

ORACLE®

# TRANSFORMAR O MUNDO EMPODERANDO AS PESSOAS POR MEIO DA INOVAÇÃO

# Quantas línguas você fala? Mas sua persistência, já é poliglóta?

**Luciano Verissimo**

Principal Solution Engineer

[Luciano.verissimo@oracle.com](mailto:Luciano.verissimo@oracle.com)



lucianoverissimo

Jan, 2019

**ORACLE®**

# 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.

1

Uma visão de como chegamos na  
Persistência Poliglota?

# Monolítico, Procedural, Do-It-Yourself



```
Sessão A - [24 x 80]
Arquivo Editar Exibir Comunicação Ações Janela Ajuda
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT AEPF001.IFRJ.SOURCE(COBX01) - 01.09 Columns 00001 00072
Command ==> Scroll ==> CSR
=COLS> -----1-----2-----3-----4-----5-----6-----7--
***** ***** Top of Data *****
000100 ID DIVISION.
000200 PROGRAM-ID. COB01.
000300 * ESTE PROGRAMA CALCULA CONFORME A OPERAÇÃO INFORMADA
000500 DATA DIVISION.
000600 WORKING-STORAGE SECTION.
000700 01 N1 PIC 9(09).
000800 01 N2 PIC 9(09).
000900 01 N3 PIC 9(09).
001000 01 OP PIC X(01).
001400 PROCEDURE DIVISION.
001410 DISPLAY " EDICAO "
001500 DISPLAY " ----- INICIO -----"
001510 DISPLAY " PRIMEIRO PROGRAMA COBOL "
001600 ACCEPT N1.
001700 ACCEPT OP.
001710 ACCEPT N2.

F1=Help F2=Split F3=Exit F4=Expand F5=Rfind F6=Rchange
F7=Up F8=Down F9=Swap F10=Left F11=Right F12=Cancel

MA A 04/015
Conectado ao servidor remoto/host 127.0.0.1 utilizando lu/pool TCPLN015 e porta 22
```

Todas as declarações do programa Cobol devem começar na coluna 8.

Qualquer declaração fora da coluna 8 dá um erro na geração do programa.

DAY	MONTH	YEAR
1	jan	1980

Hey Doc, o que é isto?

Um capacitor de fluxo, com ele será possível a viagem no tempo...

Não Doc... isto aqui?

**Ah, este é primeiro banco relacional comercialmente viável  
com suporte à SQL**



# A Inovação: SQL Joins

```
SELECT A.COL  
FROM A, B  
WHERE A.COL = B.COL
```



# A.I.D.

YEAR

1983

## ATOMICIDADE

Uma transação tem vários comandos distintos que forma uma única unidade. Ou faz tudo, ou não faz nada.

## ISOLAMENTO

Há 1 unidade no estoque.  
Duas compras ao mesmo tempo?

Isto não existe... A primeira executa. Se efetivar, a segunda (em espera) tem falta de estoque. Se a primeira falhar, a segunda ainda vê 1 unidade no estoque.

## DURABILIDADE

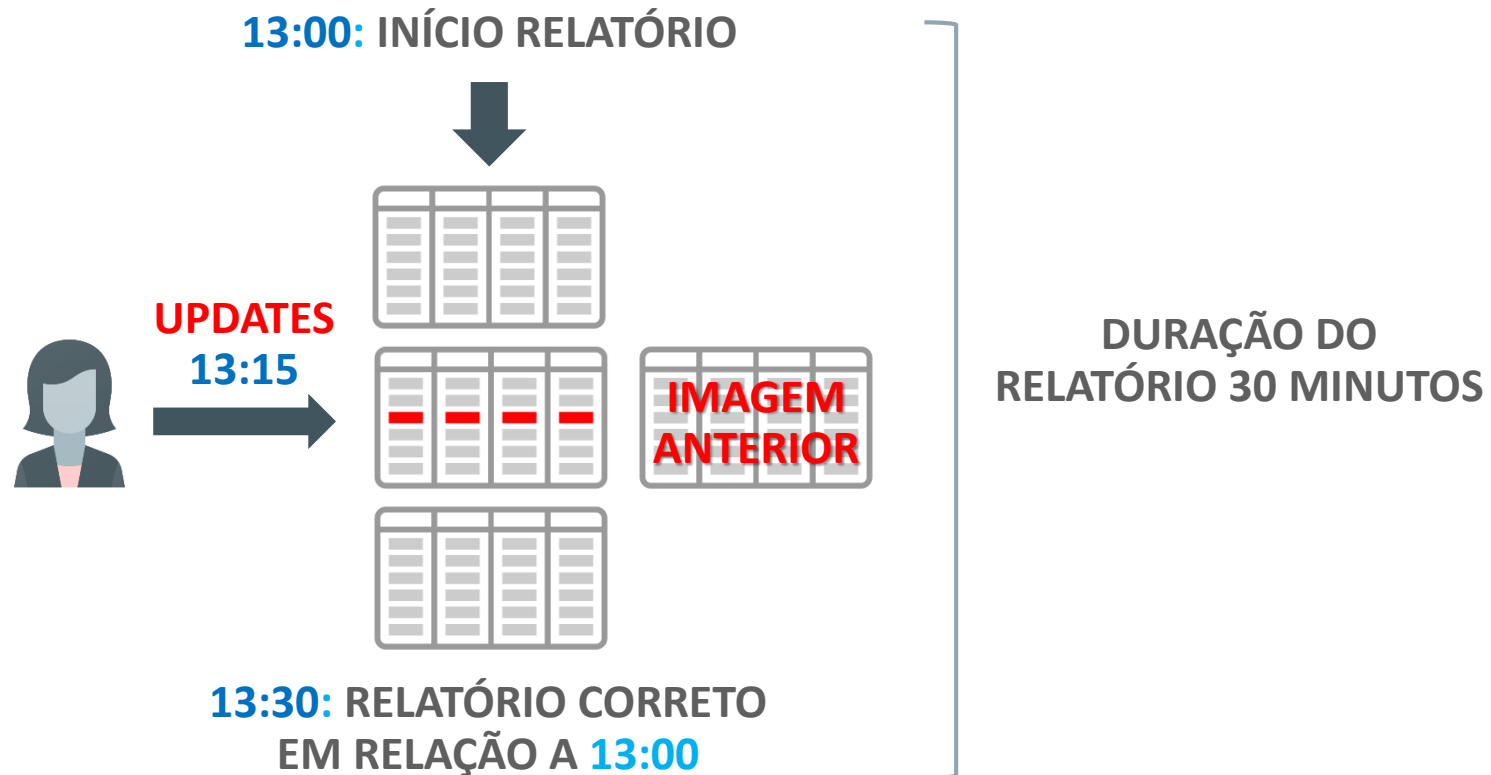
Uma transação modifica a quantidade no estoque. A quantidade será sempre a mesma até que outra transação válida altere para outro valor... mesmo que o DBMS seja abruptamente reiniciado.



# Leitura Consistente

YEAR

1984

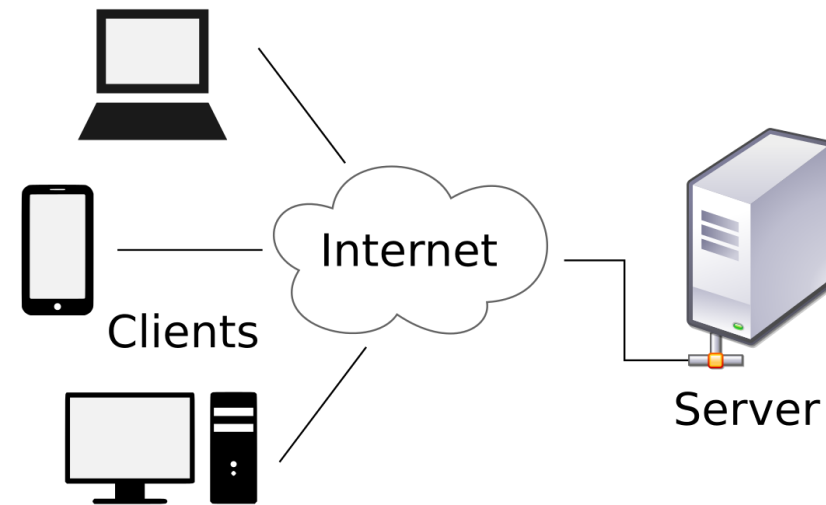


# Client/Server

YEAR

1985

Primeiro Banco de Dados  
Client/Server



# Easy\*Link (DB Links)

YEAR

1986

Não havia REST, SOAP, SOA, EAI, WEBSERVICE... ah, aplicações em 2 camadas ainda era novidade.

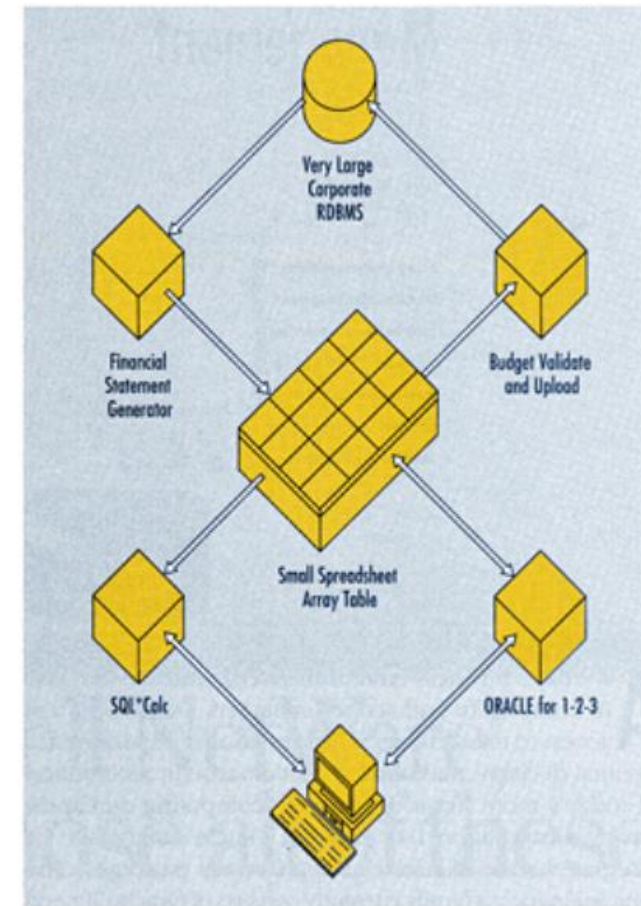


Figure 3 Oracle General Ledger's spreadsheet interface allows users to perform budgeting on a personal computer.

# A.C.I.D.

YEAR

1992

## ATOMICIDADE

Uma transação tem vários comandos distintos que forma uma única unidade. Ou faz tudo, ou não faz nada.

## ISOLAMENTO

Há 1 unidade no estoque.  
Duas compras ao mesmo tempo?

Isto não existe... A primeira executa. Se efetivar, a segunda (em espera) tem falta de estoque. Se a primeira falhar, a segunda ainda vê 1 unidade no estoque.

## DURABILIDADE

Uma transação modifica a quantidade no estoque. A quantidade será sempre a mesma até que outra transação válida altere para outro valor... mesmo que o DBMS seja abruptamente reiniciado.

## CONSISTÊNCIA

Uma transação no banco de dados ocorre e ela respeita a unicidade da chave primária, o valor que não pode ser nulo, a referência da chave estrangeira, o datatype e quaisquer outras regras definidas na tabela

# Resumo: Tudo era para o Desenvolvedor

*Porque o Desenvolvedor tinha que fazer tudo na mão!*

## Versão 3



Commit e Rollback, mas sem leitura consistente

## Versão 4



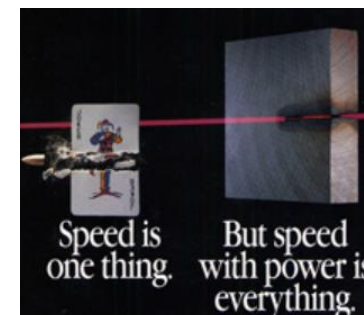
Leitura Consistente, Oracle rodando com 640k

## Versão 5



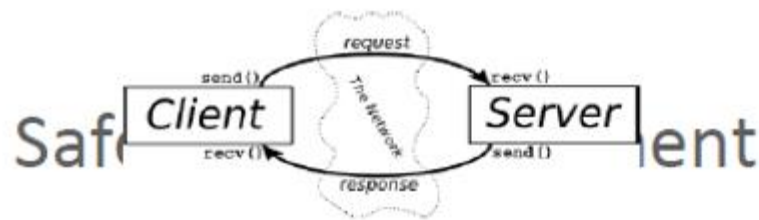
Client/Server, DB Links, e Cluster

## Versão 6



Constraints e Lock de Linha, PL/SQL anônimo

# A era da internet e a análise de dados



Release

1985 – 1997: 6, 7 and 8

1998 – 2012: 8i, 9i, 10g, 11g

Consistência e  
Integridade

Manipulação e  
Análise de dados

# SQL - Consultas Analíticas

YEAR

2007

```
SELECT
  id,
  name
  salary,
  salary - LAG (SALARY, 1) OVER (ORDER BY hire_date)
                                     as diff_sal_linha_anterior,
  salary - AVG (salary)
                OVER (PARTITION BY dept_id)
                                     as diff_avg_sal_dept
FROM employees a
```

# SQL - Machine Learning + R (in DB)

YEAR

2007

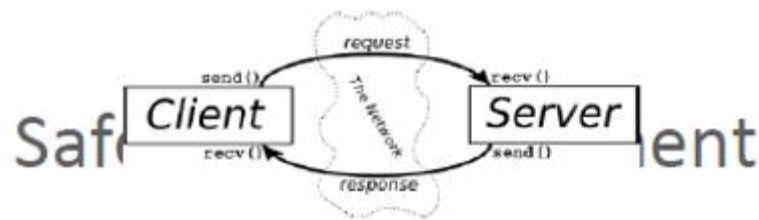
```
SELECT
  a.data,
  a.nm_loja,
  REGR_SLOPE(SYSDATE-a.data, b.qtd * c.vl_venda)
    OVER (PARTITION BY a.nm_loja) AS slp,
  REGR_INTERCEPT(SYSDATE-a.data, b.qtd * c.vl_venda)
    OVER (PARTITION BY a.nm_loja) AS icpt,
  REGR_R2(SYSDATE-a.data, b.qtd * c.vl_venda)
    OVER (PARTITION BY a.nm_loja) AS rsqr,
  REGR_COUNT(SYSDATE-a.data, b.qtd * c.vl_venda)
    OVER (PARTITION BY a.nm_loja) AS cnt
FROM venda a, item_venda b, produto c
WHERE a.id = b.venda_id
      AND b.id_produto = c.id
      AND a.data >= ?
```



# SQL - Reconhecimento de Padrões

```
SELECT *  
  FROM venda a, item_venda b  
 WHERE a.data = ? AND a.id = b.item_id)  
MATCH_RECOGNIZE  
 (PARTITION BY nm_loja  
  ORDER BY data  
  MEASURES FIRST(x.data)  
              first_t,  
              y.data last_t,  
              y.amount amount  
  
  PATTERN ( x{3,} y ) DEFINE  
  x AS (b.qtd < 2000),  
  y AS (b.qtd >= 1000000  
        AND LAST(x.data) - FIRST(x.data) < 3  
        AND y.data - LAST(x.data) < 10  
  ) ;
```

# Não existe “*One-Size-Fits-All*”



Release 1985 – 1997: 6, 7 and 8

1998 – 2012: 8i, 9i, 10g, 11g

2013 - 2015: 12c

Consistência e  
Integridade

Manipulação e  
Análise de dados

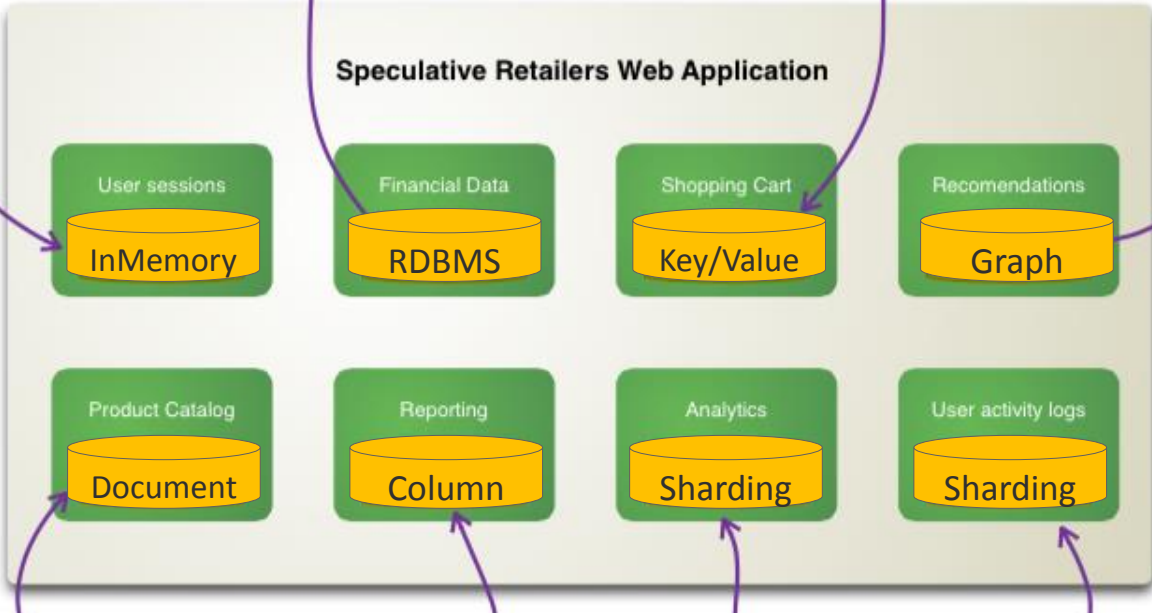
Flexibilidade e  
Escalabilidade



2

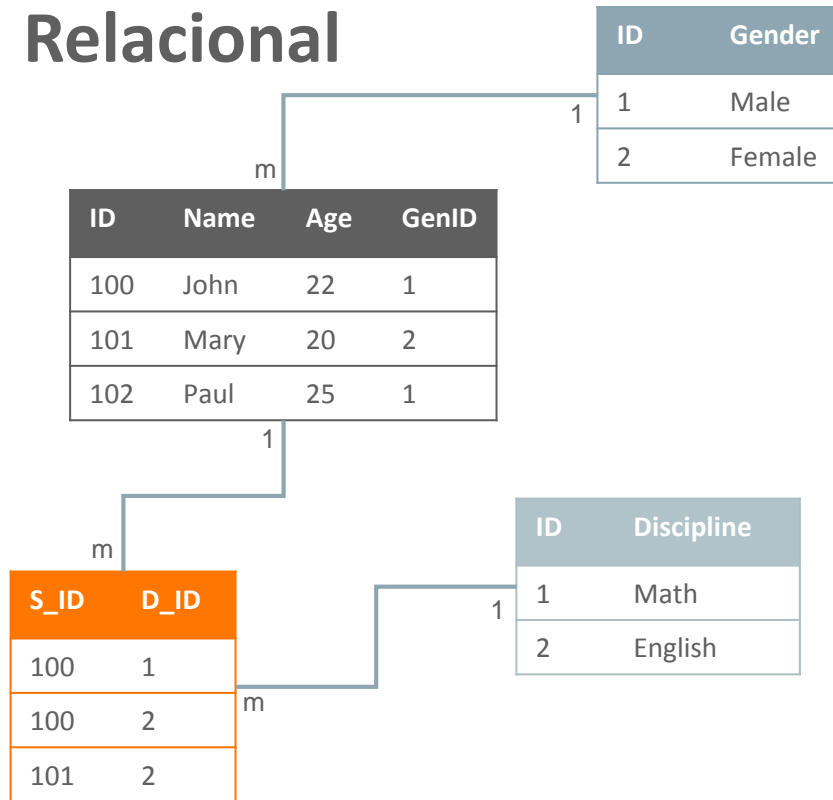
## Navegando pela persistência poliglota

# Documento, grafo, Shard... o que mais?

- Rápido acesso para leituras e escritas.
  - Não precisa ser durável
  - Precisa de atualizações transacionais (ACID)
  - Precisa de alta disponibilidade em múltiplas localizações, permite escrita inconsistente
  - Precisa atravessar rapidamente links entre amigos, compras de produtos e avaliações
- 
- The diagram illustrates a 'Speculative Retailers Web Application' architecture. It features eight data stores arranged in a 2x4 grid, each with a specific database technology and associated characteristics:
- User sessions** (InMemory): Fast access for reads and writes; does not need to be durable.
  - Financial Data** (RDBMS): Requires transactional updates (ACID).
  - Shopping Cart** (Key/Value): Requires high availability across multiple locations and allows for inconsistent writes.
  - Recomendations** (Graph): Requires fast traversal of links between friends, purchases, and product reviews.
  - Product Catalog** (Document): Many reads, few writes; data is grouped.
  - Reporting** (Column): SQL is friendly for reporting tools.
  - Analytics** (Sharding): Analytics at large scale in a cluster.
  - User activity logs** (Sharding): High volume of writes across multiple nodes.

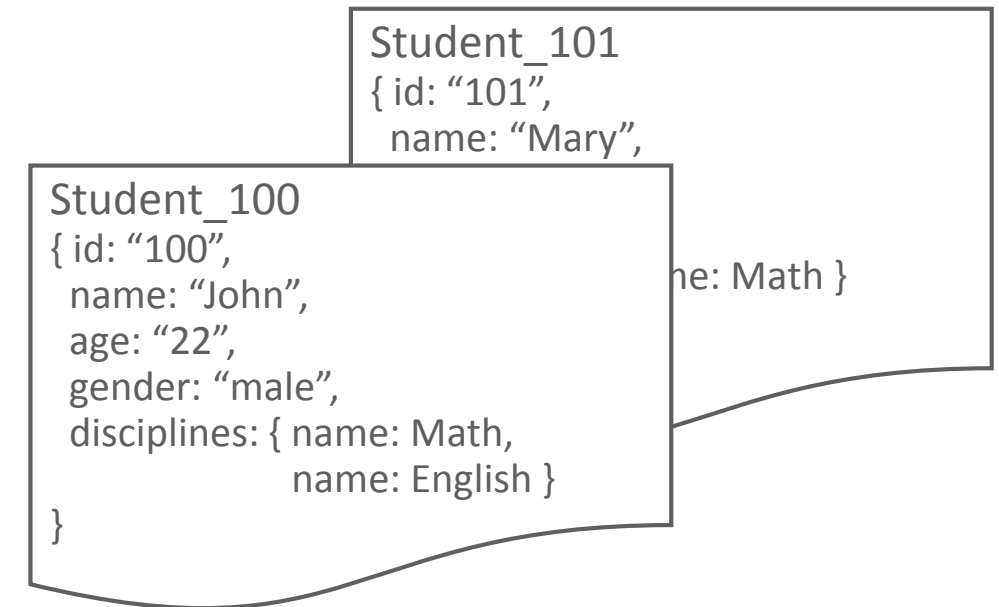
# RDBMS vs. Modelo de Documento

## • Relacional



ACID, **Schema on Write**, normalização (armazena o dado só uma vez) e SQL, **FOCO EM INTEGRIDADE**

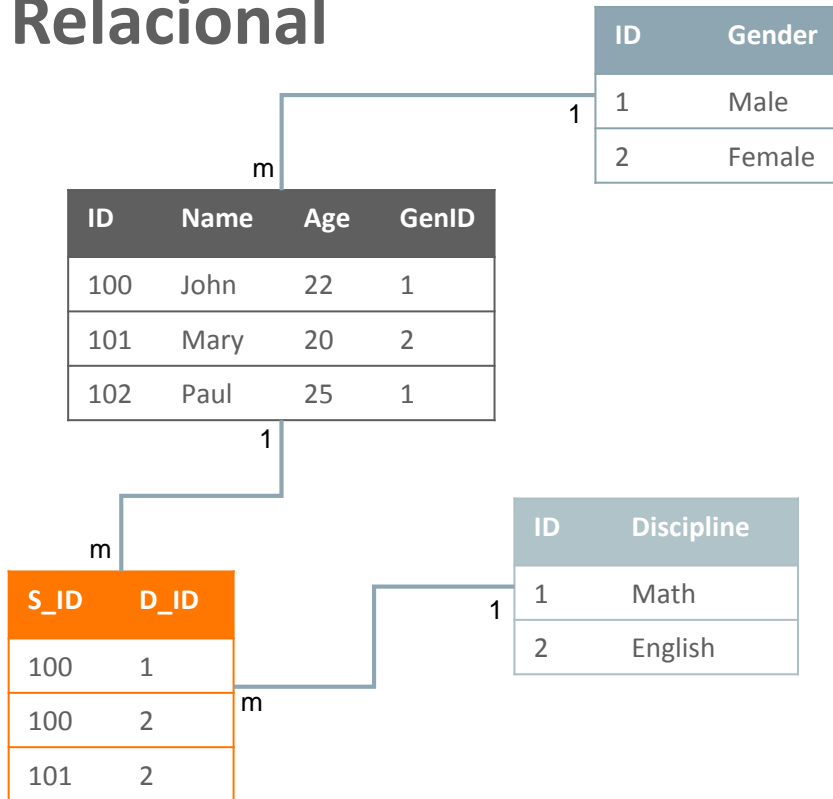
## • Documento (XML, JSON)



**Schema on Read**, SQL-like, self-contained, **FOCO EM FLEXIBILIDADE**

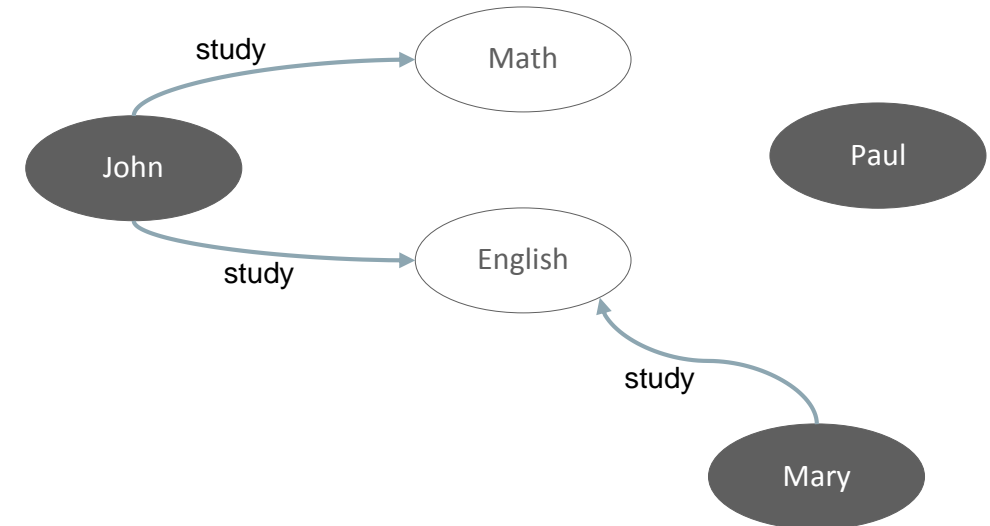
# RDBMS vs. Modelo de Grafo

- Relacional



ACID, Schema on Write, normalização (armazena o dado só uma vez) e SQL, **FOCO EM INTEGRIDADE**

- Grafo



Schema relativamente flexível, **FOCO EM CORRELACIONAMENTOS**

# RDBMS vs. Modelo Chave/Valor

- Relacional

key	Index	Index	Index
ID	Name	Age	Gender
100	John	22	Male
101	Mary	20	Female
102	Paul	25	Male

FK



ACID, Schema on Write, normalização (armazena o dado só uma vez) e SQL, **FOCO EM INTEGRIDADE**

- Chave/Valor

key	Value		
ID	Name	Age	Gender
100	John	22	Male
101	Mary	20	Female
102	Paul	25	Male

Schema on Read, **FOCO EM ALTO THROUGHPUT PARA LEITURA E ESCRITA**, geralmente operado por **GET**, **PUT** e **DELETE** através de Primary Key.

# Aplicações Diferentes tem Necessidades Diferentes

		User Sessions	Financial Transactions	Shopping Cart	Recommend. Engine	Product Catalog	Reporting	Analytics	Activity Logs
Processing	Heavy Writes								√
	Heavy Reads				√	√	√	√	
	Fast Read/Write	√							
	Data Consistency		√	√					
	Data Durability		√						
	Analytic						√	√	
	Graph				√				
	Spatial								
	Geo Distribution			√		√			√
Data	Relational		√				√	√	
	Key/Value	√		√					√
	Document/JSON					√			√
	Graph				√				



# Duas Possibilidades para Persistência Poliglota

## Single-model

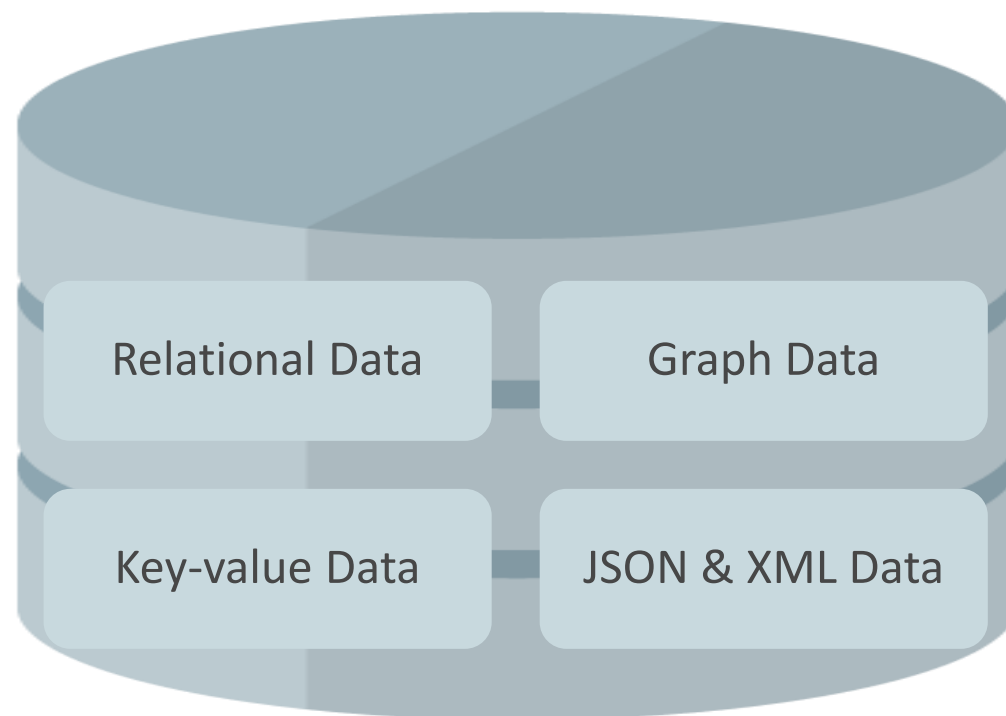
Relational  
Database

Graph Database

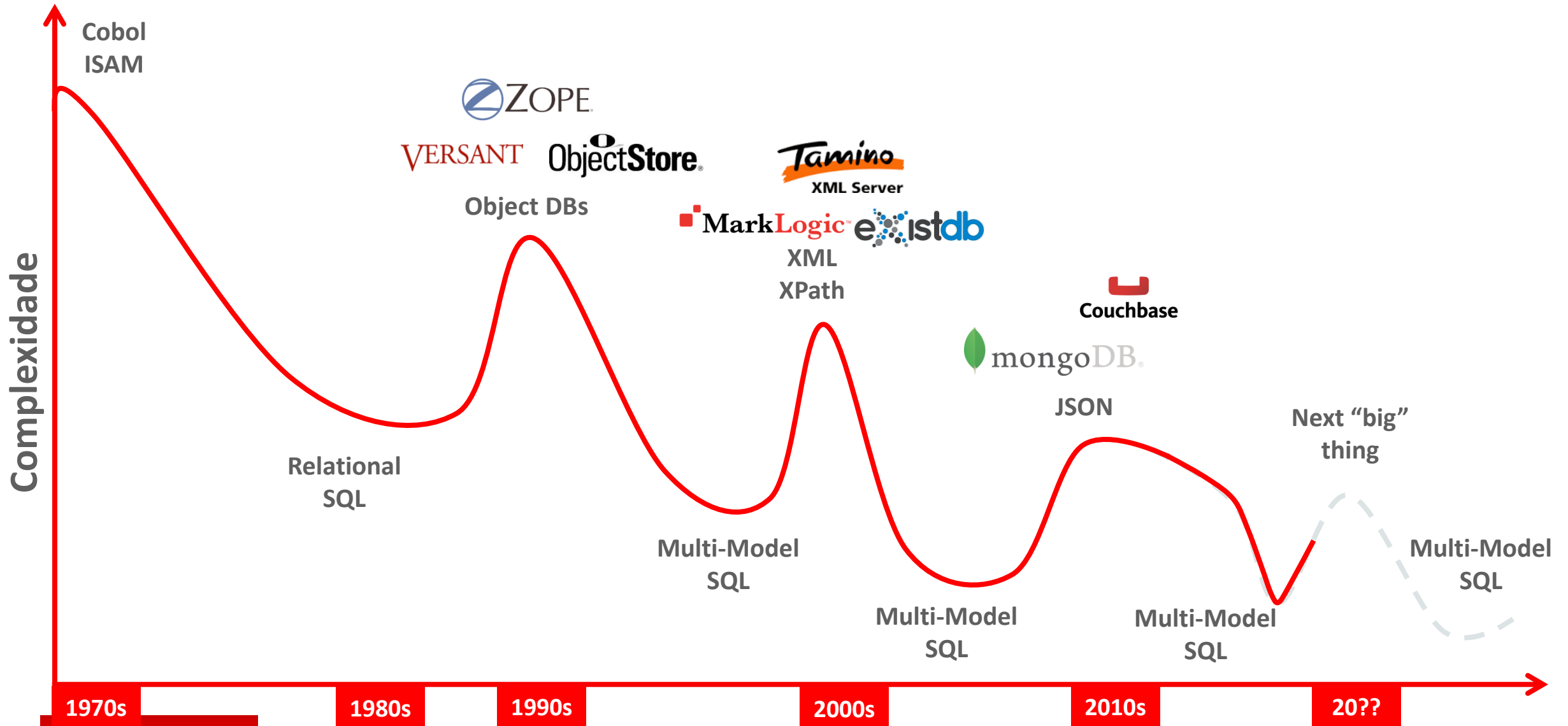
Key-value  
Store

JSON/XML  
Database

## Multi-model



# A importância do Multi-Model

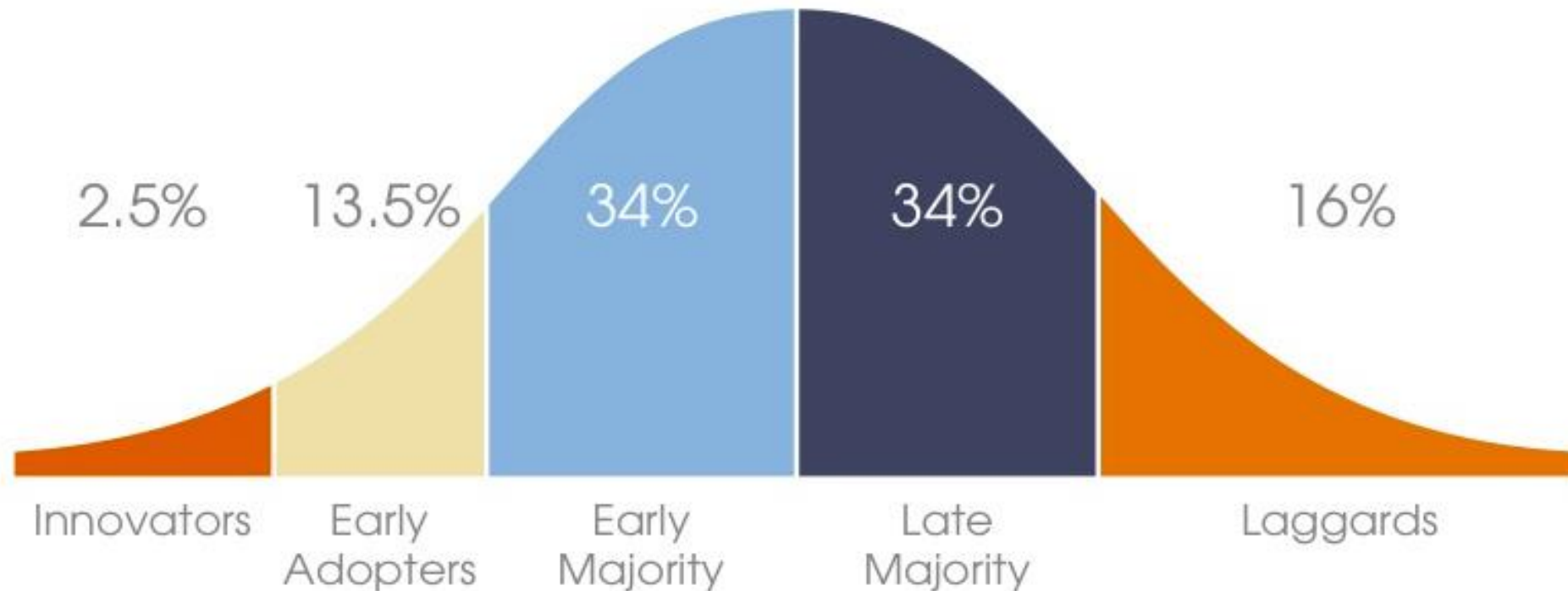


# Tendências da Persistência Poliglota

- Soluções **single-model** são mais comuns para aplicações “**de ponta**” ou “**Cutting-Edge Applications**”;
- Aplicações de negócio naturalmente convergem para **multi-model**
  - As aplicações “de ponta” de hoje serão as aplicações mainstream de amanhã
  - Com o tempo a eficiência do multi-model supera as vantagens do single-model especializado;
- **Sempre haverá soluções single-model**

# Technology Adoption Lifecycle

Entendendo o perfil das empresas em relação à adoção de novas tecnologias



# Em resumo...

**Não há uma resposta correta sobre qual é a melhor arquitetura, há apenas trade-offs**

- Qual arquitetura suporta seus requisitos?
  - Relacional, JSON, XML, ACID
- Você tem diferentes requerimentos de performance/escalabilidade?
  - Portal de compras com acesso na internet vs pagamento back-end
- Precisa compartilhar dados importantes?
  - Informação sobre saúde de um paciente, ou informações financeiras de um cliente
- Precisa compartilhar controles operacionais comuns?
  - Modelos de segurança comuns, user profiles, integrações comuns com outros sistemas, requerimentos de HA comuns, administração comuns

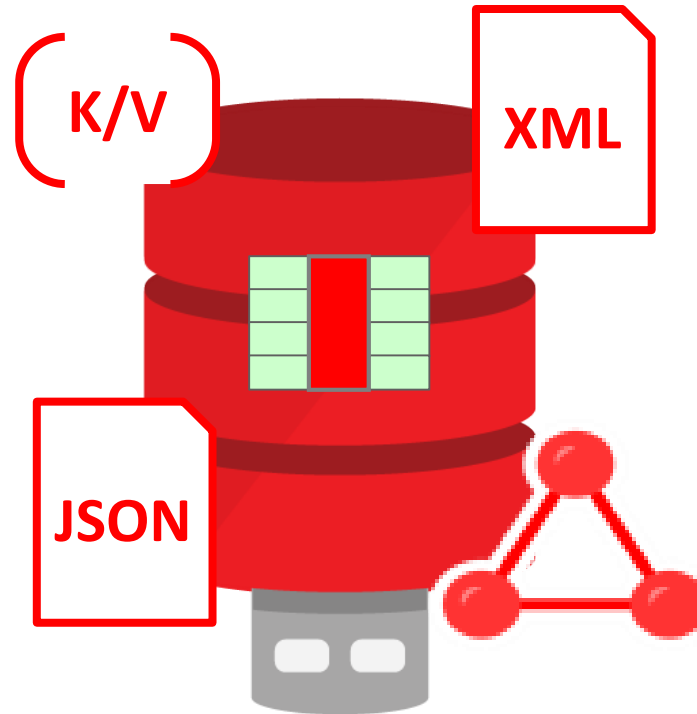
3

Um pouco de como o Oracle Database  
pode endereça a Persistência Poliglota

# Oracle RDBMS



# Oracle ~~R~~DBMS

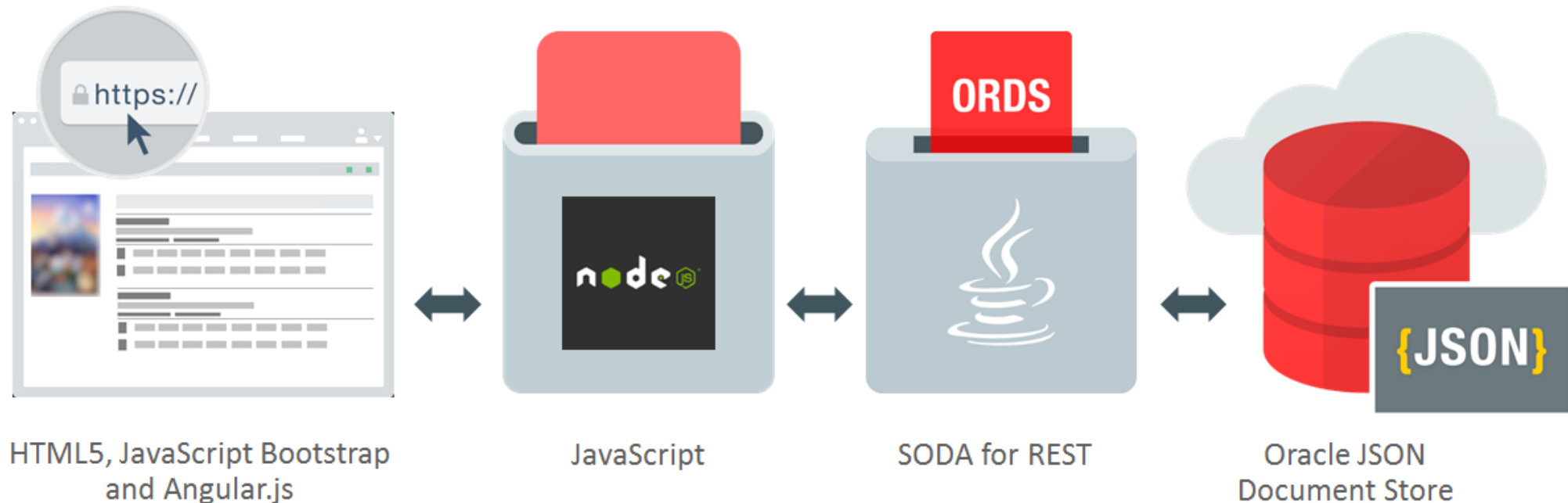






## SODA Simple Oracle Document Access

- Cria e remove collections / índices
- Permite NoSQL através de APIs que permitem executar operações **CRUD**
- Aplica filtros, também chamados **query-by-example (QBE)**
- Expõe **collections** através de chamadas **REST**
- Realiza bulk Inserts



# Acesso NoSQL: Alguns Parâmetros usando QBE no SODA

- Ordenando o retorno

```
{"$query": {}, "$orderby": {"releaseDate": -1}}
```

- Valor exato

```
{"location.city": "SAN FRANCISCO"}
```

- Lista de valores

```
{"id": {"$in": [245168, 299687, 177572, 76757]}}
```

- Full Text Searching

```
{"plot": {"$contains": "$ (colour) "}}
```

- Múltiplos predicados

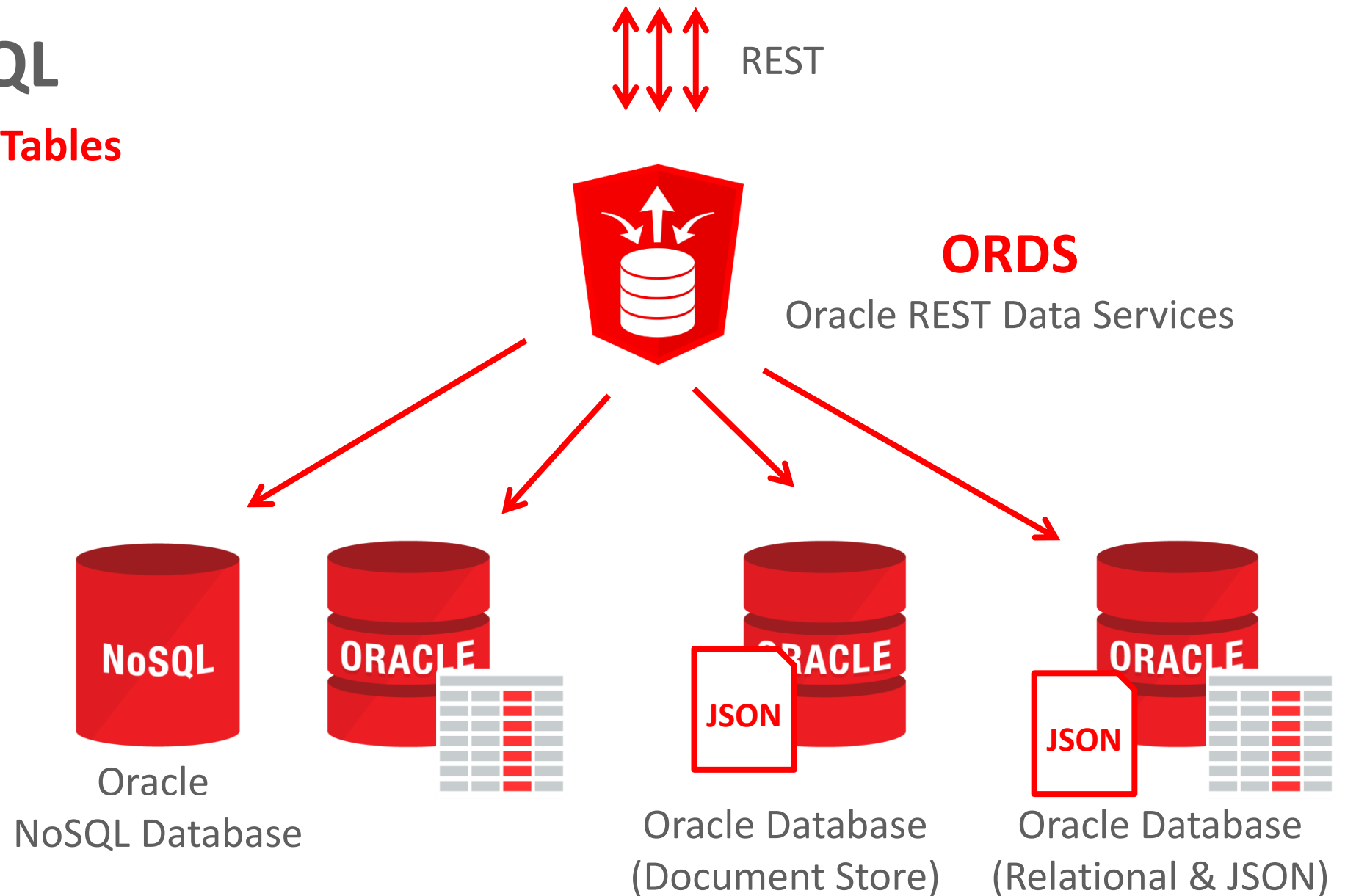
```
{"movieId": 109410,  
  "startTime": {  
    "$gte": "2016-09-12T07:00:00.000Z",  
    "$lt": "2016-09-13T07:00:00.000Z"  
  },  
  "$orderby": {"screenId": 1, "startTime": 2}  
}
```

- Pesquisa por distância

```
{"location.geoCoding": {  
  "$near": {  
    "$geometry": {  
      "type": "Point",  
      "coordinates": [37.8953, -122.1247]  
    },  
    "$distance": 5,  
    "$unit": "mile"  
  }  
}}
```

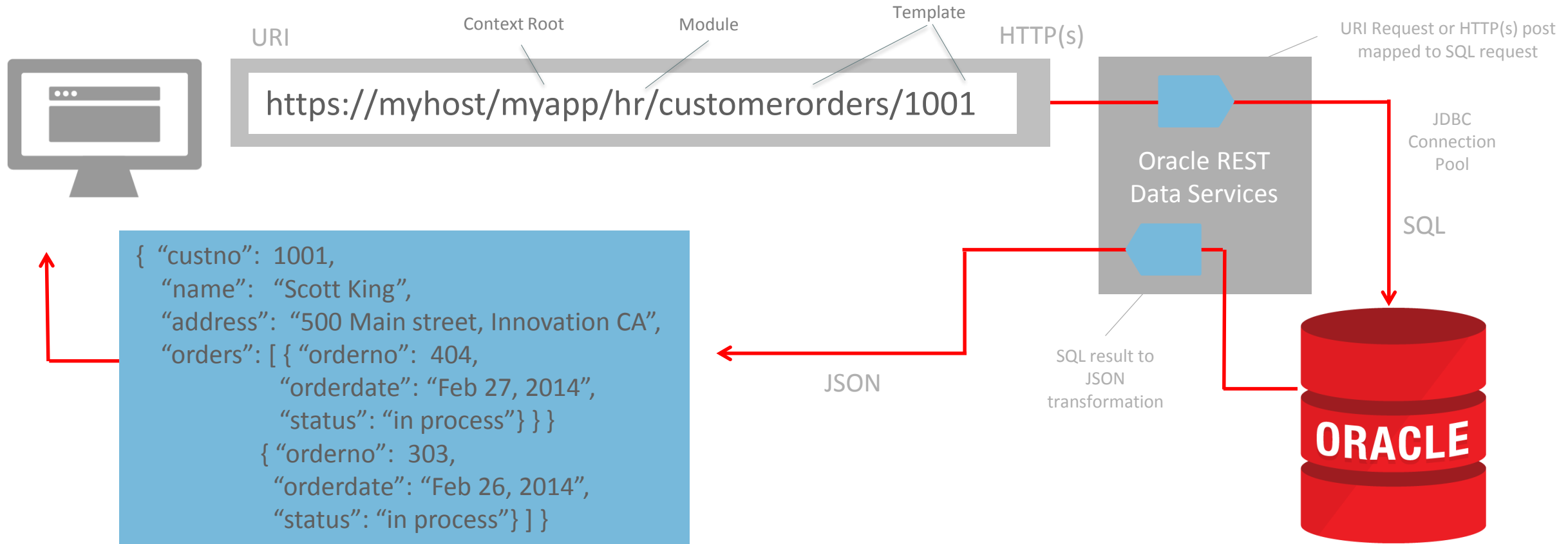
# Acesso NoSQL

## RESTful Database Tables



# Oracle REST Data Services

## HTTP(s) API App-Dev com Tabelas no 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.



# SODA

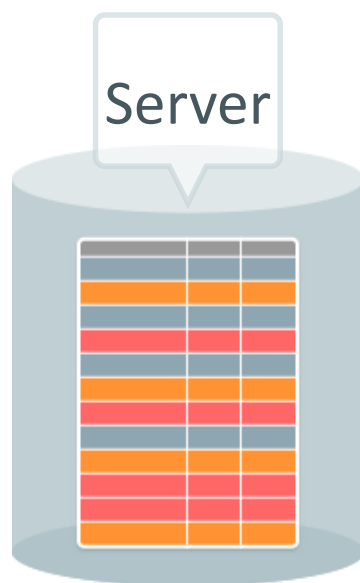
## Simple Oracle Document Access

		Native (SQL) API
Java		✓
.NET		✓
Node.js		✓
REST (ORDS)		✓
Ruby		✓
Python		✓
PHP		✓
R		✓
Perl		✓

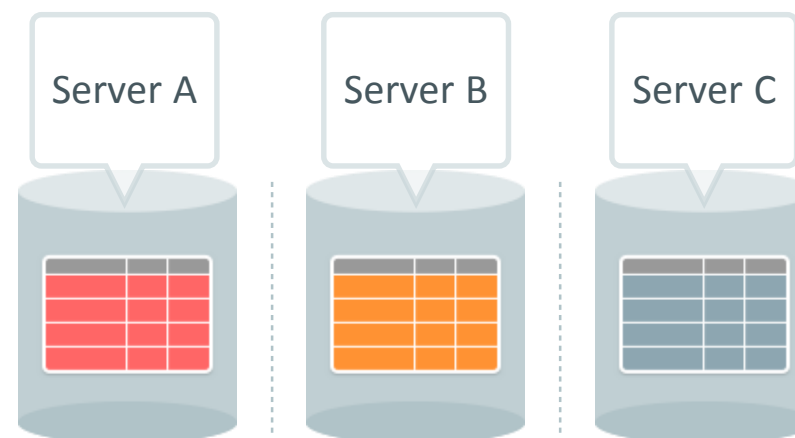
# Outros Recursos para suporte à persistência poliglota

# Oracle Database Sharding

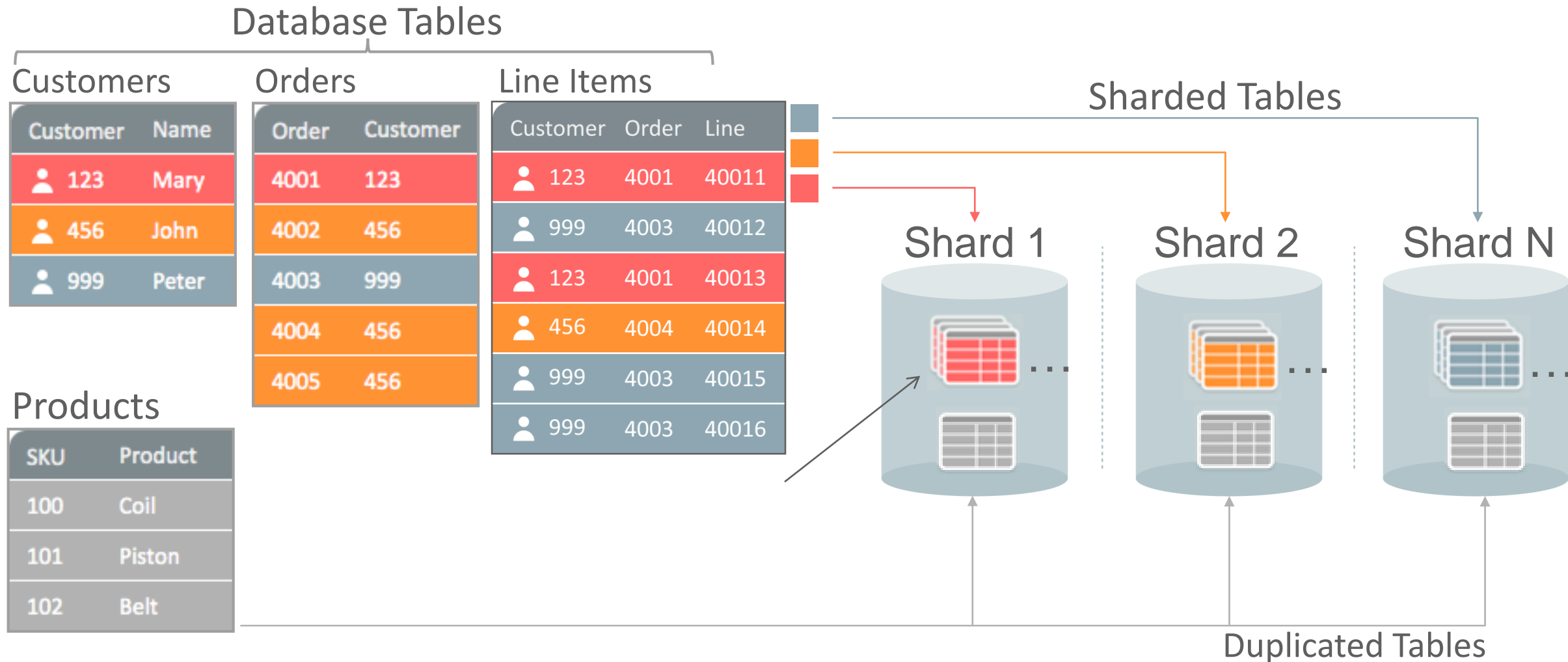
Banco de Dados Oracle  
sem Sharding



Um Banco de Dados Oracle  
com Sharding



# Criação de Schema – Tabelas Sharded e Duplicated

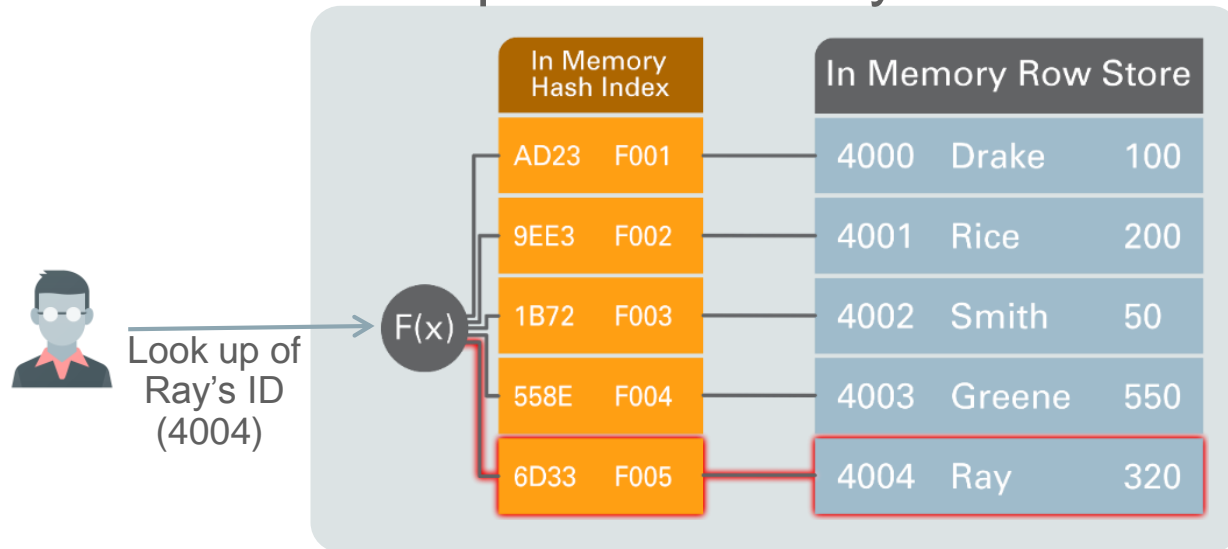




# Acesso Chave/Valor otimizado para Leitura

- Novo Hash Index para leitura otimizada
- Sem bloqueios, 100% em memória
- Desvia da camada SQL e executa diretamente na camada de dados

## Example : Read Rays Balance



```
CREATE TABLE test_lookup  
  ( key_col NUMBER(5) PRIMARY KEY,  
    value_col VARCHAR2(15))  
MEMOPTIMIZE FOR READ;
```

# “Desliga” a Durabilidade (D do ACID)

```
ALTER [SYSTEM | SESSION]  
SET COMMIT_WRITE = 'BATCH,NOWAIT';
```

Ou

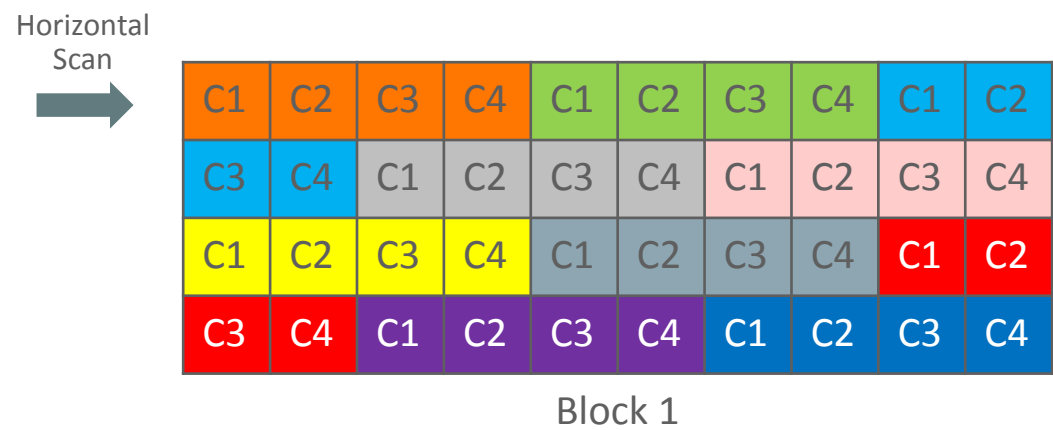
```
INSERT INTO TAB VALUES ...  
COMMIT BATCH NOWAIT;
```

# Ajusta a Leitura Consistente

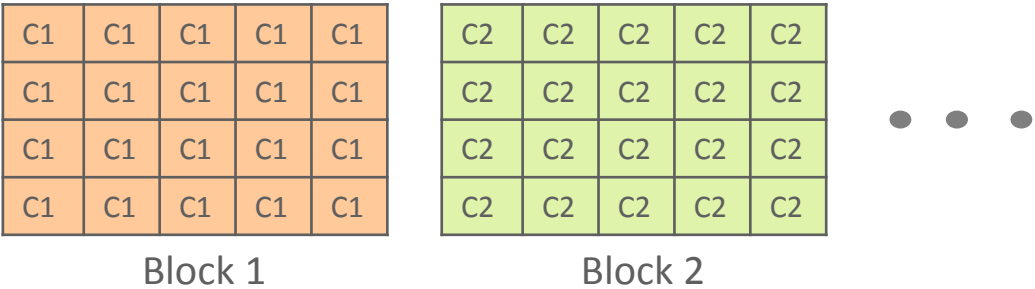
```
SELECT COLUNA  
FROM TABELA  
WHERE COLUNA = VALOR  
AS OF VERSION | TIME
```

# Formatos de Dados Híbridos Linha e Coluna

## Row Format



## InMemory Column Format



**GOOD** SELECT \* FROM TAB WHERE C1 = ? (OLTP QUERY)

**BAD** SELECT \* FROM TAB WHERE C1 = ? (OLTP QUERY)

**BAD** SELECT C2, COUNT(\*) FROM TAB GROUP BY C2 (ANALYTICAL QUERY)

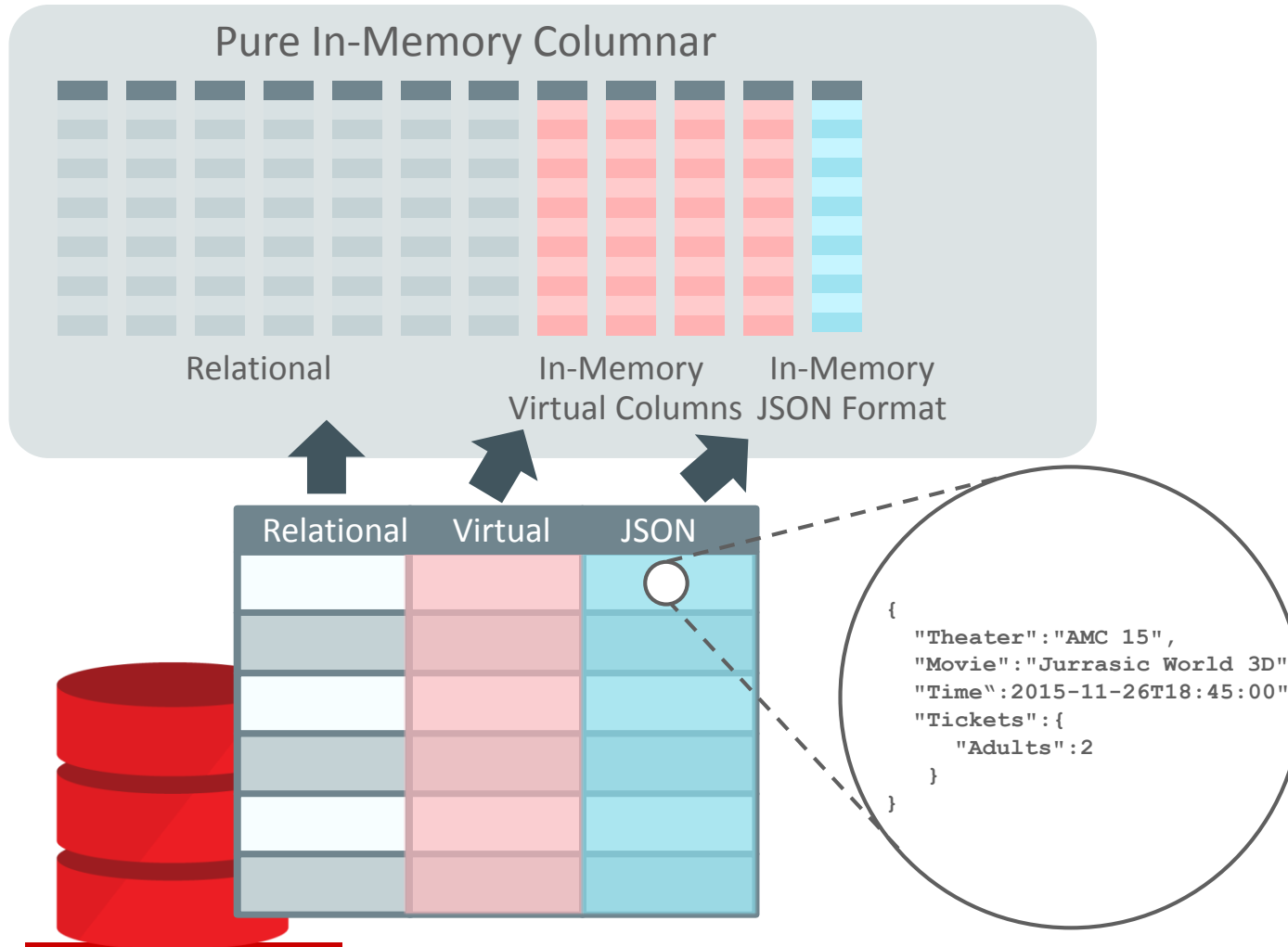
**GOOD** SELECT C2, COUNT(\*) FROM TAB GROUP BY C2 (ANALYTICAL QUERY)

C1 = Primary Key

**ALTER TABLE TAB INMEMORY;**



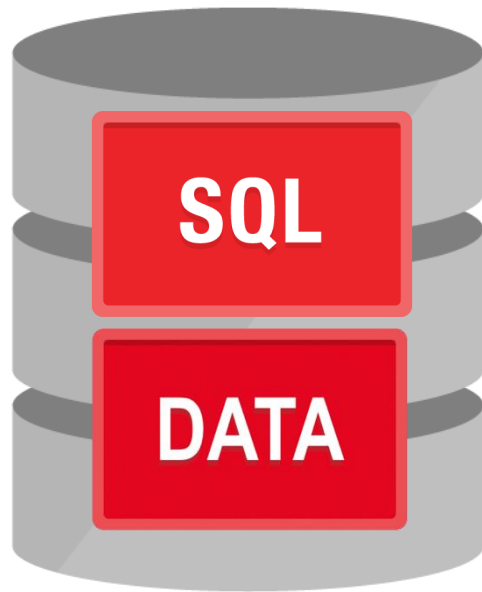
# In-Memory JSON Queries



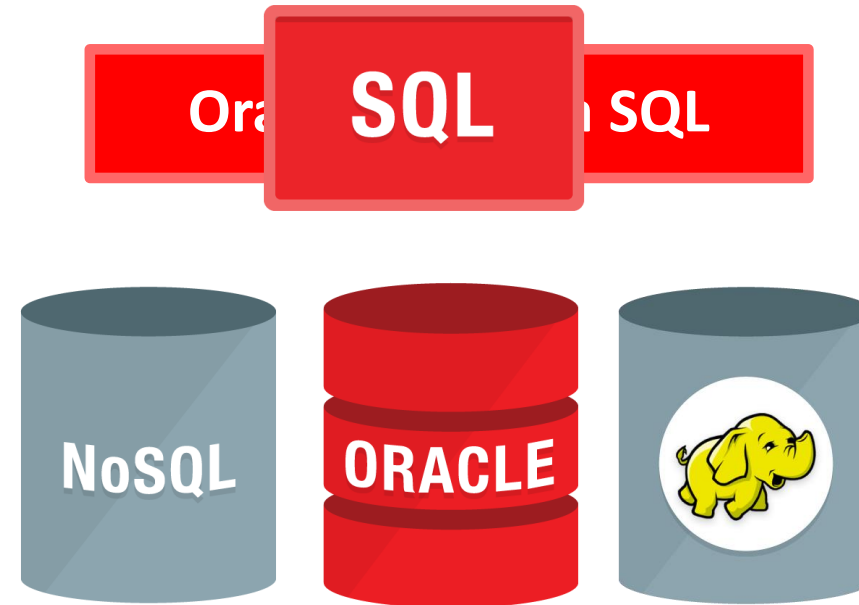
- **CREATE TABLE TAB**  
(.... COL BLOB CHECK IS JSON);
- **ALTER TABLE TAB INMEMORY;**

# Estratégia da Oracle para Big Data

Visão Convencional de  
Data Management



Visão Emergente de  
Data Management





# Integrated Cloud

## Applications & Platform Services



ORACLE®