

# De JSON a Grafos sem perder o SQL: convergindo dados em Oracle Database Palestra

ORACLE

Lucas  
Azevedo

Trilha Data Cloud Modernization



# Lucas Azevedo



<https://www.linkedin.com/in/-lucas-rehem>



# Declaração de Porto Seguro (*Safe Harbor Statement*) - Geral



O texto a seguir tem como objetivo traçar a orientação dos nossos produtos em geral. É destinado somente a fins informativos e não pode ser incorporado a um contrato. Ele não representa um compromisso de entrega de qualquer tipo de material, código ou funcionalidade e não deve ser considerado em decisões de compra. O desenvolvimento, a liberação, a data de disponibilidade e a especificação de quaisquer funcionalidades ou recursos descritos para produtos da Oracle estão sujeitos a mudanças e são de critério exclusivo da Oracle Corporation.

Esta é a tradução de uma apresentação em inglês preparada para a sede da Oracle nos Estados Unidos. A tradução é realizada como cortesia e não está isenta de erros. Os recursos e funcionalidades podem não estar disponíveis em todos os países e idiomas. Caso tenha dúvidas, entre em contato com o representante de vendas da Oracle.



# Qual tipo de banco de dados devo escolher?

---

O dilema ao criar ou modernizar as aplicações da organização.

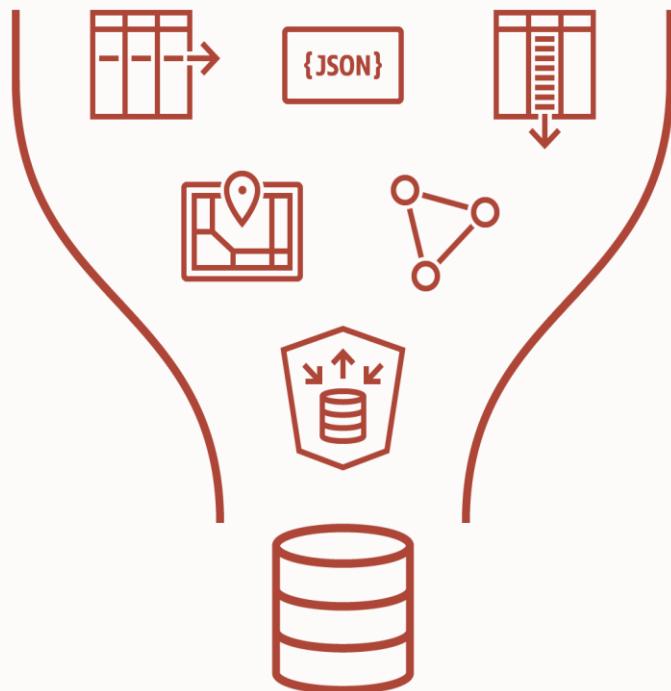
# Abordagem *single-purpose*



- “Preciso de um banco **documental** para a nova aplicação.”
- “Iremos monitorar os objetivos da organização com um **data warehouse**.”
- “Utilizaremos **grafos** para identificar possíveis casos de fraudes no nosso processo.”
- “Necessitamos de uma visão unificada de **segurança** para auxiliar na aderência à legislação de proteção dos dados.”
- “As métricas das atividades serão monitoradas em tempo real pelo banco de dados **in-memory**.”

# Abordagem *multi-purpose*

Um banco de dados convergente



## Multi-model

Suporte nativo para diversos modelos de dados e métodos de acesso.

## Multitenant

Arquitetura de banco de dados em contêiner provê consolidação, isolamento e agilidade.

## Multi-workload

Uma série de otimizações que trabalham isoladamente ou em conjunto para entregar a melhor performance em qualquer tipo de carga de trabalho.

# Multi-model

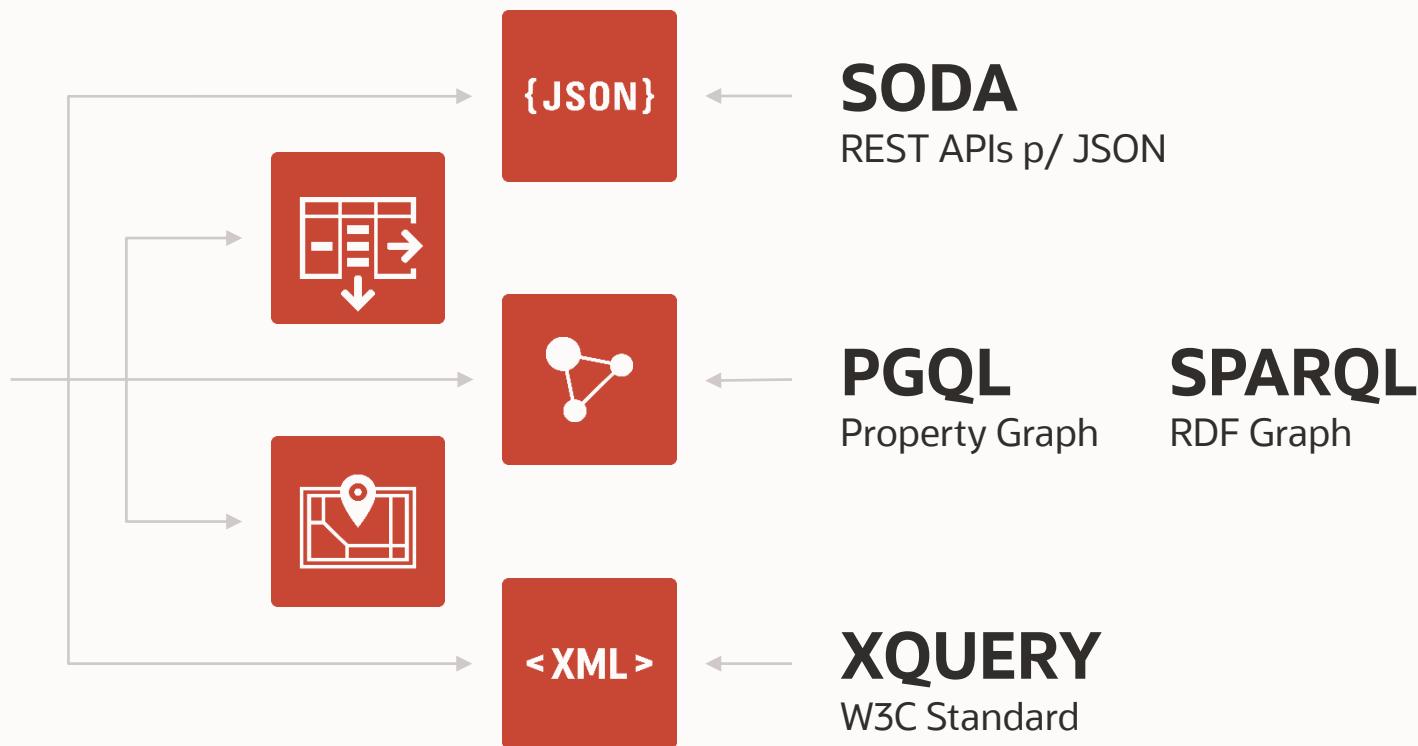
Um motor, multipas personalidades



## Cross-Model Data Access

### SQL & REST

Relacional, JSON,  
Espacial, Grafo, XML



## Model-Specific Data Access

### SODA

REST APIs p/ JSON

### PGQL

Property Graph

### SPARQL

RDF Graph

### XQUERY

W3C Standard

# Multitenant

Consolidação, isolamento e agilidade



## Consolidação

- Bancos de dados plugáveis independentes para cada aplicação ou serviço.
- Operações em comum no nível do container dos bancos de dados (ex. backup, upgrade).



## Isolamento

- *Lockdown Profile*
- Criptografia transparente
- Isolamento de recursos
- PDB único por container, se necessário



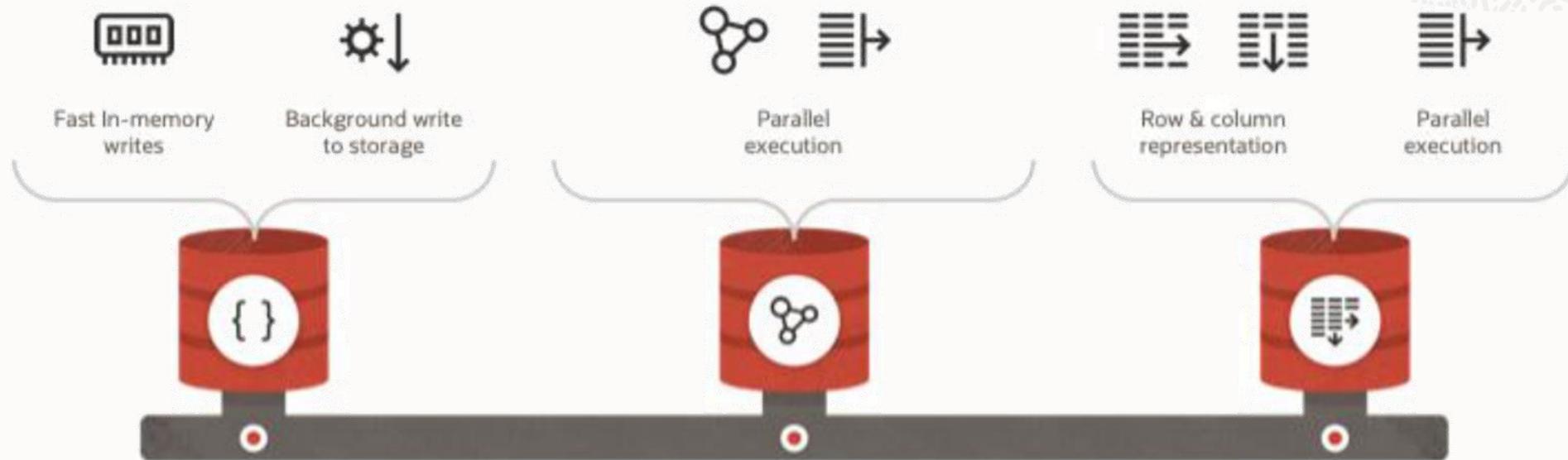
## Agilidade

- Provisionamento rápido
- Realocação on-line entre nuvem pública, nuvem privada ou on-premises



# Multi-workload

Executando diversas cargas de uma vez só



## Internet of Things

Escrita em memória otimizada para dados em *streaming*, como de sensores inteligentes.

## Machine Learning

Execução em paralelo para algoritmos de machine learning ou dados de grafos.

## Real-time Analytics

Representação de linhas e colunas *in-memory*, paralelismo para análises em tempo real em dados transacionais.

# Deep Dive

---

As possibilidades de um banco de dados multi-modelo

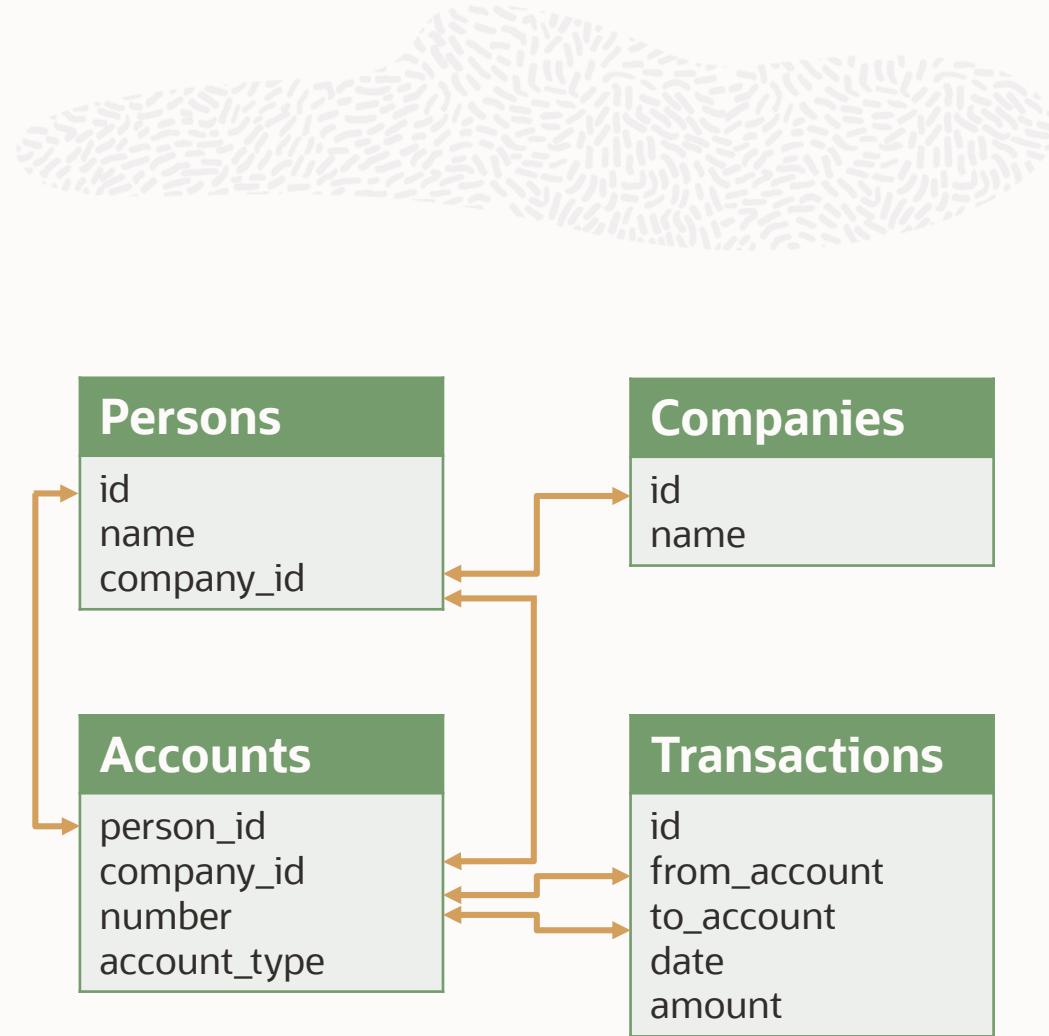
# Transacional

Integridade e consistência nos dados

Modelos **normalizados** (ou que utilizam a terceira forma normal – 3FN) são os mais comuns em sistemas transacionais.

Implementam características como:

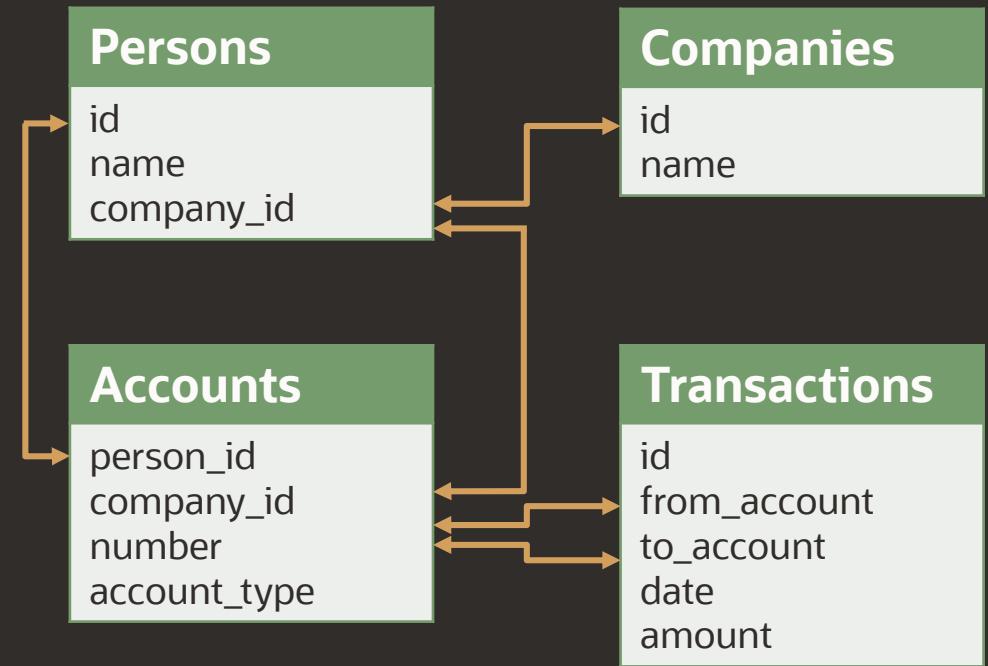
- Transações ACID
- Alta concorrência
- Baixa latência



# Transacional

## CRUD

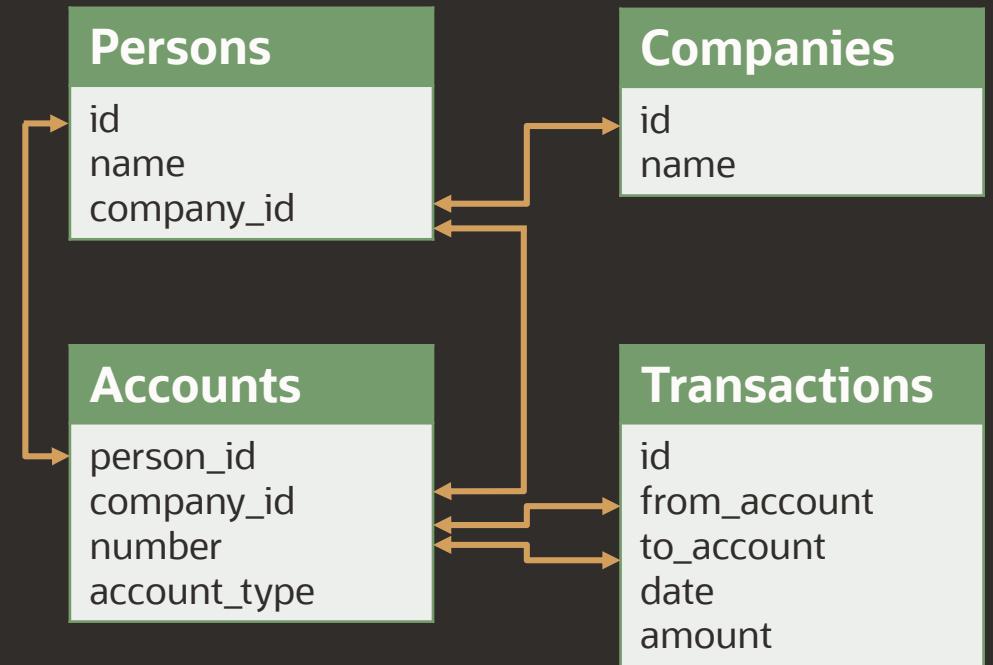
```
/* Create */  
CREATE TABLE persons (  
    id          NUMBER,  
    name        VARCHAR2(50) NOT NULL,  
    company_id NUMBER,  
CONSTRAINT pk_persons PRIMARY KEY ( id ),  
CONSTRAINT fk_company FOREIGN KEY ( company_id )  
    REFERENCES companies ( id )  
);  
  
/* Insert */  
INSERT INTO persons ( id , name )  
    VALUES ( 1 , 'Liam' );
```

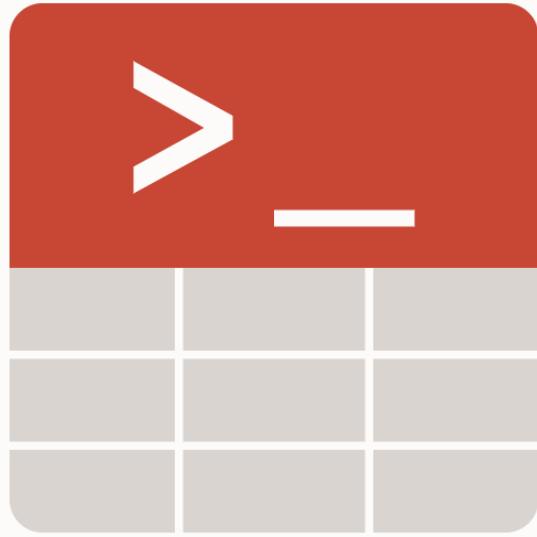


# Transacional

## CRUD

```
/* Update */  
UPDATE persons  
    SET name = 'Liliam'  
WHERE id = 1;  
  
/* Delete */  
DELETE FROM persons WHERE id = 1;  
  
/* Drop */  
DROP TABLE persons CASCADE CONSTRAINTS;
```





# Live SQL

Learn and share SQL

**Tutorial**  
Introduction to SQL

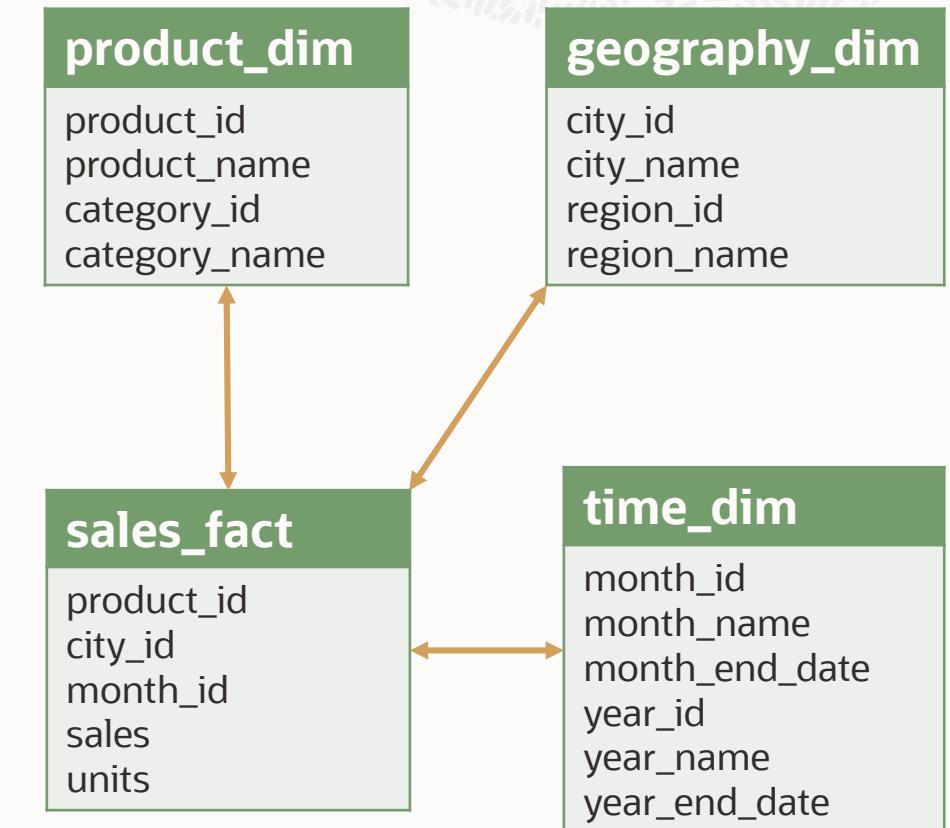
# Data Warehouse

## Análise de dados

Modelos **desnormalizados** são utilizados em sistemas analíticos. Simplificam os relacionamentos e melhoram a performance de consultas em grandes volumes de dados.

Implementam características como:

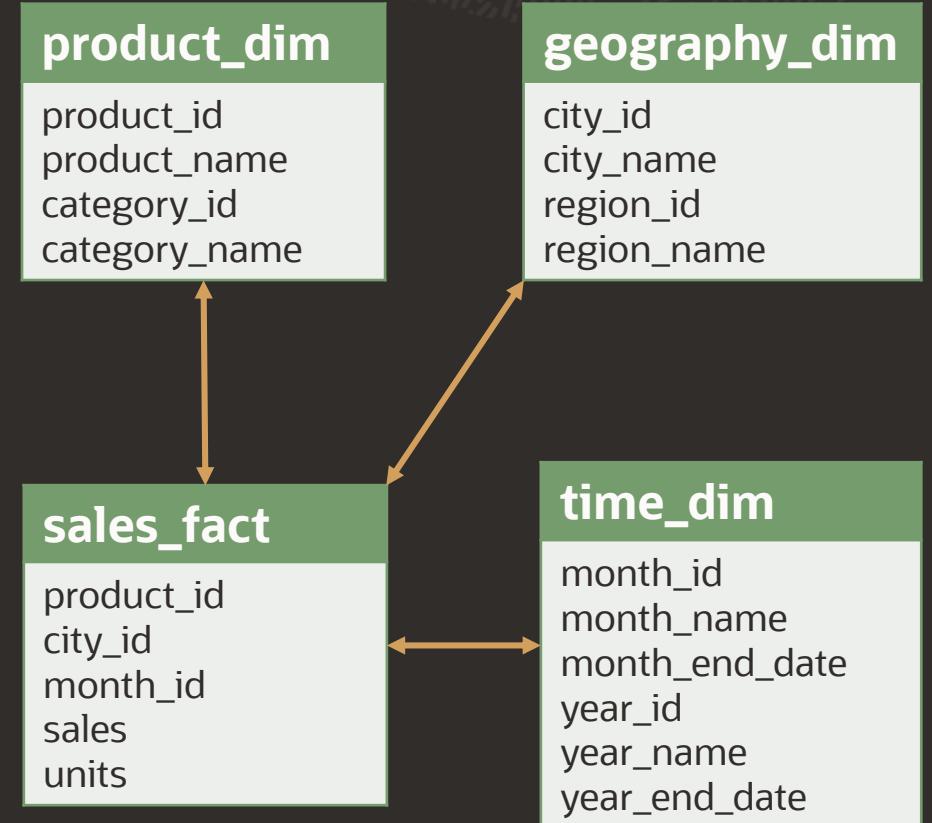
- Arientado ao assunto
- Integrado
- Não volátil
- Variável no tempo



# Data Warehouse

## Visão analítica

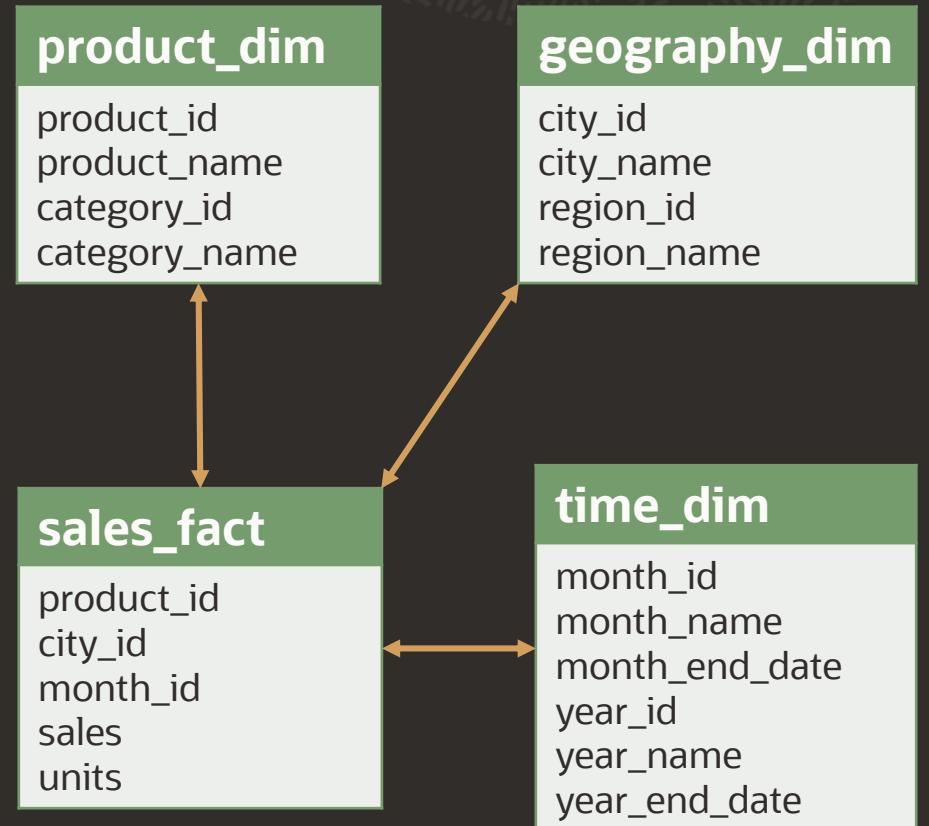
```
/* CREATE DIMENSION */  
CREATE OR REPLACE ATTRIBUTE DIMENSION time_attr_dim  
USING time_dim ATTRIBUTES (  
    year_id  
        CLASSIFICATION caption VALUE 'YEAR_ID'  
        CLASSIFICATION description VALUE 'YEAR ID',  
    year_name  
        CLASSIFICATION caption VALUE 'YEAR_NAME'  
        CLASSIFICATION description VALUE 'Year',  
    year_end_date  
        CLASSIFICATION caption VALUE 'YEAR_END_DATE'  
        CLASSIFICATION description VALUE 'Year End Date',  
    month_id  
        CLASSIFICATION caption VALUE 'MONTH_ID'  
        CLASSIFICATION description VALUE 'MONTH ID',  
    month_name  
        CLASSIFICATION caption VALUE 'MONTH_NAME'  
        CLASSIFICATION description VALUE 'Month',  
    month_end_date  
        CLASSIFICATION caption VALUE 'MONTH_END_DATE'  
        CLASSIFICATION description VALUE 'Month End Date')  
/* CLICK TO CONTINUE... */
```



# Data Warehouse

## Visão analítica

```
/* ... */  
  
LEVEL month  
    CLASSIFICATION caption VALUE 'MONTH'  
    CLASSIFICATION description VALUE 'Month'  
    KEY month_id  
    MEMBER NAME month_name  
    MEMBER CAPTION month_name  
    ORDER BY month_end_date  
    DETERMINES ( month_end_date )  
  
LEVEL year  
    CLASSIFICATION caption VALUE 'YEAR'  
    CLASSIFICATION description VALUE 'Year'  
    KEY year_id  
    MEMBER NAME year_name  
    MEMBER CAPTION year_name  
    MEMBER DESCRIPTION year_name  
    ORDER BY year_end_date  
    DETERMINES ( year_end_date )  
  
ALL MEMBER NAME 'ALL TIMES';
```

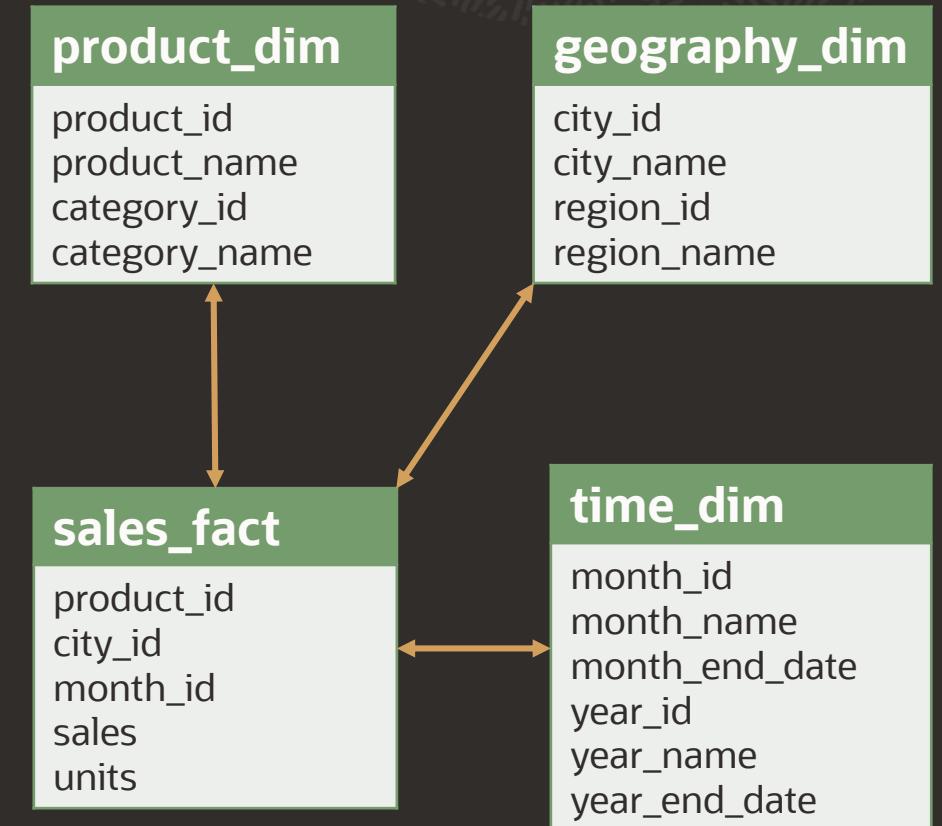


# Data Warehouse

## Visão analítica

```
/* DEFINE HIERARCHY */

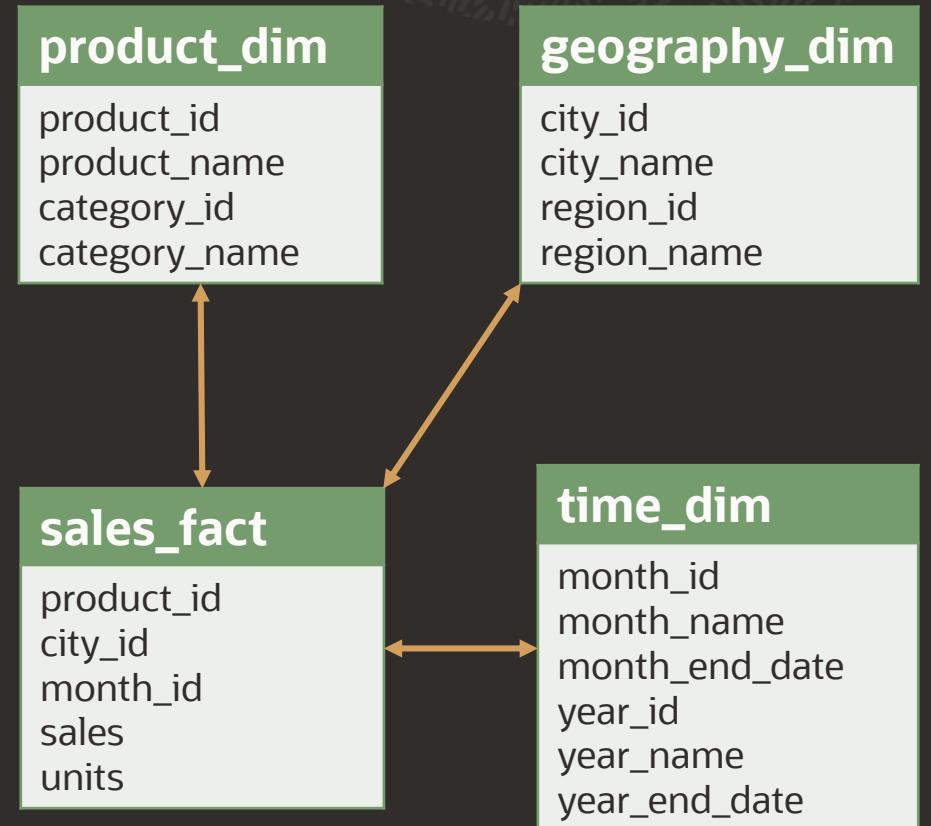
CREATE OR REPLACE HIERARCHY time_hier
  CLASSIFICATION caption VALUE 'CALENDAR'
  CLASSIFICATION description VALUE 'CALENDAR'
USING time_attr_dim
  (month CHILD OF
  year);
```



# Data Warehouse

## Visão analítica

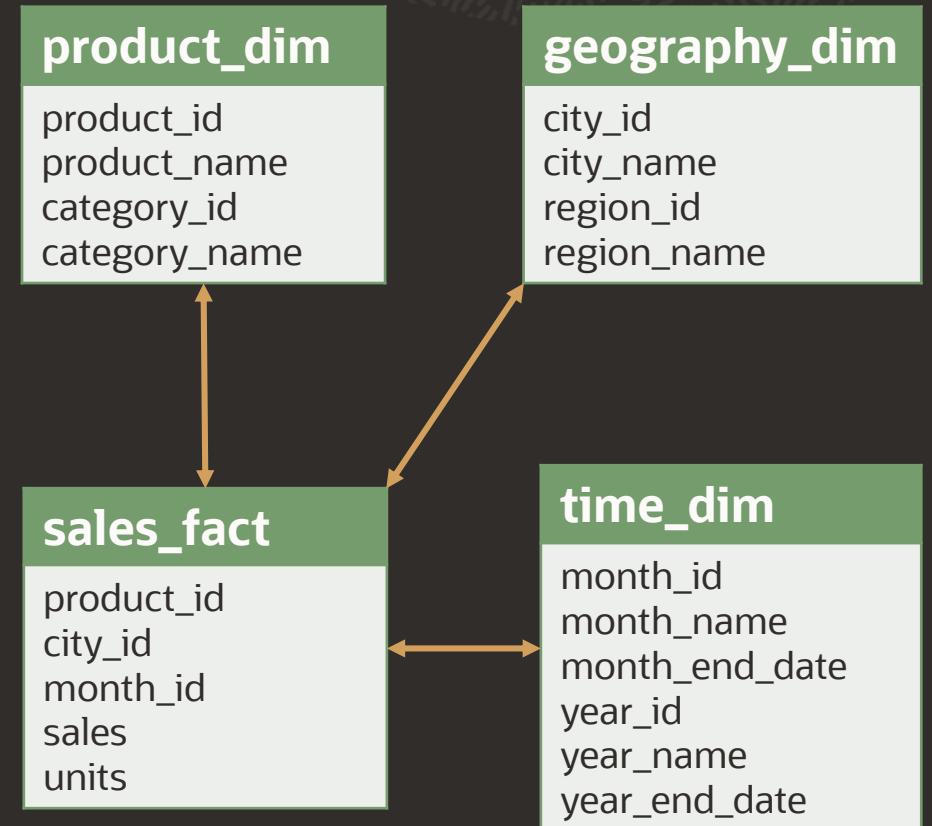
```
/* CREATE ANALYTIC VIEW */
CREATE OR REPLACE ANALYTIC VIEW sales_av
USING sales_fact
DIMENSION BY (
    time_attr_dim
        KEY month_id REFERENCES month_id
    HIERARCHIES (
        time_hier DEFAULT),
    product_attr_dim
        KEY category_id REFERENCES category_id
    HIERARCHIES (
        product_hier DEFAULT),
    geography_attr_dim
        KEY city_id REFERENCES city_id
    HIERARCHIES (
        geography_hier DEFAULT)
)
/* CLICK TO CONTINUE... */
```



# Data Warehouse

## Visão analítica

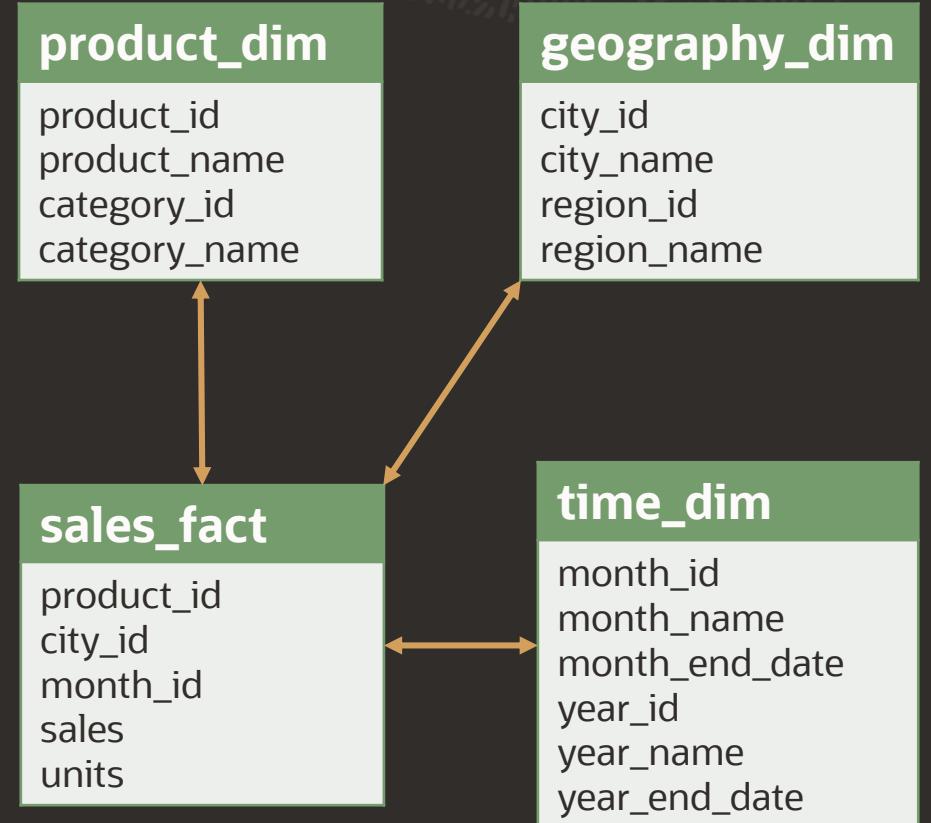
```
/*           ...          */
/* DEFINE MEASURES */
MEASURES
(sales FACT sales,
 units FACT units,
 sales_prior_period AS
    (LAG(sales) OVER (HIERARCHY time_hier OFFSET 1)),
sales_share_prod_parent AS
    (SHARE_OF(sales HIERARCHY product_hier PARENT)),
sales_share_geog_parent AS
    (SHARE_OF(sales HIERARCHY geography_hier PARENT))
)
DEFAULT MEASURE sales
/* CLICK TO CONTINUE... */
```



# Data Warehouse

## Visão analítica

```
/*      ...      */  
/* DEFINE AGGREGATIONS */  
CACHE  
MEASURE GROUP (  
    sales,  
    units  
)  
LEVELS (  
    time_hier.year,  
    product_hier.category,  
    geography_hier.region  
)  
MATERIALIZED;
```



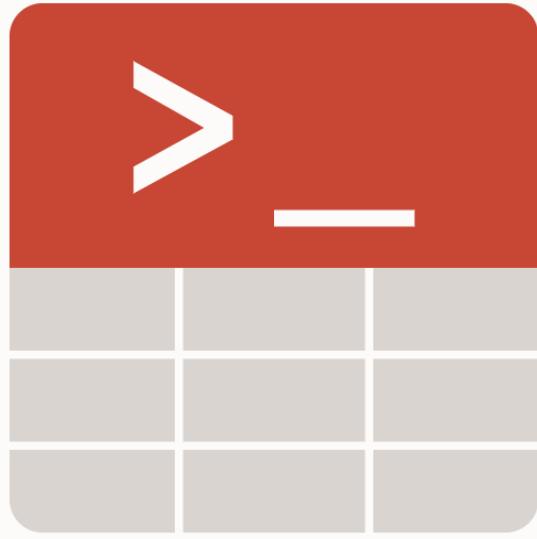
# Data Warehouse

## Visão analítica

```
/* ANALYTIC QUERY */  
SELECT  
    b.*,  
    COUNT( * ) OVER (  
        PARTITION BY colour  
        ORDER BY brick_id  
    ) running_total,  
    SUM( weight ) OVER (  
        PARTITION BY colour  
        ORDER BY brick_id  
    ) running_weight  
FROM bricks b;
```



```
/* AGGREGATE FUNCTIONS */  
COUNT()  
SUM()  
AVG()  
MEDIAN()  
STDDEV()  
MIN()  
MAX()  
CORR()  
FIRST()  
LAST()  
LAG()  
LEAD()  
RANK()  
OVER PARTITION BY  
/* ... */
```



# Live SQL

Learn and share SQL

## Tutoriais

[Analytic View Materialized Aggregate Cache](#)

[Analytic Functions: Database for Developers](#)



# Documento JSON

## Flexibilidade e agilidade

Uma das principais categorias de banco de dados NoSQL. Ao implementar conceitos de armazenamento de dados em **chave-valor**, torna os **esquemas dinâmicos** e facilita alterações em sua estrutura.



```
{ transaction: 'aKl91qh0dFpqi',
  from: {
    id: 10039,
    name: 'Camille'
  },
  to: {
    id: 8021,
    name: 'Nikita',
    company: 'Pizza Co.'
  },
  date: '2021-11-18',
  value: 69.90
}
```

# Documento

## Flexibilidade e agilidade

```
/* CREATE DOCUMENT */  
CREATE TABLE invoice (  
    id    AS NUMBER,  
    doc   AS JSON  
);  
  
{ transaction: 'aKl91qh0dFpqi',  
  from: {  
    id: 10039,  
    name: 'Camille'  
  },  
  to: {  
    id: 8021,  
    name: 'Nikita',  
    company: 'Pizza Co.'  
  },  
  date: '2030-11-17',  
  value: 69.90  
}
```

# Documento

## Flexibilidade e agilidade

```
/* CREATE DOCUMENT */  
CREATE TABLE invoice (  
    id    AS NUMBER,  
    doc    AS JSON  
);  
  
/* INSERT DOCUMENT */  
INSERT INTO invoice VALUES (  
    1001,  
    '  
);
```



```
{ transaction: 'aKl91qh0dFpqi',  
  from: {  
    id: 10039,  
    name: 'Camille'  
  },  
  to: {  
    id: 8021,  
    name: 'Nikita',  
    company: 'Pizza Co.'  
  },  
  date: '2030-11-17',  
  value: 69.90  
}
```

# Documento

Flexibilidade e agilidade

```
/* QUERY DOCUMENT */  
SELECT  
    i.doc.to.company AS COMPANY,  
    ,i.doc.value      AS VALUE  
FROM invoice i;
```

The diagram illustrates the mapping of a database query result to a JSON document. On the left, a SQL query is shown:

```
/* QUERY DOCUMENT */  
SELECT  
    i.doc.to.company AS COMPANY,  
    ,i.doc.value      AS VALUE  
FROM invoice i;
```

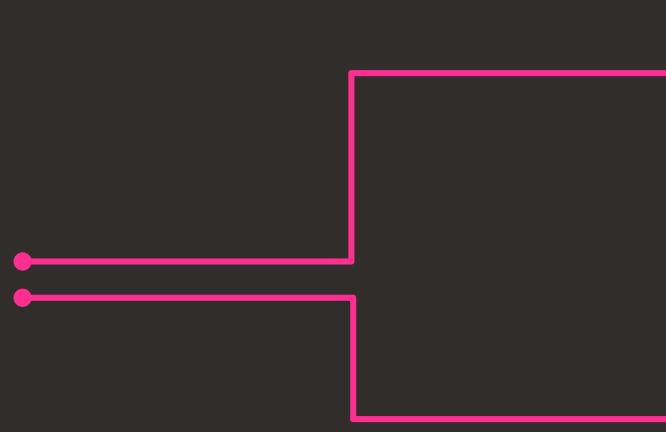
Two columns from this query, `COMPANY` and `VALUE`, are highlighted with pink dots and connected by a pink bracket to the corresponding fields in a JSON object on the right. The JSON object is:

```
{ transaction: 'aKl91qh0dFpqi',  
  from: {  
    id: 10039,  
    name: 'Camille'  
  },  
  to: {  
    id: 8021,  
    name: 'Nikita',  
    company: 'Pizza Co.'  
  },  
  date: '2030-11-17',  
  value: 69.90  
}
```

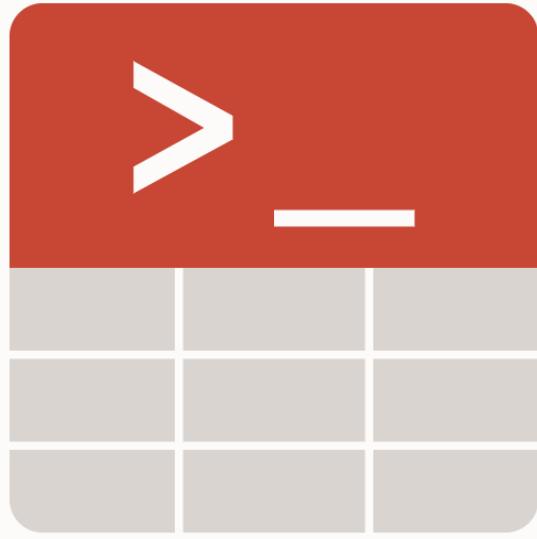
# Documento

## Flexibilidade e agilidade

```
/* QUERY DOCUMENT */  
SELECT  
    i.doc.to.company AS COMPANY  
    ,i.doc.value      AS VALUE  
FROM invoice i;  
  
/* UPDATE DOCUMENT */  
UPDATE invoice  
SET doc = JSON_TRANSFORM (  
    doc,  
    REMOVE '$.to.name',  
    SET '$.value' = 59.90  
)  
FROM invoice i  
WHERE i.id = 1001;
```



```
{ transaction: 'aKl91qh0dFpqi',  
  from: {  
    id: 10039,  
    name: 'Camille'  
  },  
  to: {  
    id: 8021,  
    name: 'Nikita',  
    company: 'Pizza Co.'  
  },  
  date: '2030-11-17',  
  value: 59.90  
}
```



# Live SQL

Learn and share SQL

## Tutorial

[SQL/JSON Features in Database](#)

# Documento XML

## Interoperabilidade e semântica

O XML é uma recomendação da W3C para gerar **linguagens de marcação** para necessidades especiais. A simplicidade e flexibilidade deste tipo de documento, que se utiliza de **tags e atributos**, permite que diferentes sistemas possam interoperar por meio de um **formato universal**.



```
<transaction id="aKl91qh0dFpqi">
  <from>
    <id>10039</id>
    <name>Camille</name>
  </from>
  <to>
    <id>8021</id>
    <name>Nikita</name>
    <company>Pizza Co.</company>
  </to>
  <date>2021-11-18</date>
  <value>69.90</value>
</transaction>
```

# Documento

## Interoperabilidade e semântica

```
/* CREATE DOCUMENT */  
CREATE TABLE invoice (  
    id    AS NUMBER,  
    xml   AS XMLType  
);
```

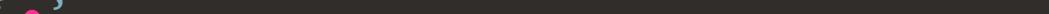


```
<transaction id="aKl91qh0dFpqi">  
    <from>  
        <id>10039</id>  
        <name>Camille</name>  
    </from>  
    <to>  
        <id>8021</id>  
        <name>Nikita</name>  
        <company>Pizza Co.</company>  
    </to>  
    <date>2021-11-18</date>  
    <value>69.90</value>  
</transaction>
```

# Documento

## Interoperabilidade e semântica

```
/* CREATE DOCUMENT */  
CREATE TABLE invoice (  
    id    AS NUMBER,  
    xml   AS XMLType  
);  
  
/* INSERT DOCUMENT */  
INSERT INTO invoice VALUES (  
    1001,  
    '  
);
```



```
<transaction id="aK191qh0dFpqi">  
  <from>  
    <id>10039</id>  
    <name>Camille</name>  
  </from>  
  <to>  
    <id>8021</id>  
    <name>Nikita</name>  
    <company>Pizza Co.</company>  
  </to>  
  <date>2021-11-18</date>  
  <value>69.90</value>  
</transaction>
```

# Documento

## Interoperabilidade e semântica

```
/* QUERY DOCUMENT OPTION 1*/
```

```
SELECT  
    XMLQUERY('/transaction/to/company'  
        PASSING i.xml  
        RETURNING CONTENT).GETCLOBVAL() AS COMPANY
```

```
,XMLQUERY('/transaction/value'  
        PASSING i.xml  
        RETURNING CONTENT).GETCLOBVAL() AS VALUE
```

```
FROM invoice i;
```

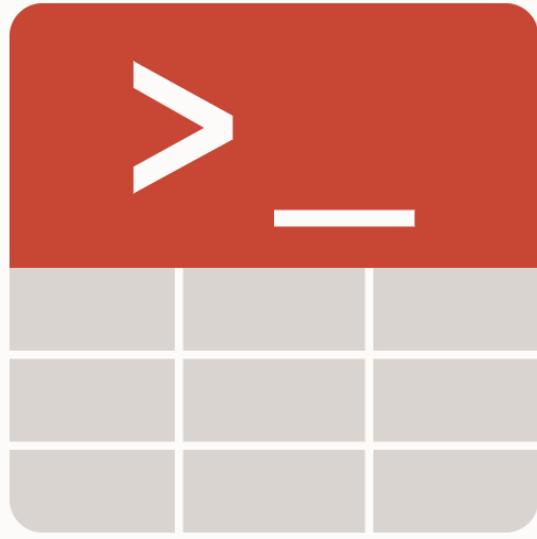
```
<transaction id="aK191qh0dFpqi">  
  <from>  
    <id>10039</id>  
    <name>Camille</name>  
  </from>  
  <to>  
    <id>8021</id>  
    <name>Nikita</name>  
    <company>Pizza Co.</company>  
  </to>  
  <date>2021-11-18</date>  
  <value>69.90</value>  
</transaction>
```

# Documento

## Interoperabilidade e semântica

```
/* QUERY DOCUMENT OPTION 2*/
SELECT
    inf.*
FROM invoice i
,XMLTABLE('/transaction' PASSING i.xml
          COLUMNS
            company VARCHAR2(128) PATH 'to/company'
            value    NUMBER        PATH 'value'
          ) inf
;
```

```
<transaction id="aK191qh0dFpqi">
  <from>
    <id>10039</id>
    <name>Camille</name>
  </from>
  <to>
    <id>8021</id>
    <name>Nikita</name>
    <company>Pizza Co.</company>
  </to>
  <date>2021-11-18</date>
  <value>69.90</value>
</transaction>
```



# Live SQL

Learn and share SQL

## Tutorial

Storing and Processing XML Documents

# Espacial

## Analisando dados geoespaciais

Armazenamento, processamento e análise de **dados geoespaciais**.

- 2D Spatial Data
- 3D Point Cloud e LiDAR
- Raster Data
- Topology Data
- Streaming Point Data



# Espacial

## Analisando dados geoespaciais



```
/* GEOMETRY DATATYPE */
CREATE TABLE warehouses
(
    warehouse_id      NUMBER(3,0)
    ,warehouse_name   VARCHAR2(35 CHAR)
    ,location_id      NUMBER(4,0)
    ,wh_geo_location  SDO_Geometry
);
```



# Espacial

## Analisando dados geoespaciais

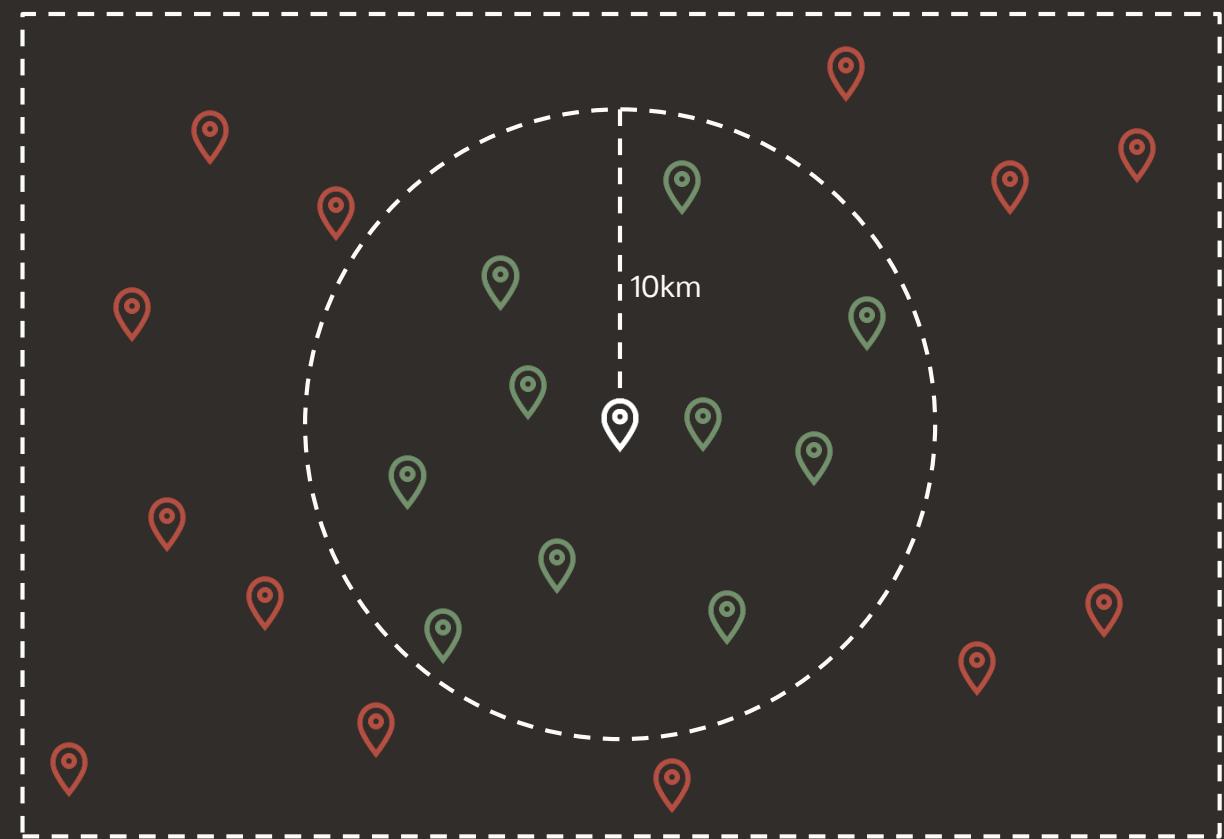
```
/*INSERT GEOMETRY DATA*/
INSERT INTO warehouses VALUES (
  3
,'São Paulo, BR'
,1400
,SDO_Geometry(
  2001           -- Single point 2D
,4326            -- Coordinate System WGS 84
,MDSYS.SDO_POINT_TYPE(
  -23.5489      -- Longitude
,-46.6388       -- Latitude
, NULL)
, NULL
, NULL
);
)
```



# Espacial

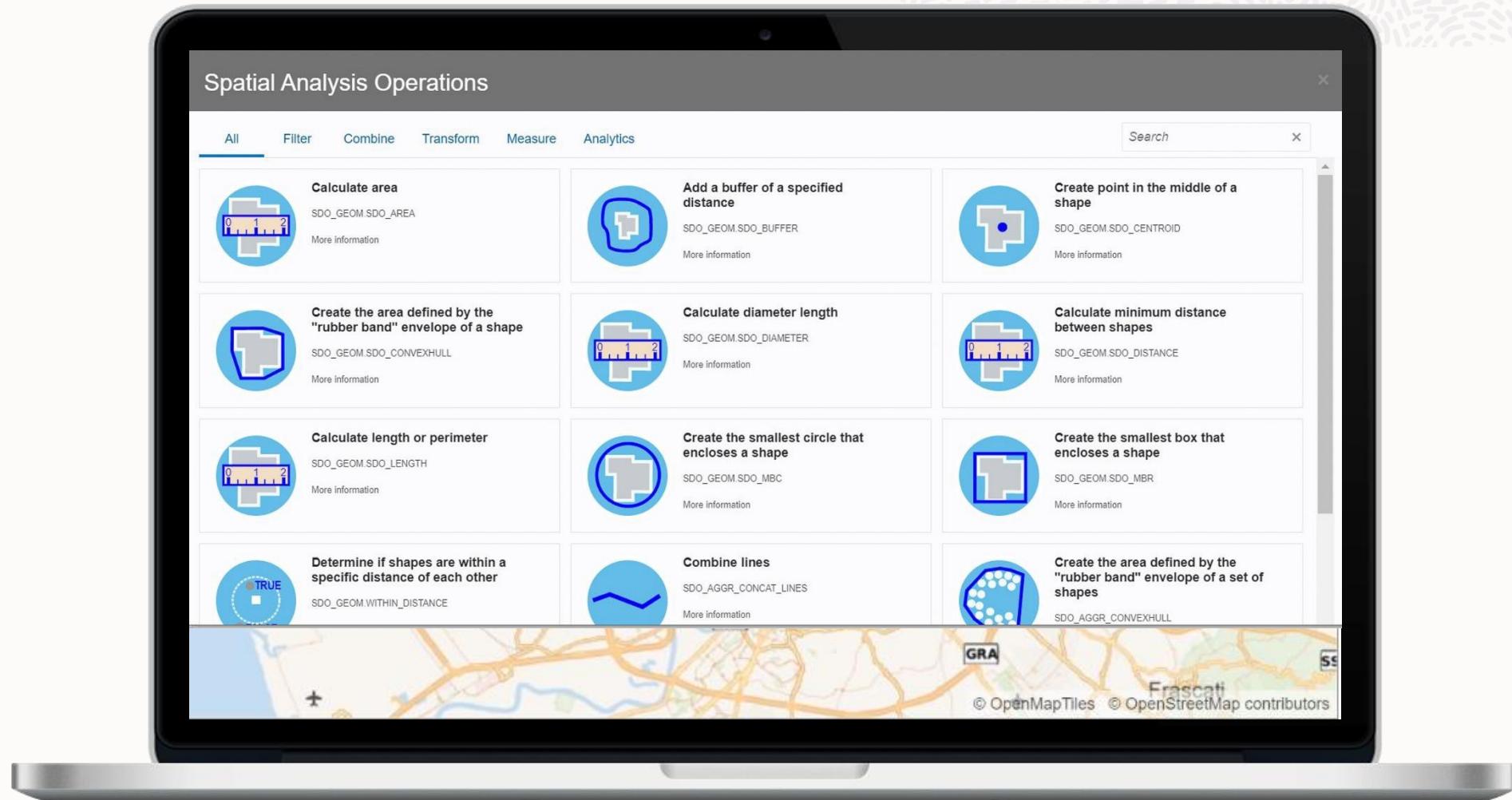
## Analisando dados geoespaciais

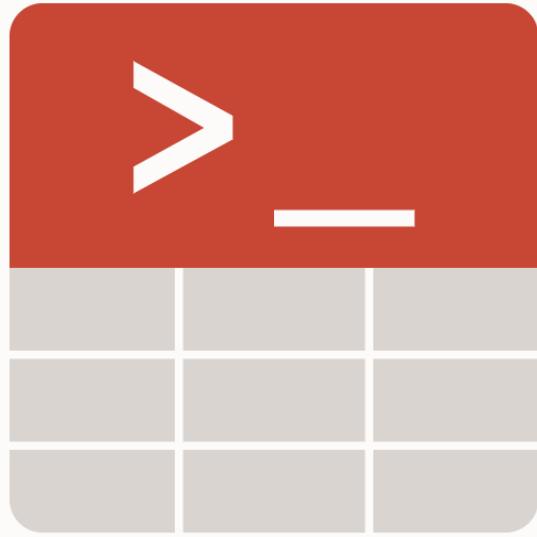
```
/* LOCATION-BASED QUERY */
SELECT
    c.customer_id
    ,c.cust_last_name
    ,c.gender
FROM warehouses w,
     customers c
WHERE
    w.warehouse_id = 3
AND SDO_WITHIN_DISTANCE(
    c.cust_geo_location
    ,w.wh_geo_location
    ,'DISTANCE=10 UNIT=KM') = 'TRUE'
;
```



# Espacial

## Spatial Studio





# Live SQL

Learn and share SQL

## Tutorial

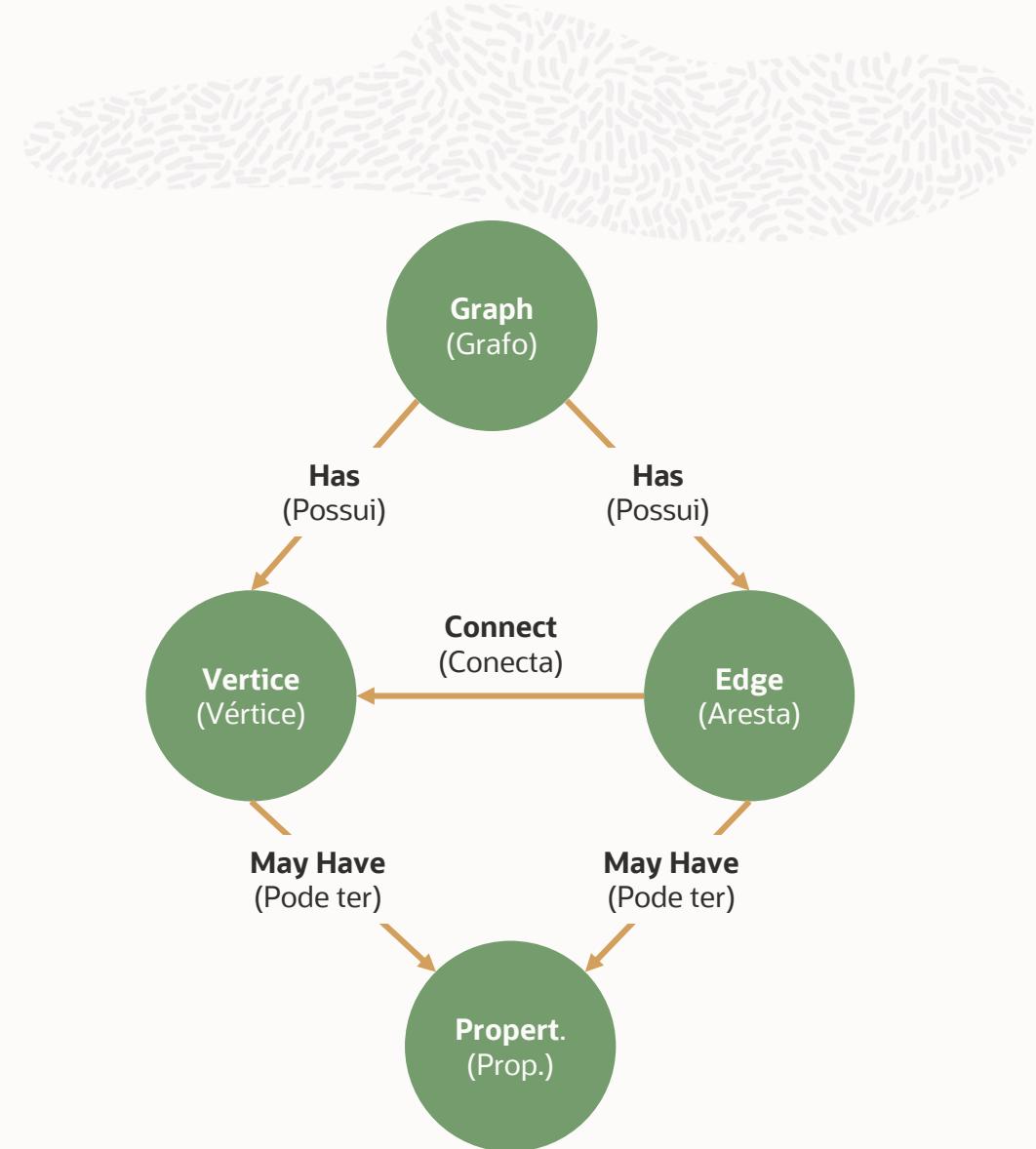
Introduction to location-based analysis using spatial features

# Grafo

## Análise de relacionamentos

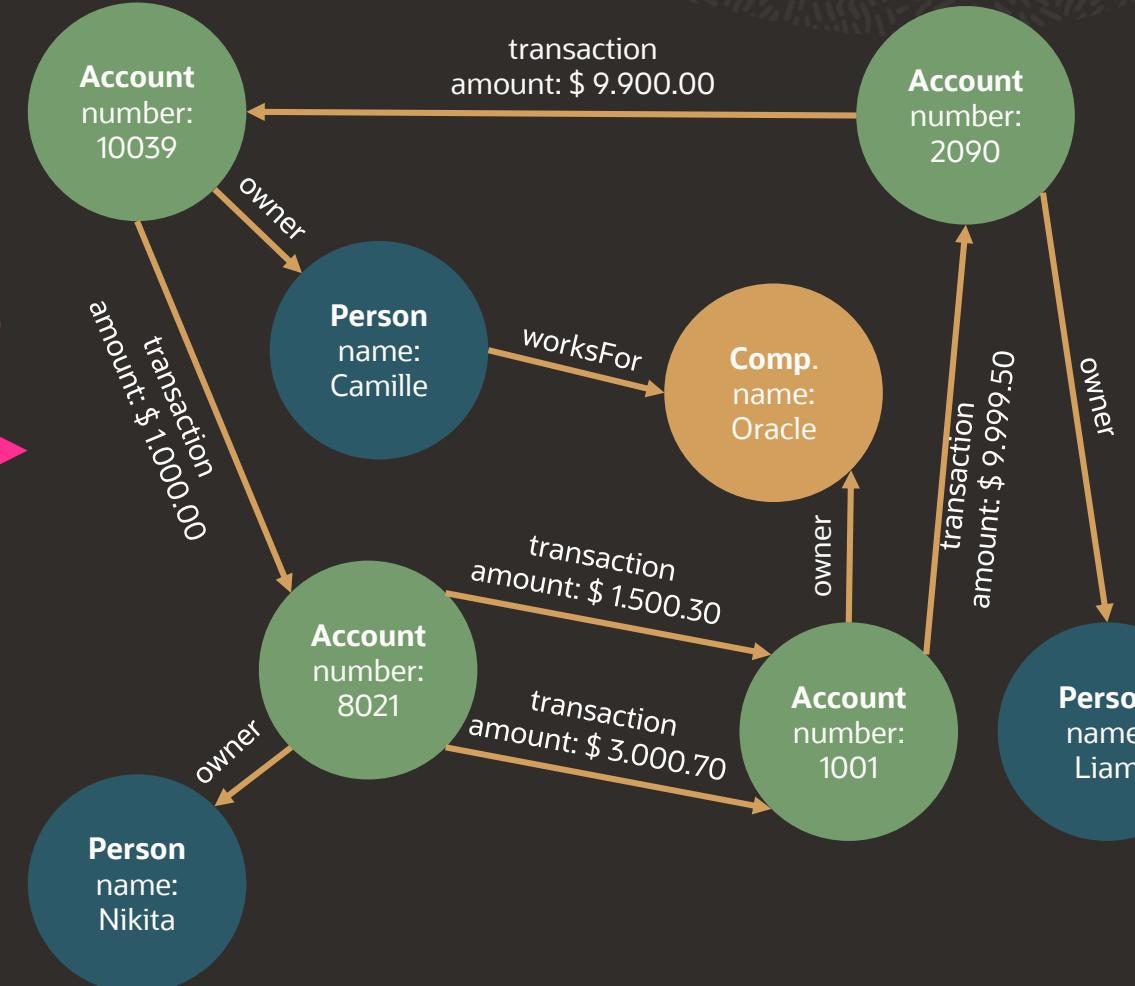
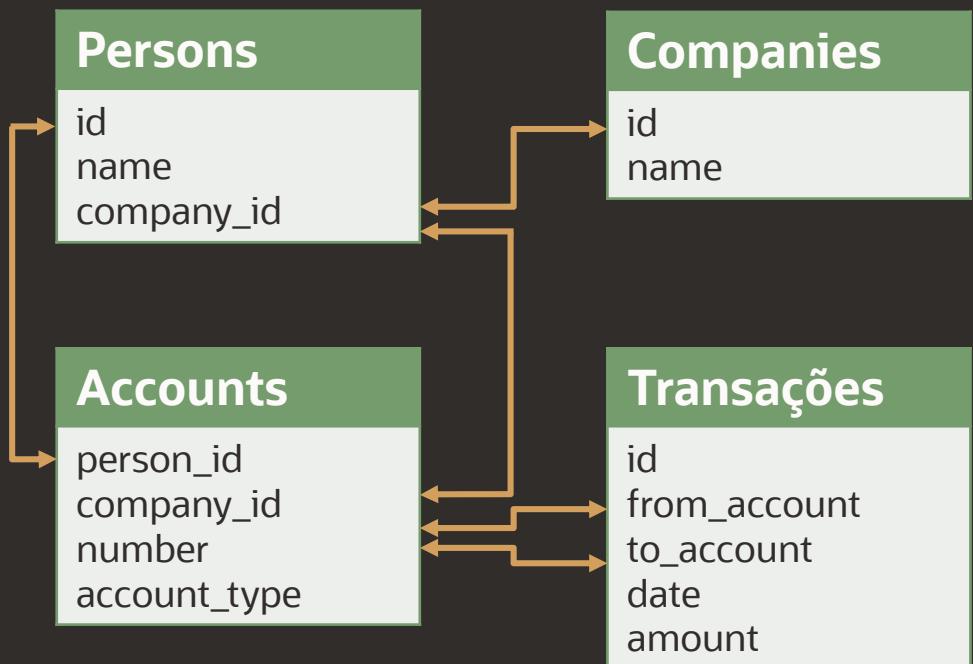
Grafo é a representação de um conjunto de objetos em que seus **pares de objetos** estão, de alguma maneira, **relacionados**.

Apesar de possuir uma estrutura simples, pode representar conceitos complexos e seu maior potencial é alcançado utilizando algoritmos analíticos.



# Grafos

Mapeamento de vértices e arestas



# Grafos

## Mapeamento de vértices e arestas

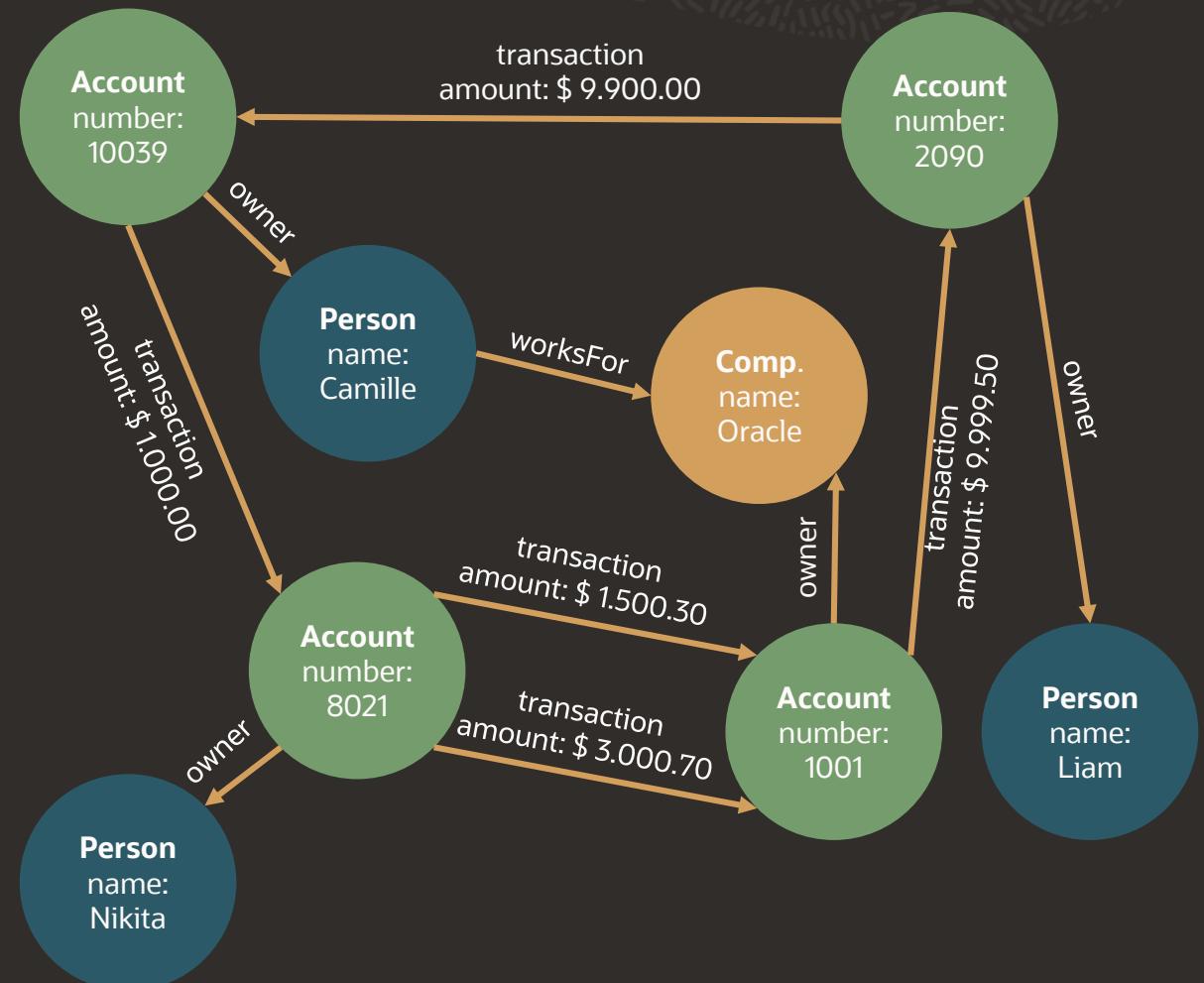
```
/* DEFINE VERTEX */
CREATE PROPERTY GRAPH financial_transactions
VERTEX TABLES (
    Persons LABEL Person PROPERTIES ( name ),
    Companies LABEL Company PROPERTIES ( name ),
    Accounts LABEL Account PROPERTIES ( number )
)
/* CLICK TO CONTINUE... */
```



# Grafos

## Mapeamento de vértices e arestas

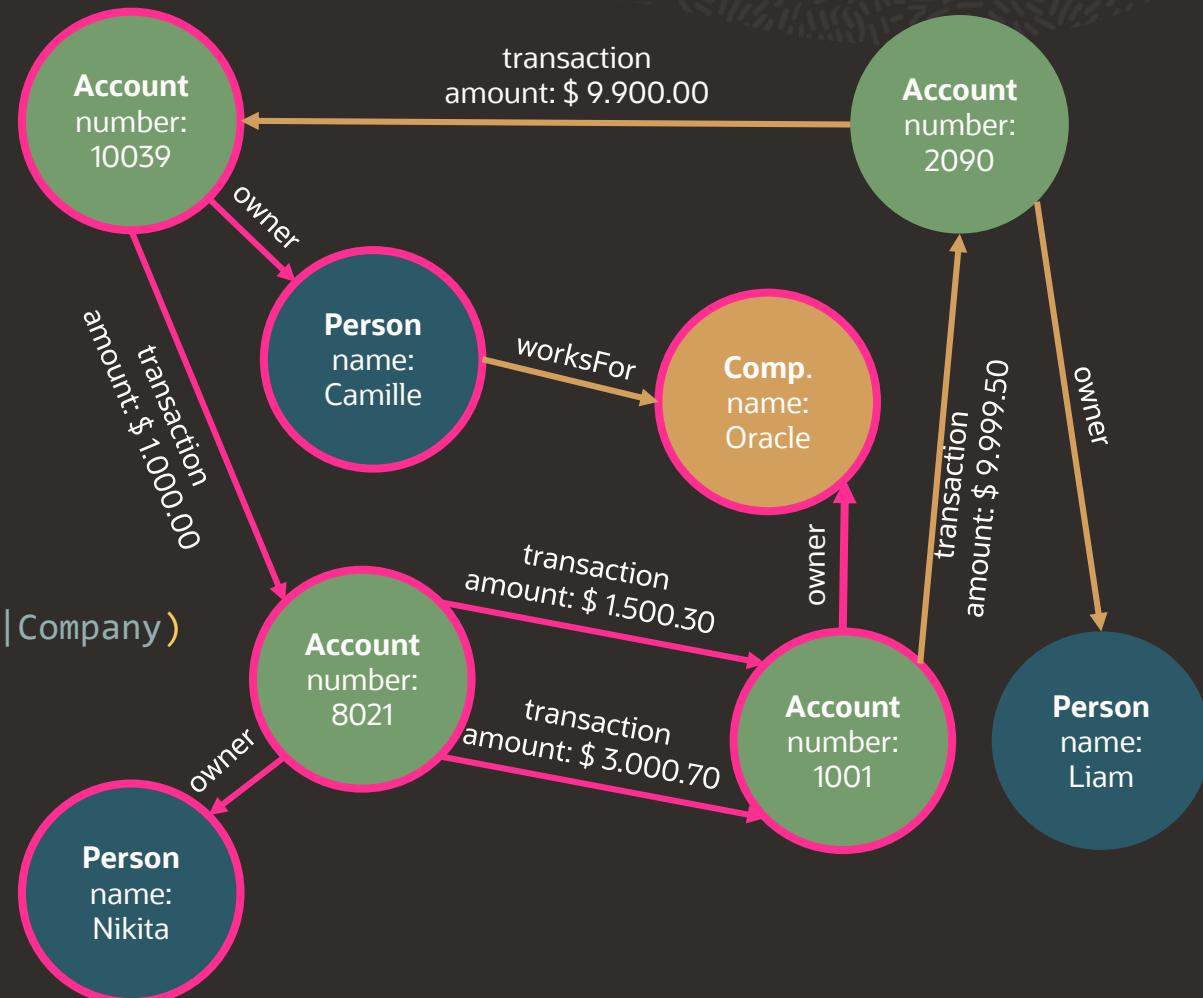
```
/* ... */  
/* DEFINE EDGE */  
  
EDGE TABLES (  
    Transactions  
        SOURCE KEY (from_account) REFERENCES Accounts  
        DESTINATION KEY (to_account) REFERENCES Accounts  
        LABEL transaction PROPERTIES (amount),  
        Accounts AS PersonOwner  
        SOURCE KEY ( id ) REFERENCES Accounts  
        DESTINATION Persons  
        LABEL owner NO PROPERTIES,  
        Accounts AS CompanyOwner  
        SOURCE KEY ( id ) REFERENCES Accounts  
        DESTINATION Companies  
        LABEL owner NO PROPERTIES,  
        Persons AS worksFor  
        SOURCE KEY ( id ) REFERENCES Persons  
        DESTINATION Companies  
        NO PROPERTIES  
)
```



# Grafos

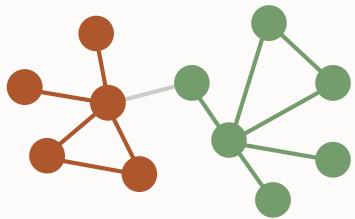
## Mapeamento de vértices e arestas

```
/* PGQL QUERY */
SELECT
    owner.name AS account_holder,
    SUM(t.amount) AS total_transacted
FROM
    MATCH (p:Person) <-[:owner]- (account1:Account),
    MATCH (account1) -[t:transaction]- (account2),
    MATCH (account2:Account) -[:owner]-> (owner:Person|Company)
WHERE p.name = 'Nikita'
GROUP BY owner
```



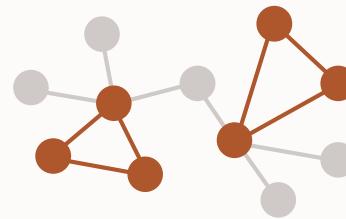
# Grafo

## Análise de relacionamentos



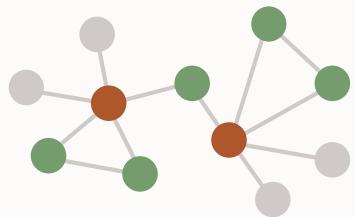
### Detecção de componentes e comunidades

Strongly Connected Components,  
Weakly Connected Components,  
Label Propagation,  
Conductance Minimization,  
Infomap



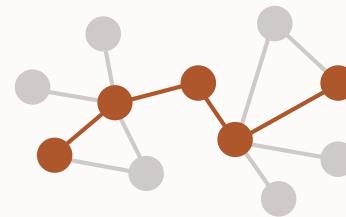
### Avaliação de estruturas

Adamic-Adar Index, Conductance,  
Cycle Detection, Degree Distribution,  
Eccentricity, K-Core, LCC, Modularity,  
Reachability Topological Ordering,  
Triangle Counting



### Ranking and Walking

PageRank, Personalized PageRank,  
Degree Centrality, Closeness Centrality,  
Vertex Betweenness Centrality,  
Eigenvector Centrality, HITS, SALSA,  
Random Walk with Restart

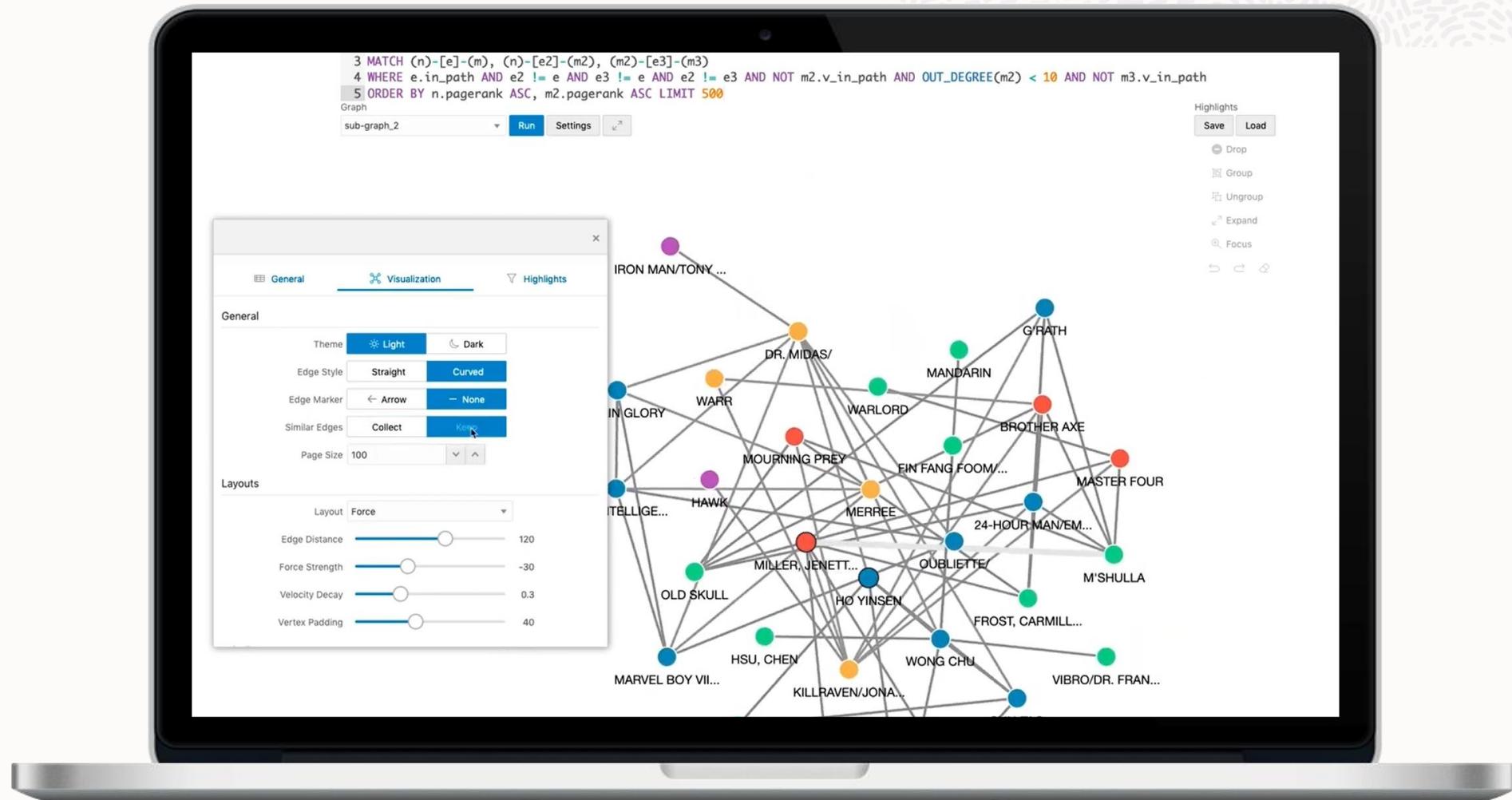


### Path-finding

Shortest Path (Bellman-Ford, Dijkstra,  
Bidirectional Dijkstra), Fattest Path,  
Compute Distance Index,  
Enumerate Simple Paths,  
Fast Path Finding, Hop Distance

# Grafo

## Graph Server and Client

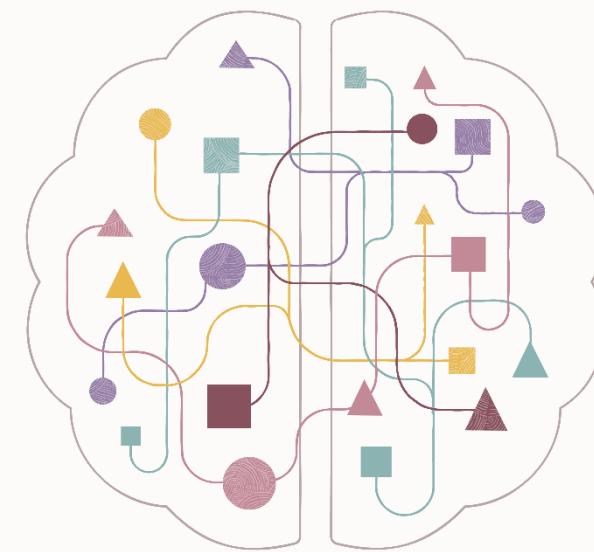


# Machine Learning

Treinamento de modelos e predição de valores



Com a funcionalidade de **Machine Learning**, a Oracle move os algoritmos para os dados, processando-os onde estiverem localizados - **minimizando ou eliminando a movimentação de dados**, obtendo escalabilidade, preservando a segurança e acelerando a implementação até o modelo.



- PL/SQL
- OML4SQL
- OML4R
- OML4Py\*

- Decision Tree
- Explicit Semantic Analysis
- Extreme Gradient Boosting
- Logistic Regression / Generalized Linear Model
- Naïve Bayes
- Neural Network
- Random Forest
- Support Vector Machine

## Classification

- Extreme Gradient Boosting
- Generalized Linear Model
- Neural Network
- Stepwise Regression
- Support Vector Machine

## Regression

- Exponential Smoothing
- CUR Decomposition

## Time Series

## Row Importance

- CUR Decomposition
- Expectation Maximization
- Minimum Description Length

## Att. Importance

- Extreme Gradient Boosting
- Expectation Maximization
- K-Means
- Orthogonal Partitioning

## Ranking

- Explicit Semantic Analysis
- Non-negative Matrix Factorization
- Principal Component Analysis
- Singular Value Decomposition

## Feature Extraction

- One-Class SVM
- MSET-SPRT

## Anomaly Detection

- Apriori

## Association Rules

# Machine Learning

## Treinando um modelo



```
/* CREATE A SETTINGS TABLE */  
CREATE TABLE svmc_sh_sample_settings (  
    setting_name  VARCHAR2(30),  
    setting_value VARCHAR2(4000)  
);
```

# Machine Learning

## Treinando um modelo

```
/* CREATE A SETTINGS TABLE */
CREATE TABLE svmc_sh_sample_settings (
    setting_name  VARCHAR2(30),
    setting_value VARCHAR2(4000)
);
```

```
/* POPULATE THE SETTINGS TABLE */
BEGIN
    INSERT INTO svmc_sh_sample_settings VALUES
        (dbms_data_mining.algo_name,
         dbms_data_mining.algo_support_vector_machines);
    INSERT INTO svmc_sh_sample_settings VALUES
        (dbms_data_mining.svms_kernel_function,
         dbms_data_mining.svms_linear);
    INSERT INTO svmc_sh_sample_settings VALUES
        (dbms_data_mining.clas_weights_table_name,
         'svmc_sh_sample_class_wt');
    INSERT INTO svmc_sh_sample_settings VALUES
        (dbms_data_mining.prep_auto,
         dbms_data_mining.prep_auto_on);
END;
/
```

# Machine Learning

## Treinando um modelo



```
/* CREATE THE MODEL */  
BEGIN  
  DBMS_DATA_MINING.CREATE_MODEL(  
    model_name          => 'SVMC_SH_Clas_sample',  
    mining_function     => dbms_data_mining.classification,  
    data_table_name      => 'mining_data_build_v',  
    case_id_column_name => 'cust_id',  
    target_column_name   => 'affinity_card',  
    settings_table_name => 'svmc_sh_sample_settings');  
END;  
/  
/
```

# Machine Learning

## Treinando um modelo



```
/* CREATE THE MODEL */
BEGIN
  DBMS_DATA_MINING.CREATE_MODEL(
    model_name          => 'SVMC_SH_Clas_sample',
    mining_function     => dbms_data_mining.classification,
    data_table_name     => 'mining_data_build_v',
    case_id_column_name => 'cust_id',
    target_column_name  => 'affinity_card',
    settings_table_name => 'svmc_sh_sample_settings');
END;
/
```

```
/* USE THE MODEL */
SELECT cust_gender,
       COUNT(*) AS cnt,
       ROUND(AVG(age)) AS avg_age
  FROM mining_data_apply_v
 WHERE PREDICTION(SVMC_SH_Clas_sample
                   USING *) = 1
 GROUP BY cust_gender
 ORDER BY cust_gender;
```



# GitHub

Exemplos disponíveis em:  
**oracle/oracle-db-examples**

**Marcos Arancibia**

Product Manager  
Data Science, Advanced Analytics / Data Mining

# Blockchain Table

Prevenção de manipulações nos dados

Tabelas blockchain utilizam **criptografia para encadear** seus registros. Os registros tornam-se imutáveis, auxiliando na identificação e **prevenção de manipulações não autorizadas.** Permite que a cadeia possa ser verificada e assinada por seus participantes.

ID	User	Value	Hash
1	Tom	500	ADSJS
2	Carol	176	%SHS
3	Steve	500	SH@1
4	John	176	DHD3
5	Mike	332	*EGG
6	Sarah	632	AH11
7	Eve	25	LIO\$
8	Prisha	850	SHS4

**BLOCKCHAIN TABLE**

# Blockchain Table

Prevenção de manipulações nos dados

```
/* CREATE A BLOCKCHAIN TABLE */
CREATE BLOCKCHAIN TABLE ledger_emp (
    employee_id NUMBER
    , employee_name VARCHAR2(128)
    , salary      NUMBER
)
NO DROP UNTIL 31 DAYS IDLE
NO DELETE LOCKED
HASHING USING "SHA2_512"
VERSION "v1";
```

ID	User	Value	Hash	Link
1	Tom	500	ADSJS	
2	Carol	176	%SHS	
3	Steve	500	SH@1	
4	John	176	DHD3	
5	Mike	332	*EGG	
6	Sarah	632	AH11	
7	Eve	25	LIO\$	
8	Prisha	850	SHS4	

BLOCKCHAIN TABLE

# Blockchain Table

Prevenção de manipulações nos dados

```
/* INSERT DATA */  
INSERT INTO ledger_emp VALUES (1,'Tom',500);  
  
/* DELETE DATA */  
DELETE FROM ledger_emp  
WHERE employee_id = '1';  
ERROR at line 1: ORA-05715:  
operation not allowed on the blockchain table
```

ID	User	Value	Hash	
1	Tom	500	ADSJS	
2	Carol	176	%SHS	
3	Steve	500	SH@1	
4	John	176	DHD3	
5	Mike	332	*EGG	
6	Sarah	632	AH11	
7	Eve	25	LIO\$	
8	Prisha	850	SHS4	

BLOCKCHAIN TABLE

# Blockchain Table

Prevenção de manipulações nos dados

```
/* VERIFY NO ONE ATTEMPTED TO MANIPULATE DATA */  
DBMS_BLOCKCHAIN_TABLE.VERIFY_ROWS(  
  'AUDITOR'  
, 'ledger_emp'  
, ...  
, number_of_rows_verified => v_row);
```

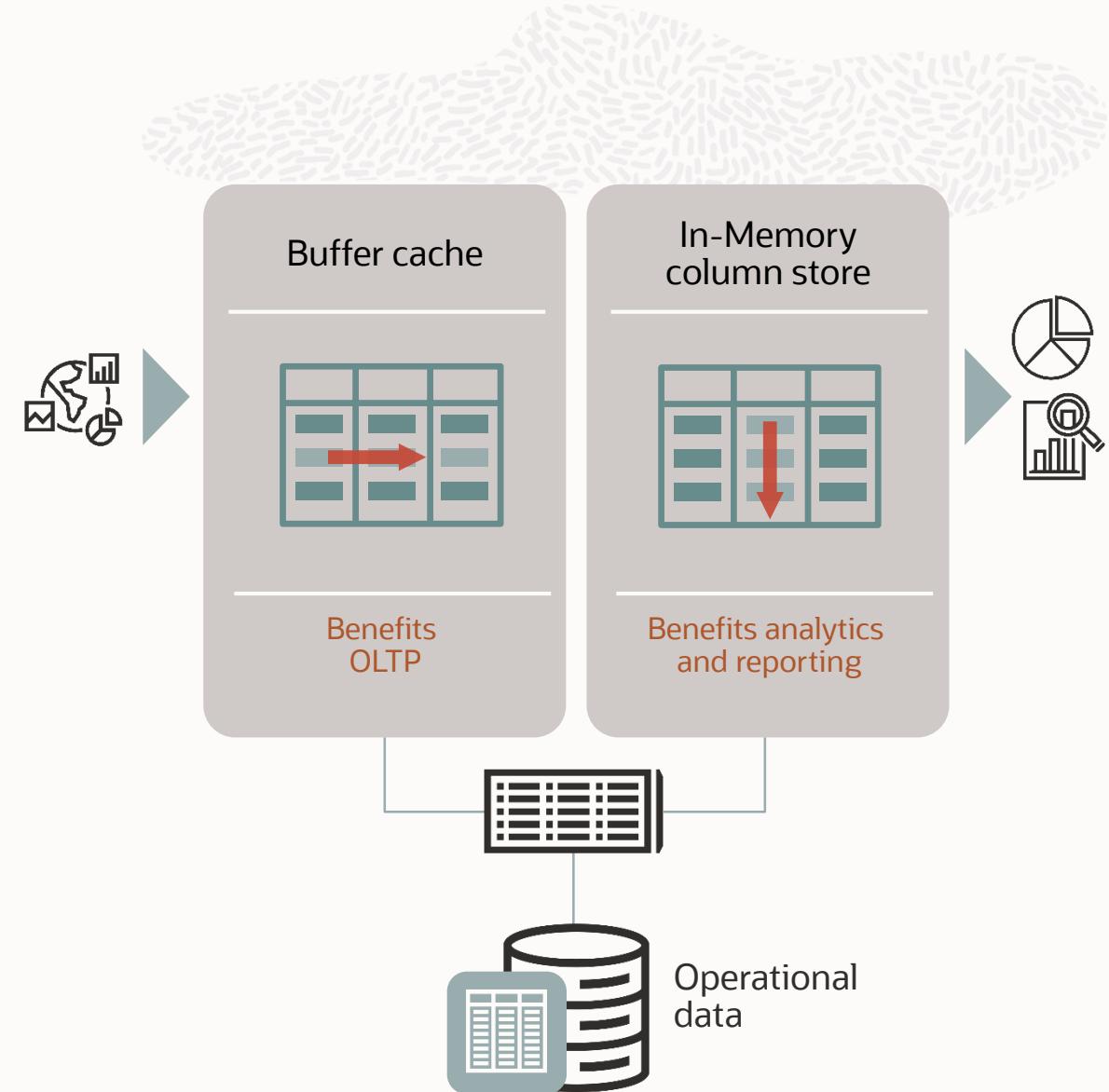
ID	User	Value	Hash	
1	Tom	500	ADSJS	
2	Carol	176	%SHS	
3	Steve	500	SH@1	
4	John	176	DHD3	
5	Mike	332	*EGG	
6	Sarah	632	AH11	
7	Eve	25	LIO\$	
8	Prisha	850	SHS4	

BLOCKCHAIN TABLE

# Database In-Memory

Transacional e analítico de alto desempenho

**In-memory Column Store** é uma porção opcional do System Global Area (SGA) para armazenar **cópias de tabelas**, partições de tabelas ou outros objetos do banco de dados.



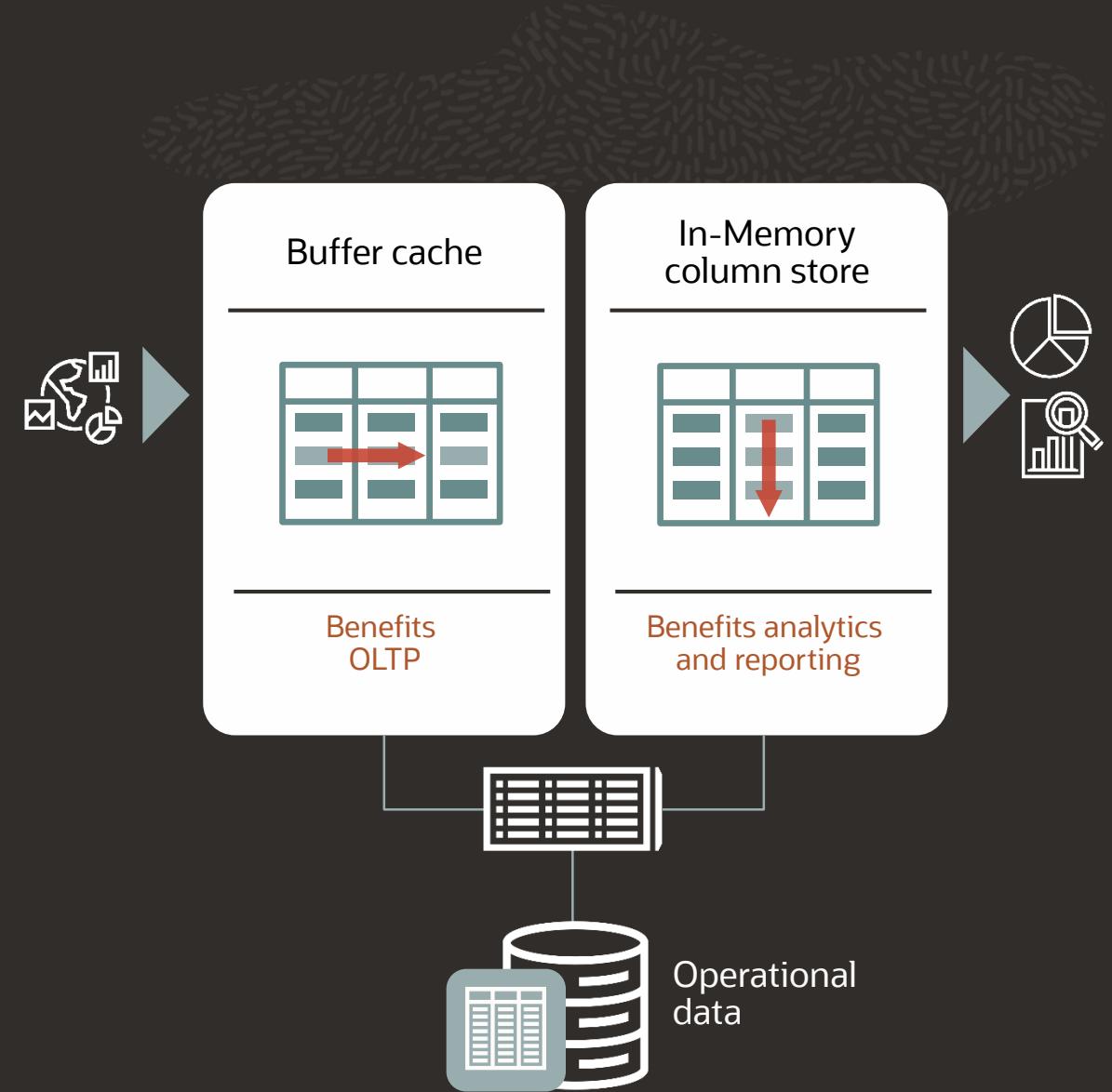
# Database In-Memory

Transacional e analítico de alto desempenho

```
/* ENABLE TABLE FOR IN-MEMORY */
CREATE TABLE transactions (
    id          NUMBER(5)
  ,from_acc    NUMBER(5)
  ,to_acc      NUMBER(5)
  ,trans_date  DATE
  ,amount      NUMBER(7,2)
) INMEMORY;

/* OR */

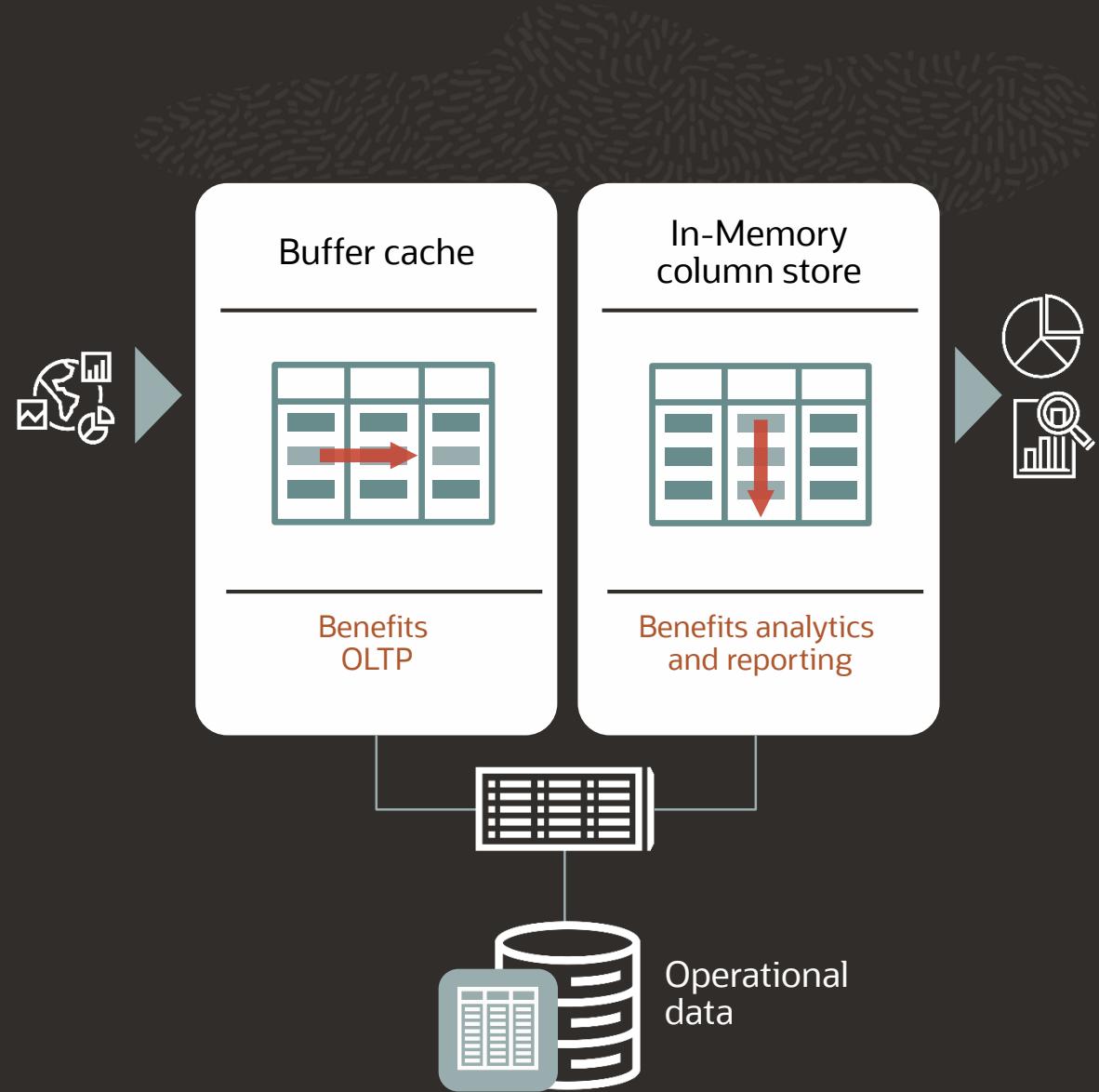
ALTER TABLE transactions INMEMORY;
```



# Database In-Memory

Transacional e analítico de alto desempenho

```
/* ENABLE IN-MEMORY BY PARTITION */  
CREATE TABLE transactions (  
    ...  
) PARTITION BY RANGE ...  
    (PARTITION p1 ...  
        INMEMORY MEMCOMPRESS FOR DML  
,PARTITION p2 ...  
        INMEMORY MEMCOMPRESS FOR QUERY  
,PARTITION p3 ...  
        INMEMORY MEMCOMPRESS FOR CAPACITY  
    ...  
,PARTITION p200 ...  
    NO INMEMORY  
);
```



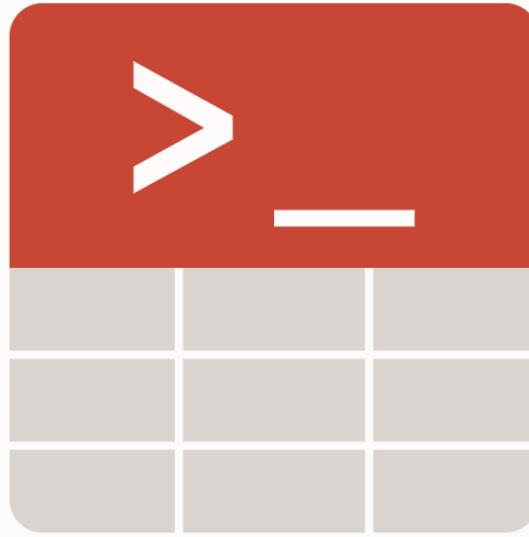
# Database In-Memory

Transacional e analítico de alto desempenho



```
/* VERIFY TABLES IN-MEMORY */  
SELECT table_name, inmemory, inmemory_priority, inmemory_compression FROM user_tables;
```

TABLE_NAME	INMEMORY	INMEMORY_PRIORITY	INMEMORY_COMPRESSION
TABLE_01	ENABLED	NONE	FOR QUERY LOW
TABLE_02	ENABLED	HIGH	FOR CAPACITY LOW
TABLE_03	ENABLED	LOW	FOR CAPACITY HIGH
TABLE_04	DISABLED	-	-



# Live SQL

Learn and share SQL

## Tutorial

Enabling and disabling tables for the  
in-memory column store

## Workshop

Boost Analytics Performance with Oracle  
Database In-Memory

## My Oracle Support

Oracle Database In-Memory Advisor  
(Doc ID 1965343.1)

# Low Code Dev

Desenvolvimento rápido de aplicações com APEX

Crie de maneira **rápida e fácil** aplicações web responsivas e **orientada a dados**.

Com **menos código** para escrever, as equipes podem prototipar, inovar e entregar rapidamente as demandas de negócio.



# ā'pěk s

(#orclapex)



# Low Code Dev

Desenvolvimento rápido de aplicações com APEX



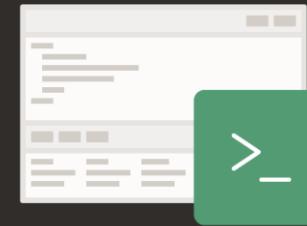
IDE de desenvolvimento  
pelo **web browser**.

Definições do aplicativo são  
armazenados como **metadados**.

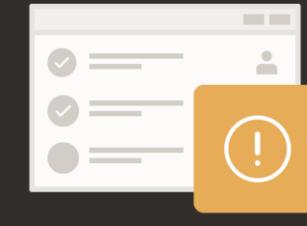
Integração com os dados locais  
permite **acesso eficiente** aos dados.



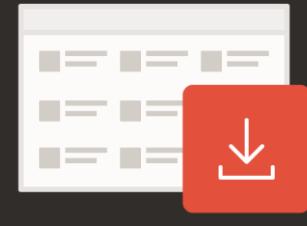
App Builder



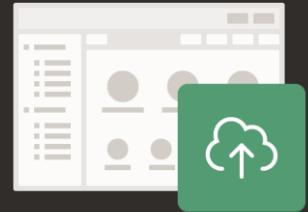
SQL Workshop



Team Development



App Gallery



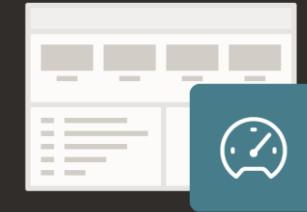
RESTful Services



Object Browser



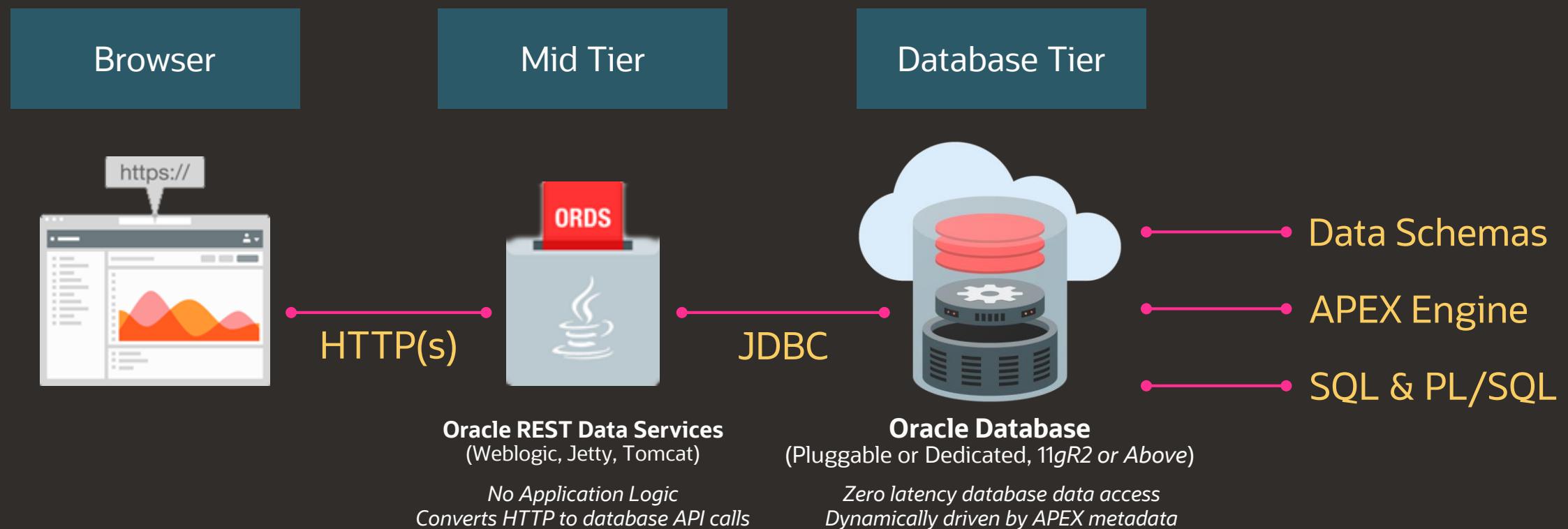
Data Loading



Monitor

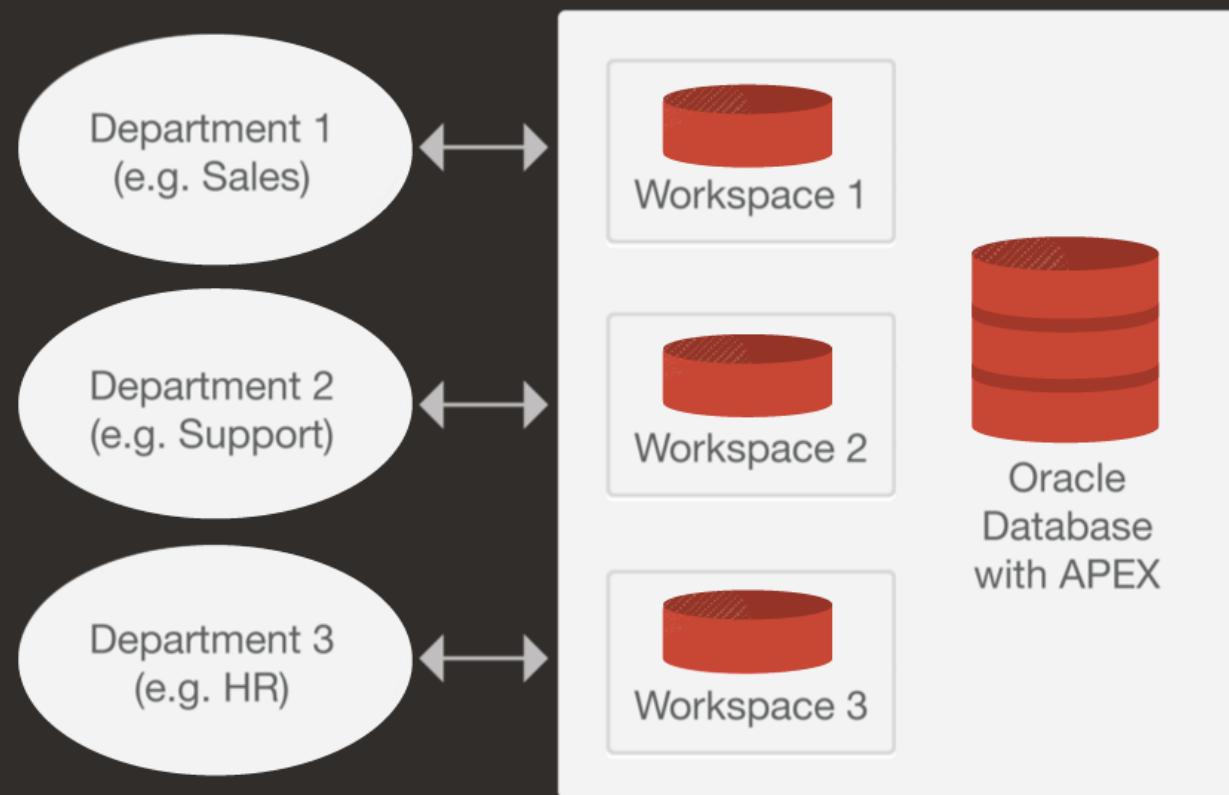
# Low Code Dev

## Arquitetura



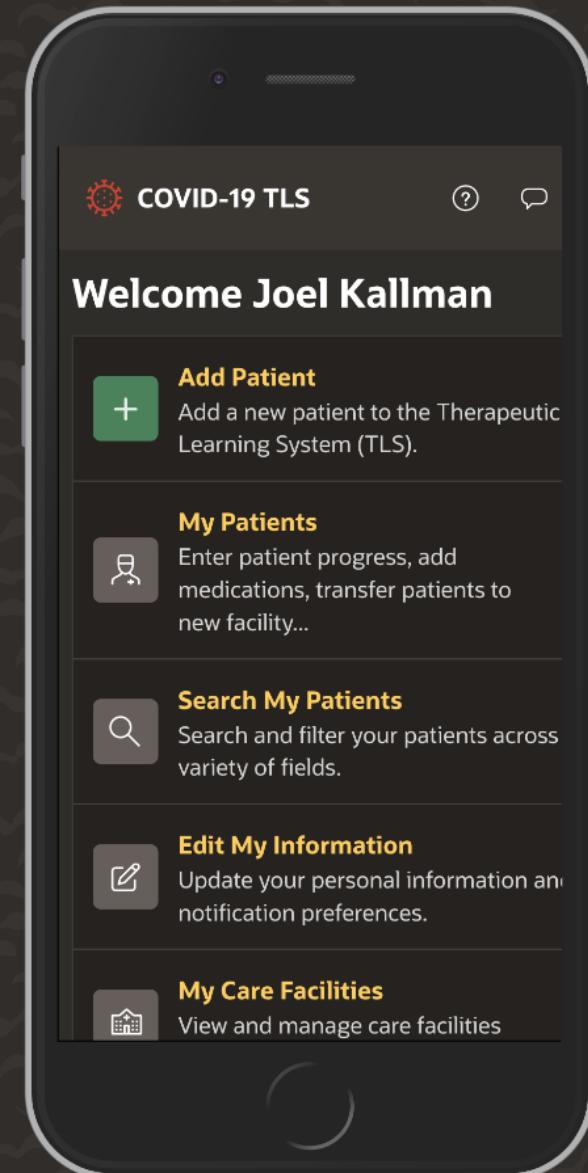
# Low Code Dev

## Workspaces



# 12 dias

foi o tempo necessário para desenvolver o COVID-19 *Therapeutic Learning System* (TLS).





# Live Labs

## Laboratórios

Low code development with ADB

Advanced low code development:  
Oracle APEX on ADB

# Text

Indexe, pesquise e analise textos



## Pesquisa de texto com múltiplas estratégias:

- Palavra-chave
- Consulta de contexto
- Operações booleanas
- Correspondência de padrões
- e mais!

### TEXTO:

“Objeto: Aquisição de itens de hardware e software para compor a plataforma onde serão hospedados os bancos de dados baseados em sistema gerenciador de banco de dados (SGBD) Oracle e que compõem o projeto 'unificação' da base de dados corporativa da organização.”

# Text

## Indexe, pesquise e analise textos

```
/* ENABLE TEXT INDEX*/
CREATE INDEX ix_ds_objeto
    ON tb_contratos (ds_objeto)
INDEXTYPE IS ctxsys.context
PARAMETERS ('sync (on commit)')

/* OR SYNC EVERY 1 MINUTE */
PARAMETERS ('sync ( every SYSDATE+1/24/60 )')
```

### TEXTO:

“Objeto: Aquisição de itens de hardware e software para compor a plataforma onde serão hospedados os bancos de dados baseados em sistema gerenciador de banco de dados (SGBD) Oracle e que compõem o projeto 'unificação' da base de dados corporativa da organização.”

# Text

Indexe, pesquise e analise textos

```
/* QUERY TEXT */  
SELECT ds_objeto  
FROM tb_contratos  
WHERE CONTAINS(ds_objeto, 'Oracle') > 0
```

**SCORE: 12**

**TEXTO:**

“Objeto: Aquisição de itens de hardware e software para compor a plataforma onde serão hospedados os bancos de dados baseados em sistema gerenciador de banco de dados (SGBD) **Oracle** e que compõem o projeto 'unificação' da base de dados corporativa da organização.”

# Text

Indexe, pesquise e analise textos

```
/* QUERY TEXT */  
SELECT ds_objeto  
FROM tb_contratos  
WHERE CONTAINS(ds_objeto, 'Oracle') > 0  
  
/* QUERY TEXT WITH FUZZY*/  
SELECT ds_objeto  
FROM tb_contratos  
WHERE CONTAINS(ds_objeto, 'fuzzy((Oracale))') > 0
```

↑  
Erro de digitação

**SCORE: 12**

**TEXTO:**

“Objeto: Aquisição de itens de hardware e software para compor a plataforma onde serão hospedados os bancos de dados baseados em sistema gerenciador de banco de dados (SGBD) **Oracle** e que compõem o projeto 'unificação' da base de dados corporativa da organização.”

# Text

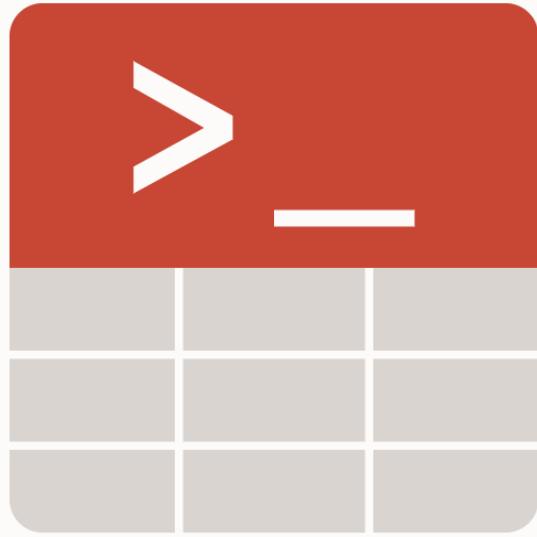
Indexe, pesquise e analise textos

```
/* QUERY TEXT COMBINING OPERATORS */  
SELECT ds_objeto  
FROM tb_contratos  
WHERE CONTAINS(  
    ds_objeto,  
    'fuzzy((Oracle))'  
ACCUM fuzzy((hardware))  
ACCUM fuzzy((sgbd))' ) > 0
```

**SCORE: 71**

**TEXTO:**

“Objeto: Aquisição de itens de **hardware** e software para compor a plataforma onde serão hospedados os bancos de dados baseados em sistema gerenciador de banco de dados (**SGBD**) **Oracle** e que compõem o projeto 'unificação' da base de dados corporativa da organização.”



# Live SQL

Learn and share SQL

## Tutorial

[Oracle Text Quickstart](#)

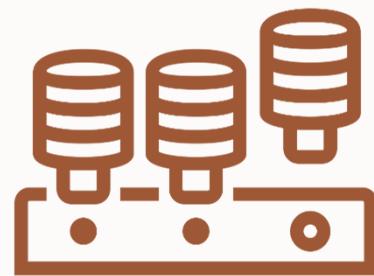
## Técnicas

[Pre-loading Oracle Text indexes into Memory](#)



# Banco de Dados Convergente

Resumo dos benefícios



**Eficiência Técnica**



**Eficiência Operacional**

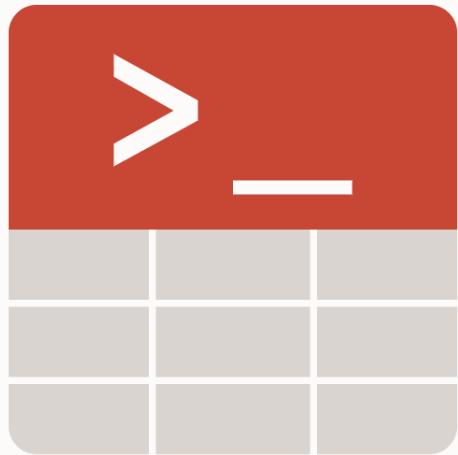


**Agilidade**



**Simplicidade**

# Aprenda e compartilhe



**Live SQL**



**Live Labs**



**University**

# Always Free

Oracle Cloud Infrastructure



## Databases

**2**

Bancos de dados

Cada um com:

1 OCPU

20 GB de Armaz.

Escolha entre os Autonomous:

Transaction Processing

Data Warehouse

JSON

## Compute

**2**

Máquinas Virtuais

Cada um com:

1/8 OCPU

1 GB de Memória RAM

## Storage

**2**

Volumes em Blocos

Cada um com:

50 GB de Armaz.

## Additional Services

### Load Balancer

1 instância

10 Mbps bandwith

### Monitoring

500 mi. ingestion datapoints

1 bi. retrieval datapoints

### Notifications

1 mi. p/ https por mês

1.000 p/ e-mail por mês

### Outbound Data Transfer

10 TB por mês

# Oracle Cloud Modo Gratuito

<https://bit.ly/TDCCConnections2022>



## Uso Livre (Always Free)

Serviços que você pode  
usar por tempo  
ilimitado

+

## Avaliação Gratuita de 30 dias

**US\$ 500** em créditos gratuitos

# O que está incluído no Oracle Cloud – Modo Gratuito (Free Tier)?

## Uso Livre (Always Free)

Serviços em nuvem de Uso Livre:

- Dois Oracle Autonomous Databases com ferramentas avançadas como Oracle APEX e Oracle SQL Developer
- Duas VMs de Computação AMD
- Até 4 instâncias em ARM Ampere A1 Compute
- Armazenamento de Bloco, Objetos e Arquivos; Balanceador de Carga e Saída de Dados; Monitoramento e Notificações

## Avaliação Gratuita de 30 dias

US\$ 500 em créditos gratuitos (exclusivo para os participantes do TDC Connections 2022)

- Acesso a uma ampla variedade de serviços da Oracle Cloud por 30 dias, incluindo Bancos de Dados, Análise Avançada, Computação e Container Engine for Kubernetes
- Até oito instâncias em todos os serviços disponíveis
- Até 5 TB de armazenamento

Mais detalhes em: <https://www.oracle.com/br/cloud/free/>

**ORACLE**

