpDept (MS-RDBMS)

- Complex mix of mature and emerging technologies
- Helidon - microservice framework
- Java
- ADF
- JDBC/SQL
- Oracle 12c RDBMS

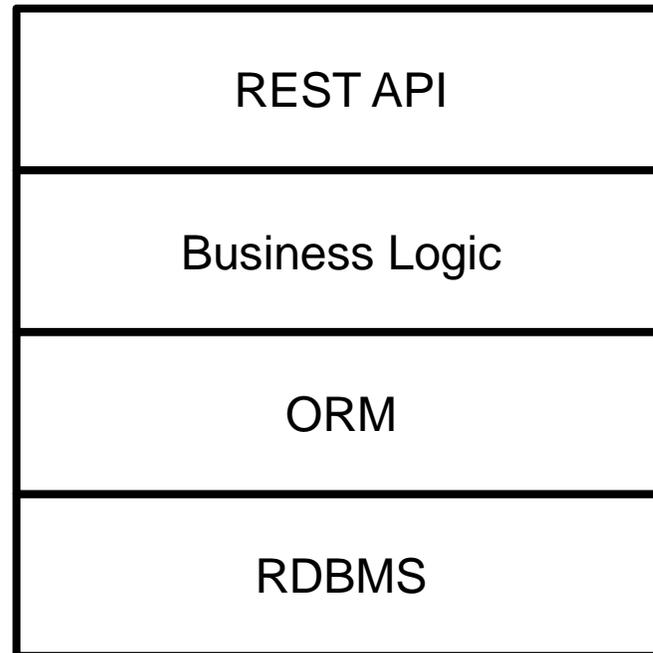


Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Architectural Layers within the Microservice





Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

REST API

- HTTP / JSON ↔ Java APIs / POJO Domain Objects
- List all employees as a list of JSON maps:
 - *curl -X "GET <http://localhost:8080/Emp>"*
- Get a single employee given the employee name, as a JSON map:
 - *curl -X "GET <http://localhost:8080/Emp/name=Nigel%20Jacobs>"*
- Create an employee:
 - *curl -X "PUT <http://localhost:8080/Emp/name=Nigel%20Jacobs&salary=123...>"*
- Routing, error handling, serialization



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Business Logic

- High level actions on business types (repositories)
- Business semantics
- Java / POJOs
- Pass-through: majority of logic pulled down into ORM



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Business Logic

EmployeeRepository:

```
public Employee find(Integer employeeNumber) // find an employee by emp no
public Collection<Employee> loadAllEmployees() // load all employees
```

DepartmentRepository:

```
public void save(Department department) // save department
```



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

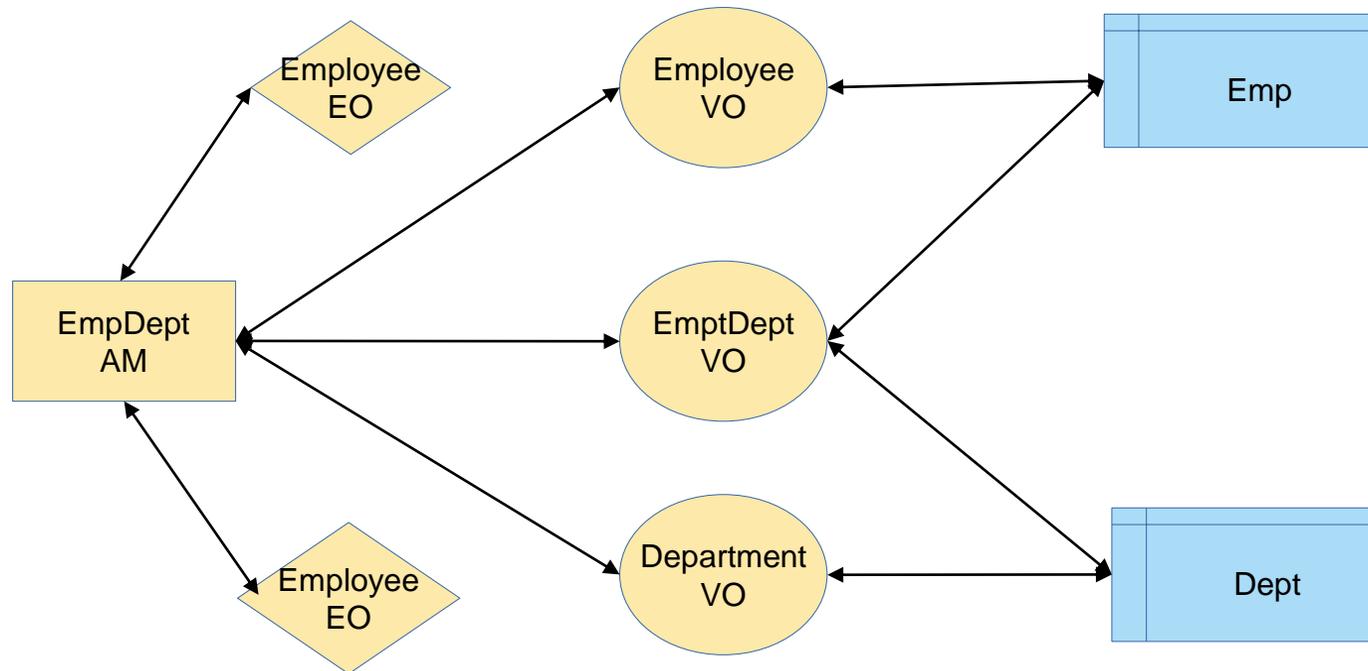
www.analyticsanddatasummit.org/techcastdays

Object Relational Mapper (ORM)

- ADF entities and mapping: EOs, AMs, Vos
- Built extended from DB schema via Jdev UI at design time
- Generated Java stubs, XML files
- POJO ↔ JDBC/SQL serialization
- Caching to reduce DB hits



Object Relational Mapper (ORM)





Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Object Relational Mapper (ORM)

```
public class EmpDeptAMImpl extends OAAApplicationModuleImpl {  
  
    /**  
     * Find a specific employee, by employee number  
     */  
    public Employee findEmployee(Integer empNo) {  
  
        CachedMapper mapper = new CachedMapper();  
        loadEmployeesIntoCache(empNo, mapper);  
        Integer deptNo = mapper.getDeptNo(empNo);  
        if (deptNo != null) {  
            // if employee has a dept, load it's data  
            loadDepartmentsIntoCache(deptNo, mapper);  
        }  
        mapper.finalize(); // apply business logic  
        return mapper.getEmployee(empNo);  
  
    } ...  
}
```



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Object Relational Mapper (ORM)

```
private void loadEmployeesIntoCache(Integer empNo, CachedMapper mapper) {

    EmpDeptVOImpl vo = getEmpDeptVO();
    RowSet rs = vo.getDefaultRowSet();

    if (empNo != null) {
        // apply 'EMP_NO = empNo' filter
        ViewCriteria vc = vo.getViewCriteria(EmpDeptVORowImpl.VC_SpecifyEmpVC);
        rs.ensureVariableManager().setVariableValue(EmpDeptVORowImpl.VAR_Empno, empNo);
        vo.appendViewCriteria(vc);
    }

    // execute query
    String q = vo.getQuery();
    AppsLogger.write(this, "Query: " + q, AppsLogger.FINE);
    vo.executeQuery();
    // read row data from query into mapper
    mapper.readRowDataIntoCache(vo);
}
```



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Object Relational Mapper (ORM)

```
/**
 * Pull just employee data through cache
 *
 * @param row
 * @return
 */
private Employee pullEmpDataThroughCache(EmpDeptVORowImpl row, Integer empNo) {
    // pull employee from cache
    Employee emp = mapEmpNoToEmployee.get(empNo);
    // if not in cache, map and put into cache
    if (emp == null) {
        // map to domain object
        emp = mapToEmployee(row);
        // put in cache
        mapEmpNoToEmployee.put(empNo, emp);
    }
    return emp;
}
```



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Object Relational Mapper (ORM)

```
/**
 * Map to employee domain object
 * @return
 */
private Employee mapToEmployee(EmpDeptVORowImpl row) {

    Employee employee = new Employee();
    employee.setEmployeeNo(row.getEmpno());
    employee.setName(row.getEname());
    employee.setJob(row.getJob());
    employee.setManager(row.getMgr());
    employee.setHireDate(row.getHiredate());
    employee.setSalary(row.getSal());
    employee.setComm(row.getComm());
    employee.setTraining(row.getTraining() != null && row.getTraining() == 1);
    // cache empNo -> employee
    mapEmpNoToDeptNo.put(employee.getEmployeeNo(), row.getDeptno());
    return employee;
}
```



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Object Relational Mapper (ORM)

```
public void finalize() {  
  
    // materialize the object links: iterate over employees, and for each employee,  
    // set the department link to the cached department object  
    for (Integer empNo : mapEmpNoToEmployee.keySet()) {  
        Integer deptNo = mapEmpNoToDeptNo.get(empNo);  
        Employee emp = mapEmpNoToEmployee.get(empNo);  
        Department dept = mapDeptNoToDepartment.get(deptNo);  
        emp.setDepartment(dept);  
    }  
  
    // For given employee and her department, calculate the department award  
    // based on the employees training and the departments existing award value  
    for (Employee emp : mapEmpNoToEmployee.values()) {  
        Department dept = emp.getDepartment();  
        boolean training = emp.hasTraining();  
        // dept has award until at least one employee without training  
        boolean award = (dept.hasAward() == null) || dept.hasAward();  
        dept.setAward(award && training);  
    }  
}
```



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Relational DBMS

- Oracle 12c
- SQL:

```
SELECT * FROM EMP, DEPT WHERE EMP.DEPTNO = EMP.DEPTNO
```

- DDL / DML:

```
CREATE TABLE EMP (EMPNO NUMBER(7) NOT NULL, ..
```

- Datasource definitions: host/service/user/password



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Findings from Microservice Approach

- Multiple formalisms and develop tools exponentially slows dev cycle time
- Multiple, heterogonous layers increases code complexity
- Loose-coupling increase network traffic: 2 DB hits / employee
- Multi-tenancy & partitioning: architecture complicates provisioning
- Scalability: complicated by architectural complexity



Analytics and Data
ORACLE USER COMMUNITY

TechCast Days

www.analyticsanddatasummit.org/techcastdays

Building Rule-Based OLTP Systems Using Oracle RDF

Summary

- Simple Schema Management.
- Business Logic is in one place – the database. This is a very “Low Code” solution.
- No Java, ORM, XML, Docker, Kubernetes, etc. required.
- No “Provisioning” is needed other than compiling views in the PDB. I.e., no need for CI / CD, Jenkins.
- Data Partitioning is built-in.
- Low DBA overhead required.
- All of the Enterprise features of Oracle and OCI are available – PDBs, backup and recovery, high availability, disaster recovery, scheduling, resource management, telemetry, events, health services . . .
- The OLTP Database can become the OLAP Database by just adding OWL constructs, i.e., **add Semantics**
- **This is an ideal environment for startups.** RDF/SPARQL is even familiar to computer science graduates because of their exposure to the Clojure Programming language and Datomic Database on AWS.
And, they don't have a body of legacy code that they have to maintain and move forward.