

Análise de Dados com Base em Processamento Massivo em Paralelo

Aula 1: Introdução

Cristina Dutra de Aguiar
ICMC/USP
cdac@icmc.usp.br



Agenda

- Business Intelligence
- Data Warehousing
- Diferenças entre os Ambientes Operacional e Informacional

Agenda

- Business Intelligence
- Data Warehousing
- Diferenças entre os Ambientes Operacional e Informacional

Business Intelligence (BI)

- Processo de transformação dos **dados** em **informação** e depois em **conhecimento**

Business Intelligence (BI)

- Processo de transformação dos **dados** em **informação** e depois em **conhecimento**

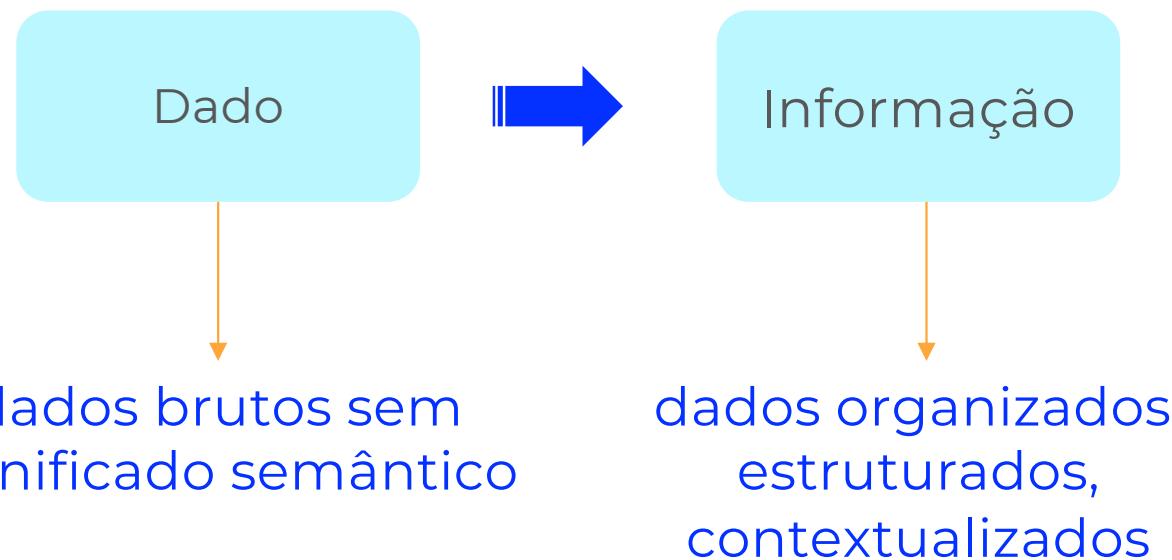
Dado



dados brutos sem
significado semântico

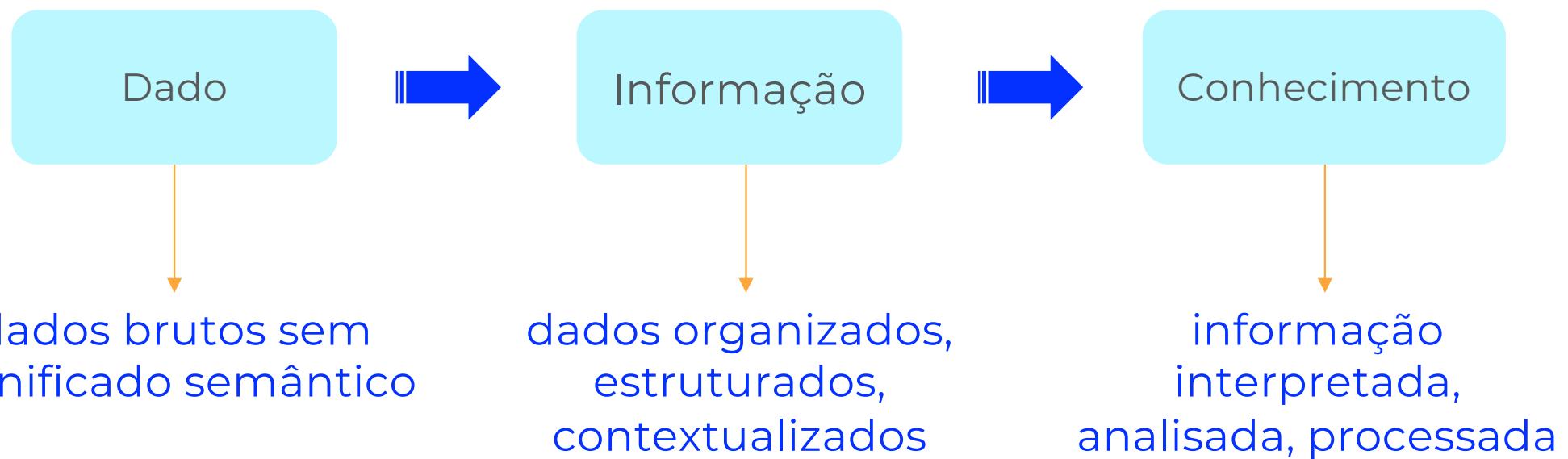
Business Intelligence (BI)

- Processo de transformação dos **dados** em **informação** e depois em **conhecimento**



Business Intelligence (BI)

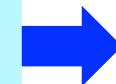
- Processo de transformação dos **dados** em **informação** e depois em **conhecimento**



Business Intelligence (BI)

- Processo de transformação dos **dados** em **informação** e depois em **conhecimento**

Possui o valor mais agregado
Orienta as ações da empresa
Possibilita a tomada de decisão



Conhecimento

informação
interpretada,
analisada, processada

Objetivos

- Satisfazer às necessidades dos usuários de sistemas de suporte à decisão
 - Analisar de forma eficiente e eficaz os dados corporativos
 - Compreender melhor a situação do negócio
 - Melhorar o processo de tomada de decisão estratégica
- Fornecer um conjunto de processos para
 - Produzir a informação certa, para a pessoa certa, na hora certa

Objetivos

- Satisfazer às necessidades dos usuários de sistemas de suporte à decisão
 - Analisar de forma eficiente e eficaz os dados corporativos
 - Compreender melhor a situação do negócio
 - Melhorar o processo de tomada de decisão estratégica
- Fornecer um conjunto de processos para
 - Produzir a **informação certa**, para a **pessoa certa**, na **hora certa**

Pensamento Motivacional

A obtenção de **informações estratégicas**, relativas ao contexto de **tomada de decisão**, é de suma importância para o sucesso de uma empresa.

Tais informações permitem à empresa um **planejamento rápido** frente às mudanças nas condições do negócio, essencial na atual conjuntura de um mercado globalizado.

Tarefas

- Criação de **medidas** (métricas) que indiquem o progresso da empresa com relação às suas metas
- Geração de **relatórios** que possibilitem análises complexas e que possuam visualização apropriada
- **Uso exploratório** das informações com possibilidade de identificar tendências e realizar previsões
- Uso de ferramentas que possibilitem o **trabalho colaborativo** e que ofereçam suporte desde a obtenção dos dados até a geração do conhecimento
- **Gerenciamento do conhecimento** para realizar a tomada de decisão estratégica bem fundamentada, resultando em ações bem sucedidas que garantam um maior retorno sobre o investimento

Tarefas

- Criação de **medidas** (métricas) que indiquem o progresso da empresa com relação às suas metas
- Geração de relatórios que possibilitem análises complexas e que possuam visualização apropriada
- Uso exploratório das informações com possibilidade de identificar tendências e realizar previsões
- Uso de ferramentas que possibilitem o trabalho colaborativo e que ofereçam suporte desde a obtenção dos dados até a geração do conhecimento
- Gerenciamento do conhecimento para realizar a tomada de decisão estratégica bem fundamentada, resultando em ações bem sucedidas que garantam um maior retorno sobre o investimento

Tarefas

- Criação de medidas (métricas) que indiquem o progresso da empresa com relação às suas metas
- Geração de **relatórios** que possibilitem **análises complexas** e que possuam **visualização** apropriada
- Uso exploratório das informações com possibilidade de identificar tendências e realizar previsões
- Uso de ferramentas que possibilitem o trabalho colaborativo e que ofereçam suporte desde a obtenção dos dados até a geração do conhecimento
- Gerenciamento do conhecimento para realizar a tomada de decisão estratégica bem fundamentada, resultando em ações bem sucedidas que garantam um maior retorno sobre o investimento

Tarefas

- Criação de medidas (métricas) que indiquem o progresso da empresa com relação às suas metas
- Geração de relatórios que possibilitem análises complexas e que possuam visualização apropriada
- Uso **exploratório** das informações com possibilidade de identificar tendências e realizar previsões
- Uso de ferramentas que possibilitem o trabalho colaborativo e que ofereçam suporte desde a obtenção dos dados até a geração do conhecimento
- Gerenciamento do conhecimento para realizar a tomada de decisão estratégica bem fundamentada, resultando em ações bem sucedidas que garantam um maior retorno sobre o investimento

Tarefas

- Criação de medidas (métricas) que indiquem o progresso da empresa com relação às suas metas
- Geração de relatórios que possibilitem análises complexas e que possuam visualização apropriada
- Uso exploratório das informações com possibilidade de identificar tendências e realizar previsões
- Uso de ferramentas que possibilitem o **trabalho colaborativo** e que ofereçam suporte desde a obtenção dos dados até a geração do conhecimento
- Gerenciamento do conhecimento para realizar a tomada de decisão estratégica bem fundamentada, resultando em ações bem sucedidas que garantam um maior retorno sobre o investimento

Tarefas

- Criação de medidas (métricas) que indiquem o progresso da empresa com relação às suas metas
- Geração de relatórios que possibilitem análises complexas e que possuam visualização apropriada
- Uso exploratório das informações com possibilidade de identificar tendências e realizar previsões
- Uso de ferramentas que possibilitem o trabalho colaborativo e que ofereçam suporte desde a obtenção dos dados até a geração do conhecimento
- **Gerenciamento do conhecimento** para realizar a **tomada de decisão estratégica** bem fundamentada, resultando em ações bem sucedidas que garantam um maior retorno sobre o investimento

Agenda

- ~~Business Intelligence~~
- Data Warehousing
- Diferenças entre os Ambientes Operacional e Informacional

Agenda

- Business Intelligence
- Data Warehousing
- Diferenças entre os Ambientes Operacional e Informacional

Data Warehousing

Engloba arquiteturas, algoritmos e ferramentas que possibilitam que dados selecionados de fontes de dados autônomas, heterogêneas e distribuídas sejam integrados em um único banco de dados, conhecido como *data warehouse (DW)*

Data Warehousing

Engloba arquiteturas, algoritmos e ferramentas que possibilitam que dados selecionados de fontes heterogêneas e distribuídas

Local onde os dados estão fisicamente armazenados

um único banco de dados, conhecido como

data warehouse (DW)

Data Warehousing

Engloba arquiteturas, algoritmos e possibilidades que permitem a integração de dados heterogêneos e distribuídos em um ambiente como um todo, englobando DW, software, hardware e peopleware.

Local onde os dados estão fisicamente armazenados

data warehouse (DW)

Acesso às Informações

- Etapa ETL (extração, transformação e carga)
 - Dados de interesse de cada fonte de dados são extraídos previamente, devendo ser traduzidos, filtrados, integrados aos dados relevantes de outras fontes e finalmente armazenados no DW
- Etapa de análise e consulta
 - As consultas, quando realizadas, são executadas diretamente no DW, sem acessar as fontes de dados originais

Acesso às Informações

- Etapa ETL (extração, transformação e carga)
 - Dados de interesse de cada fonte de dados são extraídos previamente, devendo ser traduzidos, filtrados, integrados aos dados relevantes de outras fontes e finalmente armazenados no DW
- Etapa de análise e consulta
 - As consultas analíticas, quando realizadas, são executadas diretamente no DW, sem acessar as fontes de dados originais

Aplicação 1: Área Médica

- Foco em **número** de pacientes
 - Dados integrados de **pacientes**, tipos de **exame**, **hospitais** nos quais os exames foram feitos e **datas** de coleta dos exames

Aplicação 1: Área Médica

- Foco em **número** de pacientes
 - Dados integrados de **pacientes**, tipos de **exame**, **hospitais** nos quais os exames foram feitos e **datas** de coleta dos exames
- Exemplos de **análise**
 - Qual o número de pacientes que testaram positivo para a COVID-19 por mês?
 - Qual o número de pacientes de cada faixa etária que tiveram complicações devido à COVID-19, considerando cada um dos estados do Brasil?
 - Qual a porcentagem de pacientes que vieram a falecer devido às complicações causadas pela COVID-19 de janeiro a agosto de 2020?

Aplicação 1: Área Médica

- Foco em **número** de pacientes
 - Dados integrados de **pacientes**, tipos de **exame**, **hospitais** nos quais os exames foram feitos e **datas** de coleta dos exames
- Exemplos de **análise**
 - Qual o número de pacientes que testaram positivo para a COVID-19 por mês?
 - Qual o número de pacientes de cada faixa etária que tiveram complicações devido à COVID-19, considerando cada um dos estados do Brasil?
 - Qual a porcentagem de pacientes que vieram a falecer devido às complicações causadas pela COVID-19 de janeiro a agosto de 2020?

Aplicação 1: Área Médica

- Foco em **número** de pacientes
 - Dados integrados de **pacientes**, tipos de **exame**, **hospitais** nos quais os exames foram feitos e **datas** de coleta dos exames
- Exemplos de **análise**
 - Qual o número de pacientes que testaram positivo para a COVID-19 por mês?
 - Qual o número de pacientes de cada faixa etária que tiveram complicações devido à COVID-19, considerando cada um dos estados do Brasil?
 - Qual a porcentagem de pacientes que vieram a falecer devido às complicações causadas pela COVID-19 de janeiro a agosto de 2020?

Aplicação 1: Área Médica

- Foco em **número** de pacientes
 - Dados integrados de **pacientes**, tipos de **exame**, **hospitais** nos quais os exames foram feitos e **datas** de coleta dos exames
- Exemplos de **conhecimento**
 - Curva de evolução de uma determinada doença ao longo dos meses
 - Características dos pacientes (por exemplo, tipo sanguíneo, faixa etária, faixa salarial) mais suscetíveis a uma determinada doença
 - Localidades geográficas que podem ser consideradas como epicentros

Aplicação 1: Área Médica

- Foco em **número** de pacientes
 - Dados integrados de **pacientes**, tipos de **exame**, **hospitais** nos quais os exames foram feitos e **datas** de coleta dos exames
- Exemplos de **conhecimento**
 - Curva de evolução de uma determinada doença ao longo dos meses
 - Características dos pacientes (por exemplo, tipo sanguíneo, faixa etária, faixa salarial) mais suscetíveis a uma determinada doença
 - Localidades geográficas que podem ser consideradas como epicentros

Aplicação 1: Área Médica

- Foco em **número** de pacientes
 - Dados integrados de **pacientes**, tipos de **exame**, **hospitais** nos quais os exames foram feitos e **datas** de coleta dos exames
- Exemplos de **conhecimento**
 - Curva de evolução de uma determinada doença ao longo dos meses
 - Características dos pacientes (por exemplo, tipo sanguíneo, faixa etária, faixa salarial) mais suscetíveis a uma determinada doença
 - Localidades geográficas que podem ser consideradas como epicentros

Aplicação 2: Cadeia de Supermercados

- Foco em **unidades** vendidas de produtos e seus **lucros**
 - Dados integrados de **produtos** vendidos, **promoções** realizadas, **filiais** nas quais os produtos foram vendidos e **datas** das vendas

Aplicação 2: Cadeia de Supermercados

- Foco em **unidades** vendidas de produtos e seus **lucros**
 - Dados integrados de **produtos** vendidos, **promoções** realizadas, **filiais** nas quais os produtos foram vendidos e **datas** das vendas
- Exemplos de **análise**
 - Quais as vendas mensais dos produtos de uma determinada marca nos últimos três anos?
 - Quais as vendas diárias dos produtos nas diferentes filiais, de acordo com as promoções realizadas no período do dia dos namorados e do dia das mães?
 - Quais os lucros obtidos nas vendas de produtos para tratamento estético?

Aplicação 2: Cadeia de Supermercados

- Foco em **unidades** vendidas de produtos e seus **lucros**
 - Dados integrados de **produtos** vendidos, **promoções** realizadas, **filiais** nas quais os produtos foram vendidos e **datas** das vendas
- Exemplos de **análise**
 - Quais as vendas mensais dos produtos de uma determinada marca nos últimos três anos?
 - Quais as vendas diárias dos produtos nas diferentes filiais, de acordo com as promoções realizadas no período do dia dos namorados e do dia das mães?
 - Quais os lucros obtidos nas vendas de produtos para tratamento estético?

Aplicação 2: Cadeia de Supermercados

- Foco em **unidades** vendidas de produtos e seus **lucros**
 - Dados integrados de **produtos** vendidos, **promoções** realizadas, **filiais** nas quais os produtos foram vendidos e **datas** das vendas
- Exemplos de **análise**
 - Quais as vendas mensais dos produtos de uma determinada marca nos últimos três anos?
 - Quais as vendas diárias dos produtos nas diferentes filiais, de acordo com as promoções realizadas no período do dia dos namorados e do dia das mães?
 - Quais os lucros obtidos nas vendas de produtos para tratamento estético?

Aplicação 2: Cadeia de Supermercados

- Foco em **unidades** vendidas de produtos e seus **lucros**
 - Dados integrados de **produtos** vendidos, **promoções** realizadas, **filiais** nas quais os produtos foram vendidos e **datas** das vendas
- Exemplos de **conhecimento**
 - Produtos mais vendidos e menos vendidos e os lucros ou prejuízos associados
 - Impacto das promoções realizadas na venda dos produtos e nos lucros obtidos
 - Filiais deficitárias que precisam ser fechadas ou remodeladas

Aplicação 2: Cadeia de Supermercados

- Foco em **unidades** vendidas de produtos e seus **lucros**
 - Dados integrados de **produtos** vendidos, **promoções** realizadas, **filiais** nas quais os produtos foram vendidos e **datas** das vendas
- Exemplos de **conhecimento**
 - Produtos mais vendidos e menos vendidos e os lucros ou prejuízos associados
 - Impacto das promoções realizadas na venda dos produtos e nos lucros obtidos
 - Filiais deficitárias que precisam ser fechadas ou remodeladas

Aplicação 2: Cadeia de Supermercados

- Foco em **unidades** vendidas de produtos e seus **lucros**
 - Dados integrados de **produtos** vendidos, **promoções** realizadas, **filiais** nas quais os produtos foram vendidos e **datas** das vendas
- Exemplos de **conhecimento**
 - Produtos mais vendidos e menos vendidos e os lucros ou prejuízos associados
 - Impacto das promoções realizadas na venda dos produtos e nos lucros obtidos
 - Filiais deficitárias que precisam ser fechadas ou remodeladas

Aplicação 3: BI Solutions

- Empresa exemplo que será usada ao longo da disciplina



Razão social:

BI Solutions

Slogan:

Desenvolvimento de soluções inteligentes para o seu negócio

Sobre a empresa:

A BI Solutions é uma empresa de desenvolvimento de software totalmente brasileira e com alcance internacional, que implementa soluções inteligentes para atender os clientes dos mais diversos setores de negócio.

Aplicação 3: Folha de Pagamento da BI Solutions

- Foco nos **salários** dos funcionários e na **quantidade** de lançamentos
 - Dados integrados de **funcionários**, **cargos** ocupados por estes, **filiais** nas quais os funcionários trabalham e **datas** de pagamento



BI Solutions

Aplicação 3: Folha de Pagamento da BI Solutions

- Foco nos **salários** dos funcionários e na **quantidade** de lançamentos
 - Dados integrados de **funcionários**, **cargos** ocupados por estes, **filiais** nas quais os funcionários trabalham e **datas** de pagamento
- Exemplos de **análise**
 - Quais os gastos mensais em salários dos funcionários?
 - Quais as filiais que possuem o maior gasto anual em salários de funcionários?
 - Qual a média salarial dos funcionários ocupantes de cargos de nível superior em uma determinada filial no primeiro trimestre de 2019?



Aplicação 3: Folha de Pagamento da BI Solutions

- Foco nos **salários** dos funcionários e na **quantidade** de lançamentos
 - Dados integrados de **funcionários**, **cargos** ocupados por estes, **filiais** nas quais os funcionários trabalham e **datas** de pagamento
- Exemplos de **análise**
 - Quais os gastos mensais em salários dos funcionários?
 - Quais as filiais que possuem o maior gasto anual em salários de funcionários?
 - Qual a média salarial dos funcionários ocupantes de cargos de nível superior em uma determinada filial no primeiro trimestre de 2019?



Aplicação 3: Folha de Pagamento da BI Solutions

- Foco nos **salários** dos funcionários e na **quantidade** de lançamentos
 - Dados integrados de **funcionários**, **cargos** ocupados por estes, **filiais** nas quais os funcionários trabalham e **datas** de pagamento
- Exemplos de **análise**
 - Quais os gastos mensais em salários dos funcionários?
 - Quais as filiais que possuem o maior gasto anual em salários de funcionários?
 - Qual a média salarial dos funcionários ocupantes de cargos de nível superior em uma determinada filial no primeiro trimestre de 2019?



Aplicação 3: Folha de Pagamento da BI Solutions

- Foco nos **salários** dos funcionários e na **quantidade** de lançamentos
 - Dados integrados de **funcionários**, **cargos** ocupados por estes, **filiais** nas quais os funcionários trabalham e **datas** de pagamento
- Exemplos de **conhecimento**
 - Cargos que receberam a maior soma de salários e filiais relacionadas
 - Graus de escolaridade dos funcionários e seus impactos nas médias salariais dos mesmos, bem como nos cargos ocupados
 - Curvas de gastos em salários dos funcionários por mês nos últimos anos



Aplicação 3: Folha de Pagamento da BI Solutions

- Foco nos **salários** dos funcionários e na **quantidade** de lançamentos
 - Dados integrados de **funcionários**, **cargos** ocupados por estes, **filiais** nas quais os funcionários trabalham e **datas** de pagamento
- Exemplos de **conhecimento**
 - Cargos que receberam a maior soma de salários e filiais relacionadas
 - Graus de escolaridade dos funcionários e seus impactos nas médias salariais dos mesmos, bem como nos cargos ocupados
 - Curvas de gastos em salários dos funcionários por mês nos últimos anos



Aplicação 3: Folha de Pagamento da BI Solutions

- Foco nos **salários** dos funcionários e na **quantidade** de lançamentos
 - Dados integrados de **funcionários**, **cargos** ocupados por estes, **filiais** nas quais os funcionários trabalham e **datas** de pagamento
- Exemplos de **conhecimento**
 - Cargos que receberam a maior soma de salários e filiais relacionadas
 - Graus de escolaridade dos funcionários e seus impactos nas médias salariais dos mesmos, bem como nos cargos ocupados
 - Curvas de gastos em salários dos funcionários por mês nos últimos anos



Questionamento

- Esse tipo de análise são possíveis de serem realizados usando os sistemas existentes?
 - Aplicações de banco de dados *stand-alone*
 - Aplicações desenvolvidas de forma centralizada
 - Sistemas legados
 - Uso de planilhas
- Limitação
 - Análises muito custosas com tempos de respostas proibitivos para a produção da informação certa, na hora certa, para a pessoa certa

Questionamento

- Esse tipo de análise são possíveis de serem realizados usando os sistemas existentes?
 - Aplicações de banco de dados *stand-alone*
 - Aplicações desenvolvidas de forma centralizada
 - Sistemas legados
 - Uso de planilhas
- Limitação
 - Análises muito custosas com tempos de respostas proibitivos para a produção da **informação certa**, na **hora certa**, para a **pessoa certa**

Análise usando Sistemas Existentes

- Exemplos de desafios

- Dados de interesse de análise encontram-se espalhados em diferentes fontes de dados, assumem diferentes formatos e requerem processos de limpeza acurados
- Aplicações encontram-se projetadas com foco em normalização, visando diminuir ou até mesmo eliminar a redundância
- O foco em normalização impacta a complexidade de se especificar consultas analíticas
- A complexidade das consultas impacta no desempenho das mesmas
- O tratamento de dados temporais usualmente é incipiente

Análise usando Sistemas Existentes

- Exemplos de desafios

- Dados de interesse de análise encontram-se espalhados nos diferentes sistemas, assumem diferentes formatos e requerem processos de limpeza acurados
- Aplicações encontram-se projetadas com foco em normalização, visando diminuir ou até mesmo eliminar a redundância
- O foco em normalização impacta a complexidade de se especificar consultas analíticas
- A complexidade das consultas impacta no desempenho das mesmas
- O tratamento de dados temporais usualmente é incipiente

Análise usando Sistemas Existentes

- Exemplos de desafios

- Dados de interesse de análise encontram-se espalhados nos diferentes sistemas, assumem diferentes formatos e requerem processos de limpeza acurados
- Aplicações encontram-se projetadas com foco em normalização, visando diminuir ou até mesmo eliminar a redundância
- O foco em normalização impacta a complexidade de se especificar consultas analíticas
- A complexidade das consultas impacta no desempenho das mesmas
- O tratamento de dados temporais usualmente é incipiente

Análise usando Sistemas Existentes

- Exemplos de desafios

- Dados de interesse de análise encontram-se espalhados nos diferentes sistemas, , assumem diferentes formatos e requerem processos de limpeza acurados
- Aplicações encontram-se projetadas com foco em normalização, visando diminuir ou até mesmo eliminar a redundância
- O foco em normalização impacta a complexidade de se especificar consultas analíticas
- A complexidade das consultas impacta no desempenho das mesmas
- O tratamento de dados temporais usualmente é incipiente

Análise usando Sistemas Existentes

- Exemplos de desafios

- Dados de interesse de análise encontram-se espalhados nos diferentes sistemas, assumem diferentes formatos e requerem processos de limpeza acurados
- Aplicações encontram-se projetadas com foco em normalização, visando diminuir ou até mesmo eliminar a redundância
- O foco em normalização impacta a complexidade de se especificar consultas analíticas
- A complexidade das consultas impacta no desempenho das mesmas
- O tratamento de dados temporais usualmente é incipiente

Análise usando Sistemas Existentes

- Exemplos de desafios

- Dados de interesse de análise encontram-se espalhados nos diferentes sistemas, assumem diferentes formatos e requerem processos de limpeza acurados
- Aplicações encontram-se projetadas com foco em normalização, visando diminuir ou até mesmo eliminar a redundância
- O foco em normalização impacta a complexidade de se especificar consultas analíticas
- A complexidade das consultas impacta no desempenho das mesmas
- O tratamento de dados temporais usualmente é incipiente

Vantagens do Data Warehousing

- Análises podem ser realizadas eficientemente
 - DW contém dados integrados
 - DW é projetado com foco em assuntos de interesse
 - DW modela explicitamente o aspecto temporal
- Maior disponibilidade dos dados
 - Consultas são executadas diretamente no DW sem acessar as fontes originais
- Autonomia das fontes de dados originais
 - Processamento local nas fontes de dados originais não é afetado por causa da participação destes no ambiente de *data warehousing*

Vantagens do Data Warehousing

- Análises podem ser realizadas eficientemente
 - DW contém dados integrados, cuja heterogeneidade já foi eliminada
 - DW é projetado com foco em assuntos de interesse
 - DW modela explicitamente o aspecto temporal
- Maior disponibilidade dos dados
 - Consultas são executadas diretamente no DW sem acessar as fontes originais
- Autonomia das fontes de dados originais
 - Processamento local nas fontes de dados originais não é afetado por causa da participação destes no ambiente de *data warehousing*

Vantagens do Data Warehousing

- Análises podem ser realizadas eficientemente
 - DW contém dados integrados, cuja heterogeneidade já foi eliminada
 - DW é projetado com foco em assuntos de interesse
 - DW modela explicitamente o aspecto temporal
- Maior disponibilidade dos dados
 - Consultas são executadas diretamente no DW sem acessar as fontes originais
- Autonomia das fontes de dados originais
 - Processamento local nas fontes de dados originais não é afetado por causa da participação destes no ambiente de *data warehousing*

Vantagens do Data Warehousing

- Análises podem ser realizadas eficientemente
 - DW contém dados integrados, cuja heterogeneidade já foi eliminada
 - DW é projetado com foco em assuntos de interesse
 - DW modela explícitamente os dados
- Maior disponibilidade de dados ... e muito mais ...
 - Consultas são executadas diretamente no ambiente de data warehousing, sem necessidade de acessar as fontes originais
- Autonomia das fontes de dados originais
 - Processamento local nas fontes de dados originais não é afetado por causa da participação destes no ambiente de *data warehousing*

Agenda

- Business Intelligence
- Data Warehousing
- Diferenças entre os Ambientes Operacional e Informacional

Agenda

- Inteligência do Negócio
- Data Warehousing
- Diferenças entre os Ambientes Operacional e Informacional

Separação entre os Ambientes

- Ambientes fundamentalmente diferentes
 - Dados
 - Tecnologias
 - Usuários
 - Necessidades de processamento
 - Necessidades de segurança
 - Requisitos de desempenho das aplicações

Ambientes Operacional e Informacional

- Ambiente Operacional
 - Constituído por aplicações que oferecem **suporte ao dia a dia** do negócio
 - Sistemas existentes
- Ambiente Informacional
 - Constituído por aplicações que analisam o negócio
 - *Data warehousing*

Ambientes Operacional e Informacional

- Ambiente Operacional
 - Constituído por aplicações que oferecem suporte ao dia a dia do negócio
 - Sistemas existentes
- Ambiente Informacional
 - Constituído por aplicações que **analisam** o negócio
 - *Data warehousing*

Ambientes Operacional e Informacional

- Ambiente Operacional

- Constituído por aplicações que oferecem suporte ao dia a dia do negócio
- Sistemas existentes

- Ambiente Informacional

- Constituído por aplicações que analisam o negócio
- *Data warehousing*

DW é mantido
separadamente dos
bancos de dados
operacionais

Diferenças entre os Ambientes

	Ambiente Operacional	Ambiente Informacional
Principal Característica	voltado ao processamento de transações (OLTP)	voltado ao processamento de consultas (OLAP)
Tipos de Operação mais Frequentes	inserção remoção atualização	leitura (consulta)
Foco do Desempenho	produtividade das transações	produtividade das consultas

Diferenças entre os Ambientes

	Ambiente Operacional	Ambiente Informacional
Principal Característica	voltado ao processamento de transações (OLTP)	voltado ao processamento de consultas (OLAP)
Tipos de Operação mais Frequentes	inserção remoção atualização	leitura (consulta)
Foco do Desempenho	produtividade das transações	produtividade das consultas

Diferenças entre os Ambientes

	Ambiente Operacional	Ambiente Informacional
Tipos de Usuários	administradores do sistema, projetistas, usuários finais	usuários de SSD (ex.: executivos, analistas, gerentes)
Número de Usuários Concorrentes	grande	relativamente pequeno
Interações com os Usuários	estáticas, predefinidas	dinâmicas, exploratórias

Diferenças entre os Ambientes

	Ambiente Operacional	Ambiente Informacional
Tipos de Usuários	administradores do sistema, projetistas, usuários finais	usuários de SSD (ex.: executivos, analistas, gerentes)
Número de Usuários Concorrentes	grande	relativamente pequeno
Interações com os Usuários	estáticas, predefinidas	dinâmicas, exploratórias

Diferenças entre os Ambientes

	Ambiente Operacional	Ambiente Informacional
Volume das Operações	relativamente alto	relativamente baixo
Características das Operações	mais simples, acessando menos registros por vez	mais complexas, acessando muitos registros por vez

Diferenças entre os Ambientes

	Ambiente Operacional	Ambiente Informacional
Volume das Operações	relativamente alto	relativamente baixo
Características das Operações	mais simples, acessando menos registros por vez	mais complexas, acessando muitos registros por vez

Diferenças entre os Ambientes

	Ambiente Operacional	Ambiente Informacional
Projeto do Banco de Dados	normalizado	multidimensional
Granularidade dos Dados	nível de detalhe específico	diferentes níveis de detalhe
Volume de Dados	<i>megabytes a gigabytes</i>	<i>gigabytes a terabytes a petabytes</i>

Diferenças entre os Ambientes

	Ambiente Operacional	Ambiente Informacional
Projeto do Banco de Dados	normalizado	multidimensional
Granularidade dos Dados	nível de detalhe específico	diferentes níveis de detalhe
Volume de Dados	<i>megabytes a gigabytes</i>	<i>gigabytes a terabytes a petabytes</i>

Diferenças entre os Ambientes

	Ambiente Operacional	Ambiente Informacional
Exemplos de Aplicação	transações bancárias emprestimos de livros contas a pagar matrículas em cursos	planejamento de <i>marketing</i> análise financeira tomada de decisão planejamento estratégico

Agenda

- Inteligência do Negócio
- Data Warehousing
- Diferenças entre os Ambientes Operacional e Informacional

Análise de Dados com Base em Processamento Massivo em Paralelo

Lista de Exercícios: Introdução

Profa. Dra. Cristina Dutra de Aguiar

Observação:

Esta lista contém exercícios relacionados à primeira semana de aula. A resposta de cada exercício encontra-se destacada na cor azul. Recomenda-se fortemente que a lista de exercícios seja respondida antes de se consultar as respostas dos exercícios.

1. Qual a diferença entre OLTP e OLAP?

OLTP (*on-line transaction processing*) diz respeito ao ambiente operacional, voltado ao processamento de transações. Isso significa que no ambiente OLTP existem muitas operações de inserção, remoção e atualização e que o objetivo de desempenho é realizar o processamento eficiente dessas operações.

OLAP (*on-line analytical processing*) diz respeito ao ambiente informacional, voltado ao processamento de consultas analíticas. Isso significa que no ambiente OLAP existem muitas consultas e que o objetivo de desempenho é realizar o processamento eficiente dessas consultas.

2. Liste os principais aspectos pelos quais o ambiente de *data warehousing* se difere do conceito de *data warehouse*.

Data warehouse representa o banco de dados, ou seja, é o local onde os dados são armazenados. *Data warehousing*, por sua vez, representa um ambiente, o qual é composto por *data warehouse*, *software*, *hardware* e *peopleware*.

Data warehouse é um dos componentes de maior importância do *data warehousing*, consistindo no local onde os dados resultantes do processo de ETL (*extract, transform, load*) e modelados multidimensionalmente são armazenados.

3. Na aula, foram contextualizados dois ambientes: o ambiente operacional e o ambiente informacional. Esses ambientes são distintos entre si e cada um deles tem características importantes que os definem. Descreva cada um desses ambientes, destacando as suas principais características.

No ambiente operacional, os tipos de operação mais frequentes são de inserção, remoção e atualização dos dados, o que é uma característica do OLTP (*on-line transaction processing*). As interações com os usuários são usualmente estáticas e predefinidas. As aplicações do ambiente operacional vislumbram o processamento eficiente das operações de inserção, remoção e atualização dos dados. Nesse ambiente, a quantidade de usuários que usam cada aplicação simultaneamente é muito grande.

No ambiente informacional, o tipo de operação mais frequente é a consulta aos dados, ou seja, a leitura dos dados, o que é uma característica do OLAP (*on-line analytical processing*). As interações realizadas com os usuários são usualmente dinâmicas, desde que os usuários podem consultar os dados de acordo com diferentes perspectivas de análise. As aplicações do ambiente informacional vislumbram o processamento eficiente das consultas. Essas consultas são caracterizadas por acessar inúmeros registros, desde que usualmente realizam análises massivas dos dados. Poucos usuários interagem com o ambiente informacional simultaneamente. Esses usuários geralmente são executivos, analistas e gerentes, ou seja, usuários voltados à tomada de decisão estratégica.

4. O reitor de uma universidade precisa tomar uma decisão acerca da distribuição de recursos financeiros entre os institutos da universidade. Em reunião com os pró-reitores da instituição, foi definido que aqueles institutos com maior quantidade de publicações científicas em revistas de alto fator de impacto devem ser priorizados. Sendo assim, o reitor decidiu utilizar a ferramenta OLAP da universidade para emitir um relatório contendo a quantidade de publicações por instituto, por revista e por trimestre. O relatório emitido pelo reitor pode ser classificado em qual categoria: dado, informação ou conhecimento? Justifique sua resposta detalhando o porquê do relatório não ter sido classificado nas outras duas categorias.

O relatório emitido pelo reitor representa uma informação, uma vez que os dados presentes neste relatório mostram a quantidade (ou seja número) de publicações agrupados por diferentes perspectivas (por instituto, por revista e por trimestre). O relatório não pode ser classificado como um dado bruto, uma vez que houve um processamento para sua geração. Além disso, o relatório também não pode ser classificado como conhecimento, visto que ainda não foi devidamente interpretado pelo reitor e ainda não foi obtido nenhum direcionamento, conclusão ou tomada de decisão a partir do mesmo.



5. Considere uma empresa de supermercados que possui várias filiais. Cada filial possui um sistema diferente para contabilizar os produtos vendidos e as promoções realizadas. Um executivo dessa empresa deseja fazer uma análise para descobrir filiais que precisam ser fechadas ou remodeladas. Por que não é ideal que esse executivo realize essa análise sobre os sistemas existentes da empresa?

Porque as análises propostas e o dados são consideravelmente complexos. Mesmo sendo possível usar as aplicações de bancos de dados existentes, existem diversos desafios a serem enfrentados. Esses desafios são muitas vezes extremamente custosos e, portanto, proibitivos para a produção da informação certa, na hora certa, para a pessoa certa.

Alguns desafios que podem ser ressaltados dentro do contexto exemplificado são:

- O dados de interesse estão espalhados em várias filiais. Consequentemente, esses dados devem ser obtidos de diferentes fontes de dados que normalmente assumem diferentes formatos e requerem processos de limpeza e tradução acurados.
- A complexidade das consultas impacta no desempenho das mesmas. Na descrição dada, o objetivo é realizar análises para descobrir filiais que precisam ser fechadas ou remodeladas. Isso, muito provavelmente, envolveria a obtenção de inúmeros indicadores e informações a partir dos dados.
- O tratamento dos dados temporais usualmente é incipiente, não existindo registro temporal para todas as análises possíveis de serem realizadas.

6. Considere a seguinte situação:

Uma empresa brasileira especializada em análise de dados busca estudar e entender as maiores consequências que a pandemia gerada pelo novo coronavírus ocasionou nas cidades brasileiras. O principal objetivo é determinar quais diferentes medidas podem ser tomadas, levando em consideração as diversas características socioeconômicas que cada cidade ou região do país pode ter. Para dar início ao estudo, a empresa catalogou conjuntos de dados que contêm índices socioeconômicos e dados referentes aos índices de contaminação, quantidade de testes, recuperação e óbitos para cada cidade.

Com base nessa descrição, cite quais dados podem ser extraídos, quais informações podem ser obtidas e quais conhecimentos podem ser construídos.

O dados que podem ser extraídos estão vinculados aos conjuntos de dados coletados, em sua forma bruta e sem significado semântico. Eles são: índices socioeconômicos, índices de contaminação, quantidade de testes, recuperação e óbitos para cada cidade ou região, datas referente às coletas realizadas, entre outros.

As informações que podem ser extraídas surgem a partir da organização dos dados, estruturação e contextualização dos mesmos. Exemplos de informações que podem ser extraídas são: regiões com maior taxa de contaminação, regiões que mais realizam a testagem de pessoas, regiões com maior taxa de óbitos, curva de contaminação ou recuperação de cada região, dentre outras.

O conhecimento provém das informações interpretadas, analisadas e processadas. Exemplos de conhecimento que pode ser extraído por meio das informações citadas supracitadas são:

- Regiões que têm características socioeconômicas parecidas, porém com curvas de contaminação diferentes, podem adotar medidas de combate à pandemia diferentes. Isto pode ser utilizado para se comparar a eficácia das diferentes medidas tomadas, por exemplo.
- Agregação das informações de índices socioeconômicos com curvas de contaminação, recuperação e óbito, possibilitando a detecção de padrões que facilitam a tomada de decisão estratégica. Essas informações agregadas podem considerar determinadas regiões com características peculiares, por exemplo.

7. Pesquise uma situação real na qual seria interessante aplicar consultas analíticas. Descreva o problema e elabore três perguntas para exemplificar possíveis análises. A partir dessas perguntas, descreva três exemplos de conhecimento que podem ser gerados por meio dessas análises.

Questão livre para discussão durante as tutorias. Não existe apenas uma resposta certa.



Análise de Dados com Base em Processamento Massivo em Paralelo

Aula 2: Arquitetura de Data Warehousing

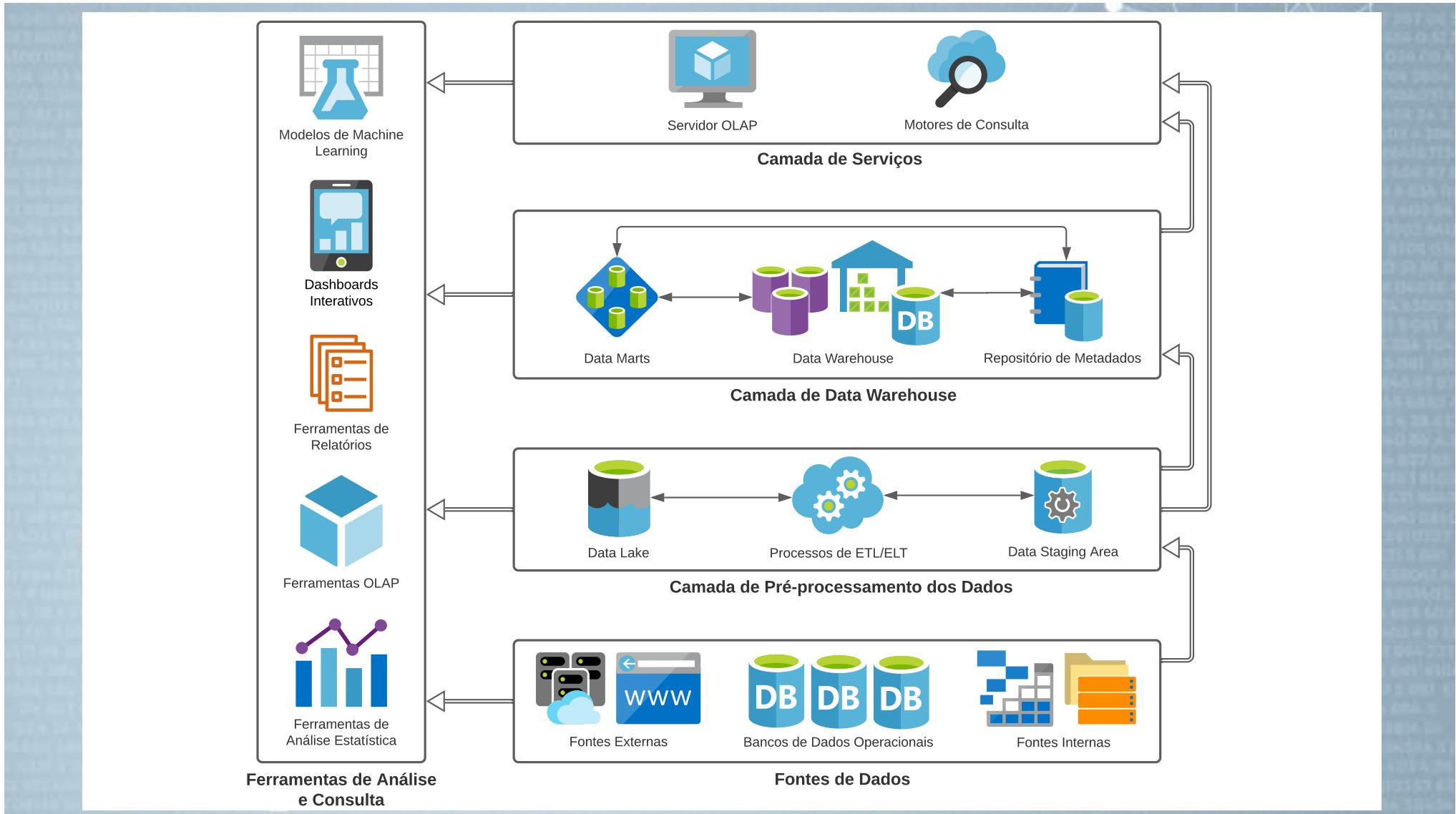
Cristina Dutra de Aguiar
ICMC/USP
cdac@icmc.usp.br

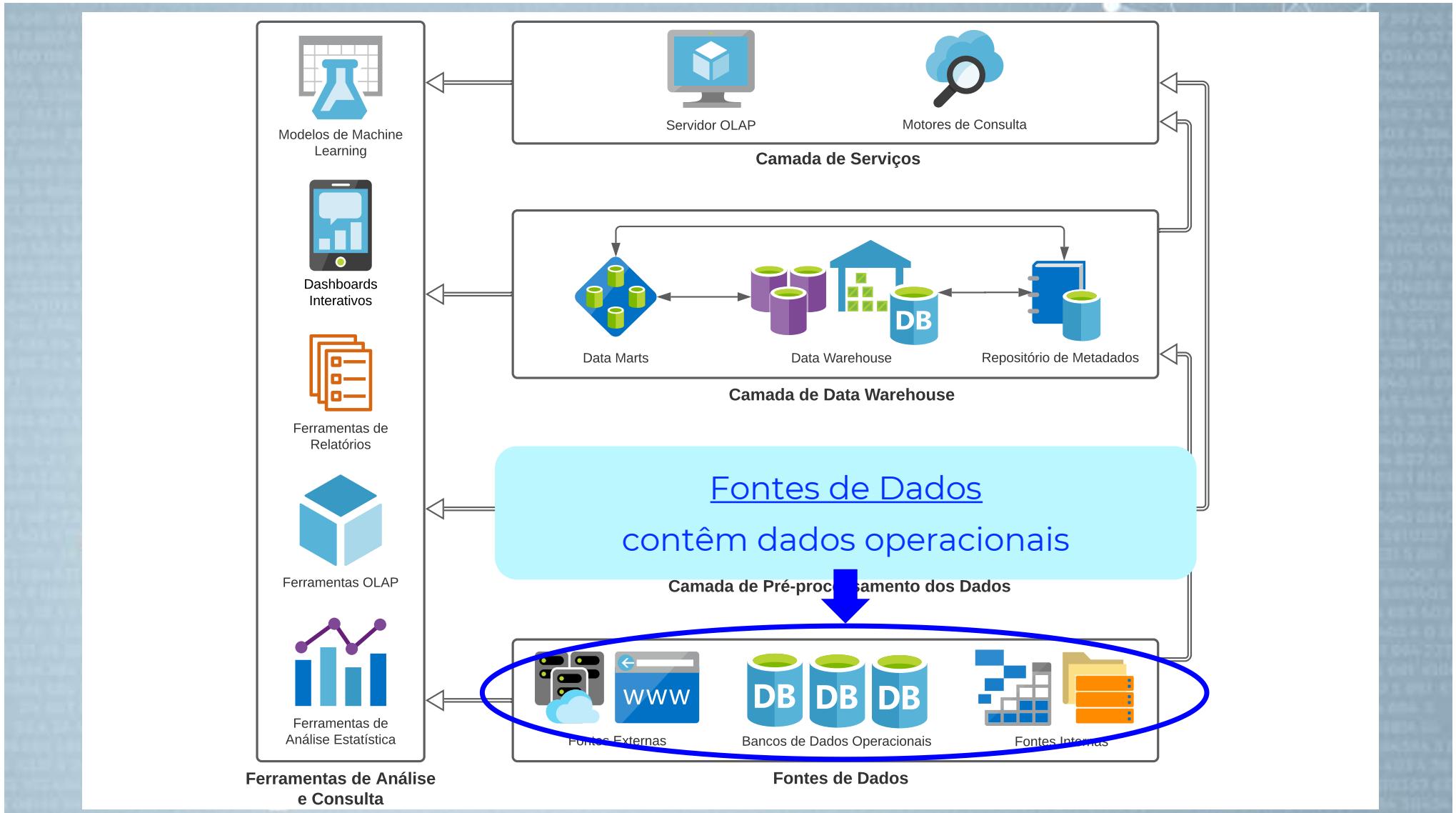


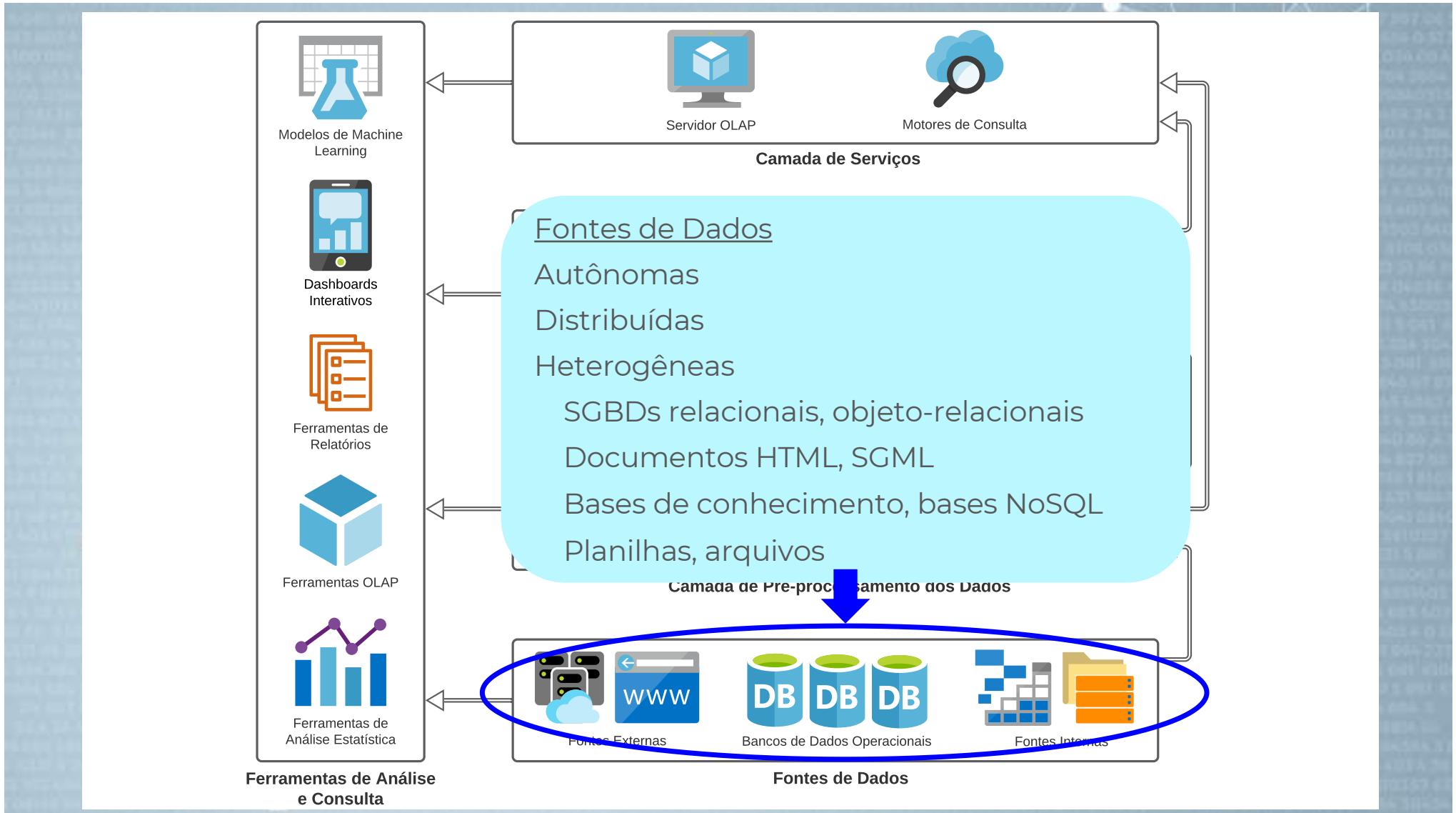
CEPID - Centro de Ciências
Matemáticas Aplicadas à Indústria

Agenda

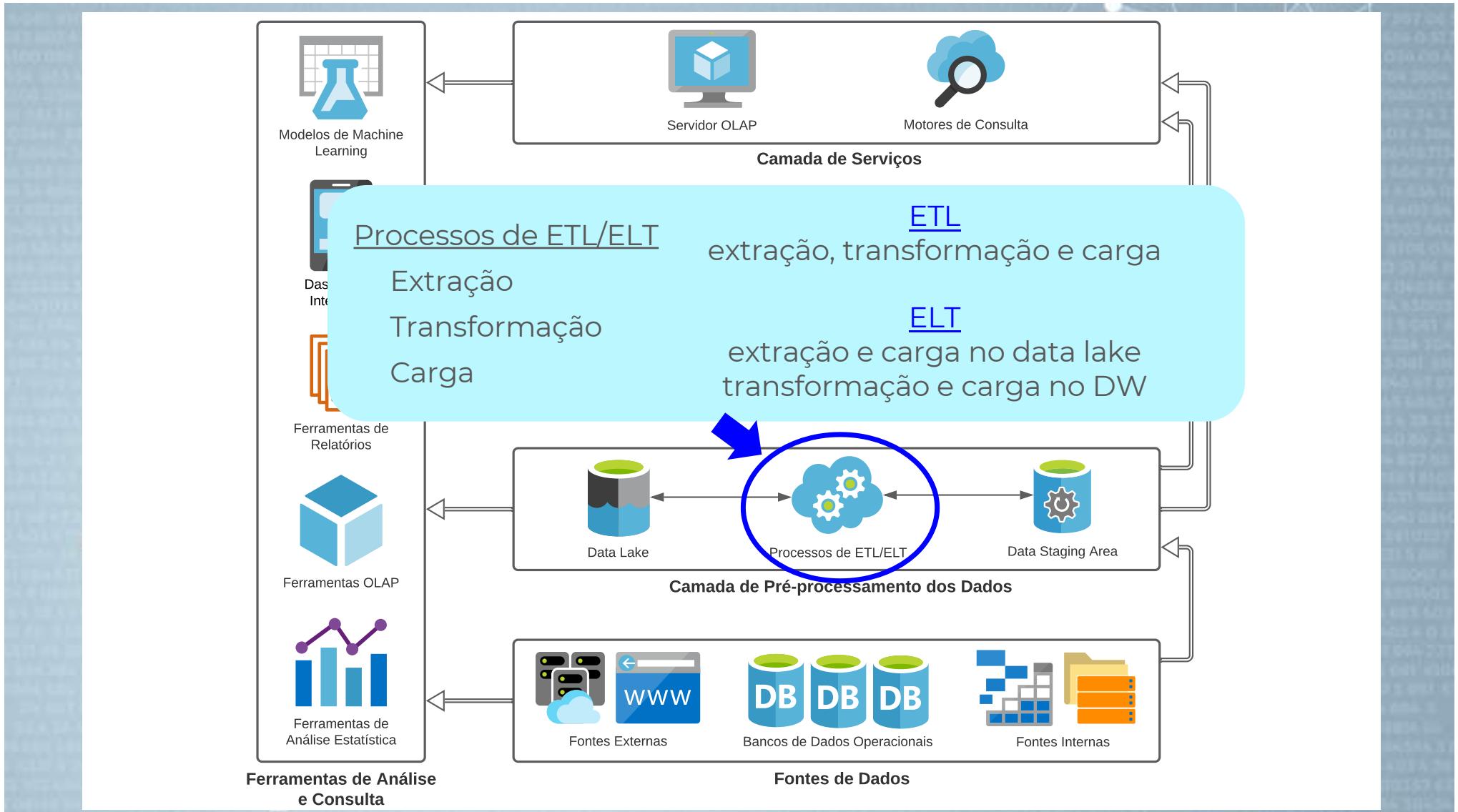
- Visão Geral
- Diferenças entre os Locais de Armazenamento
- Big Data
- Exemplos de Pipeline



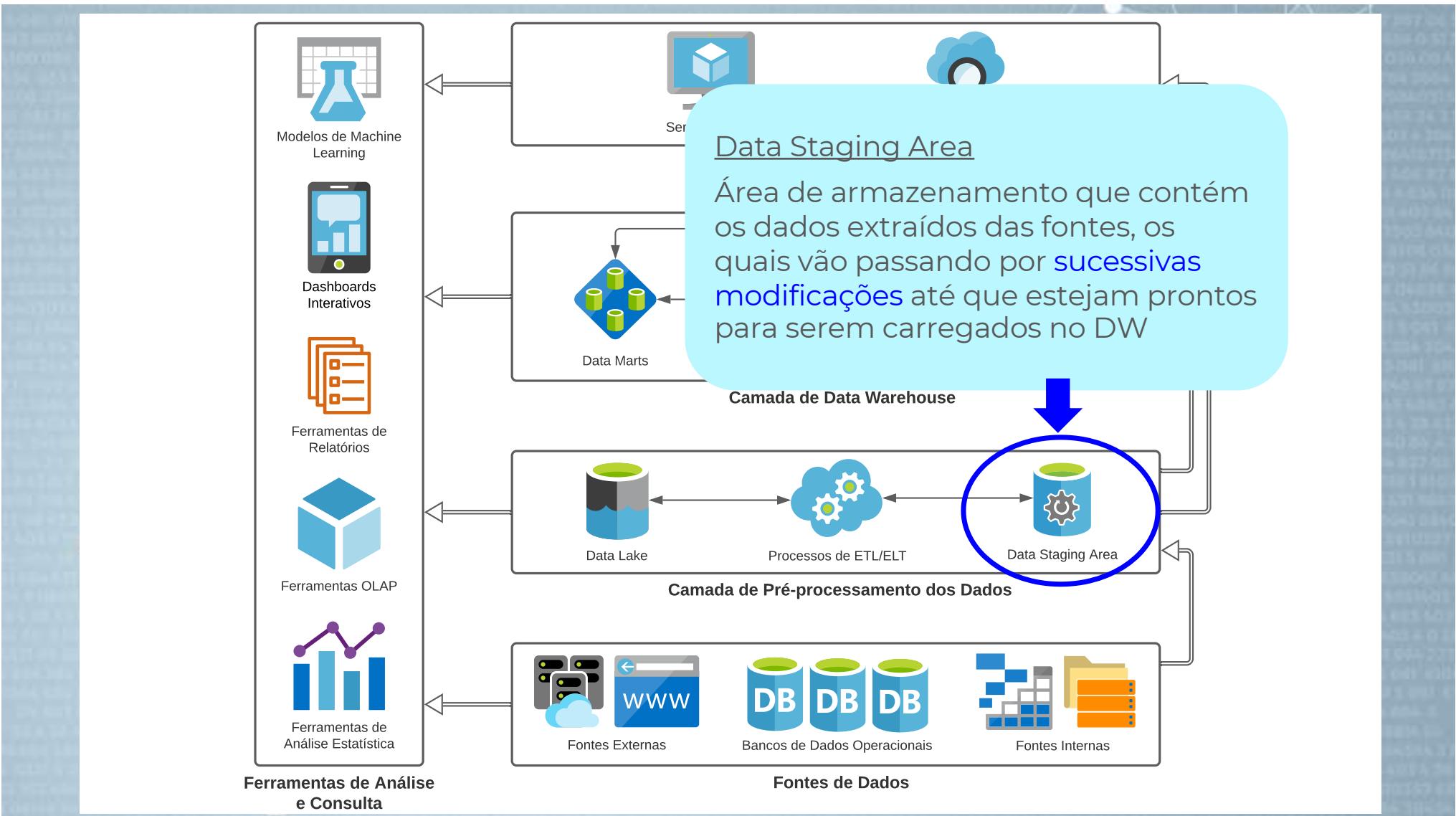






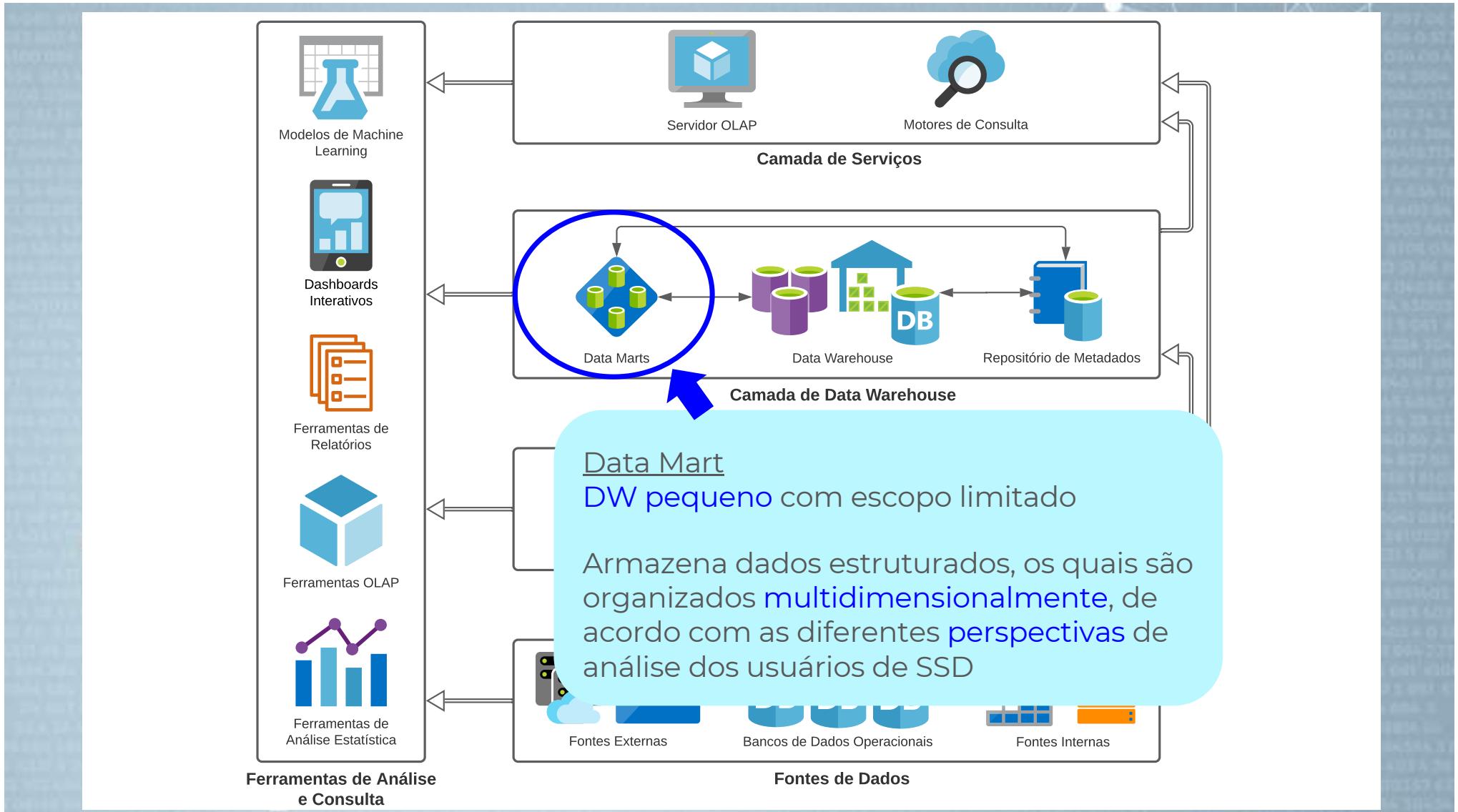




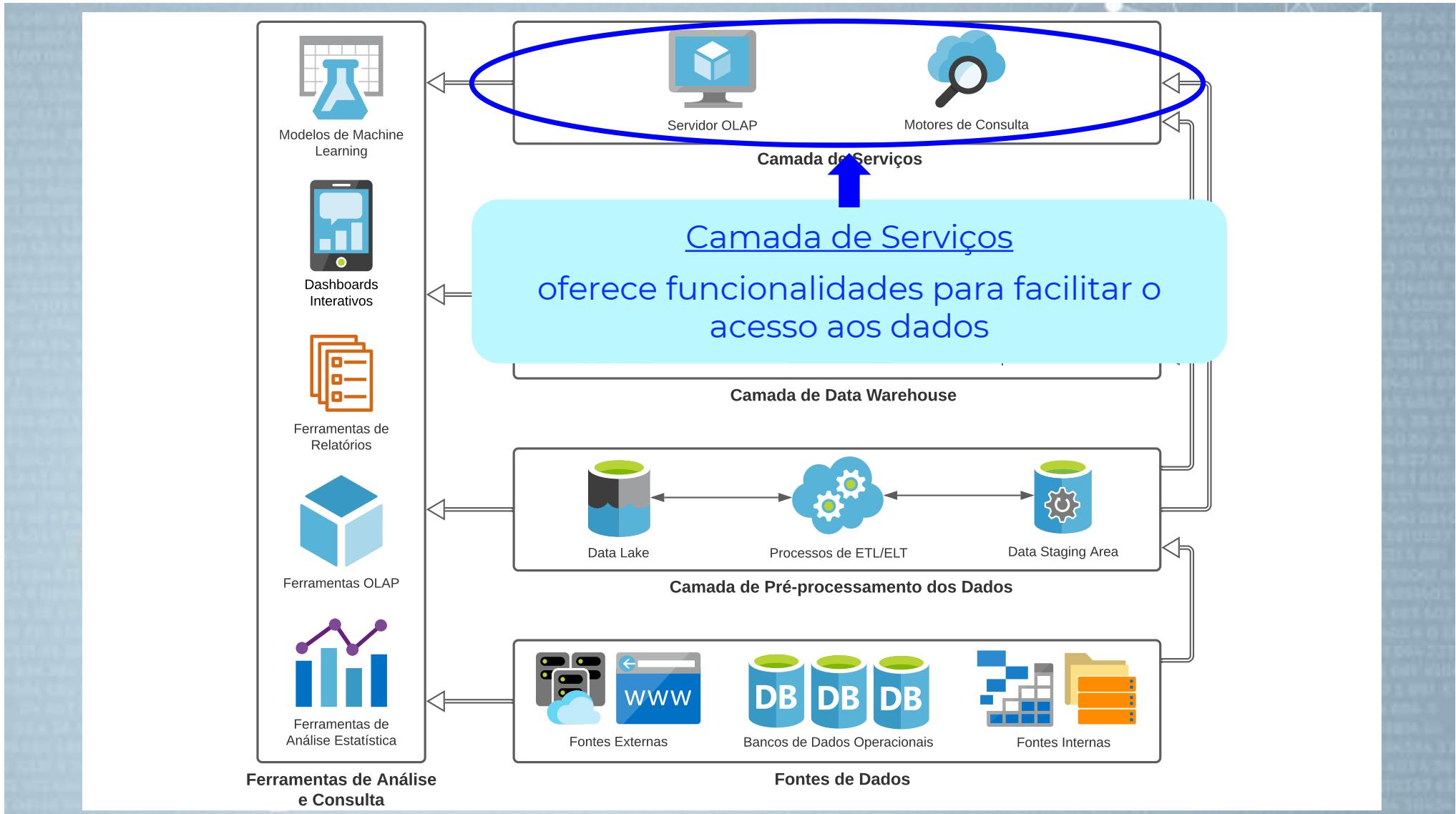


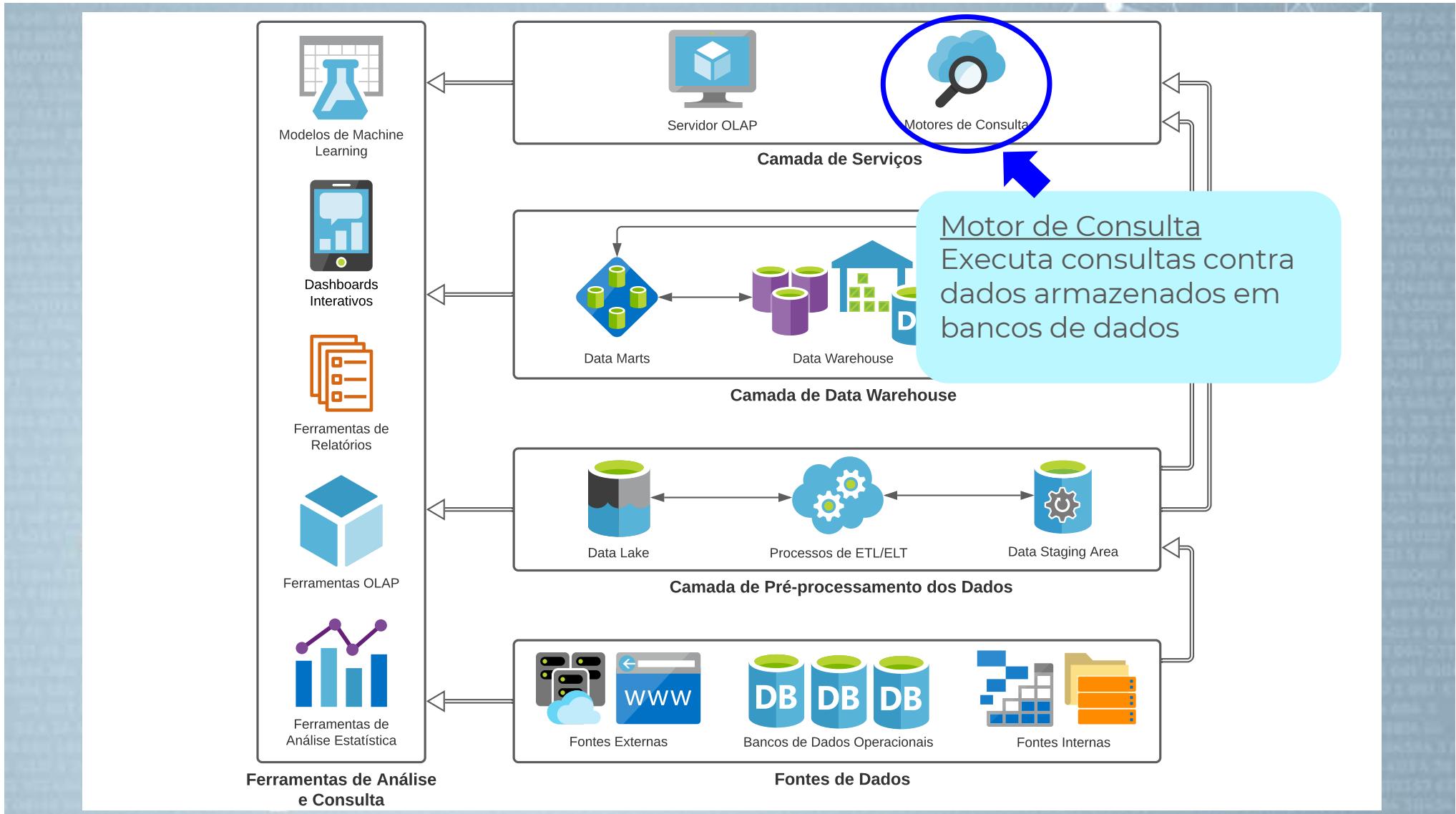


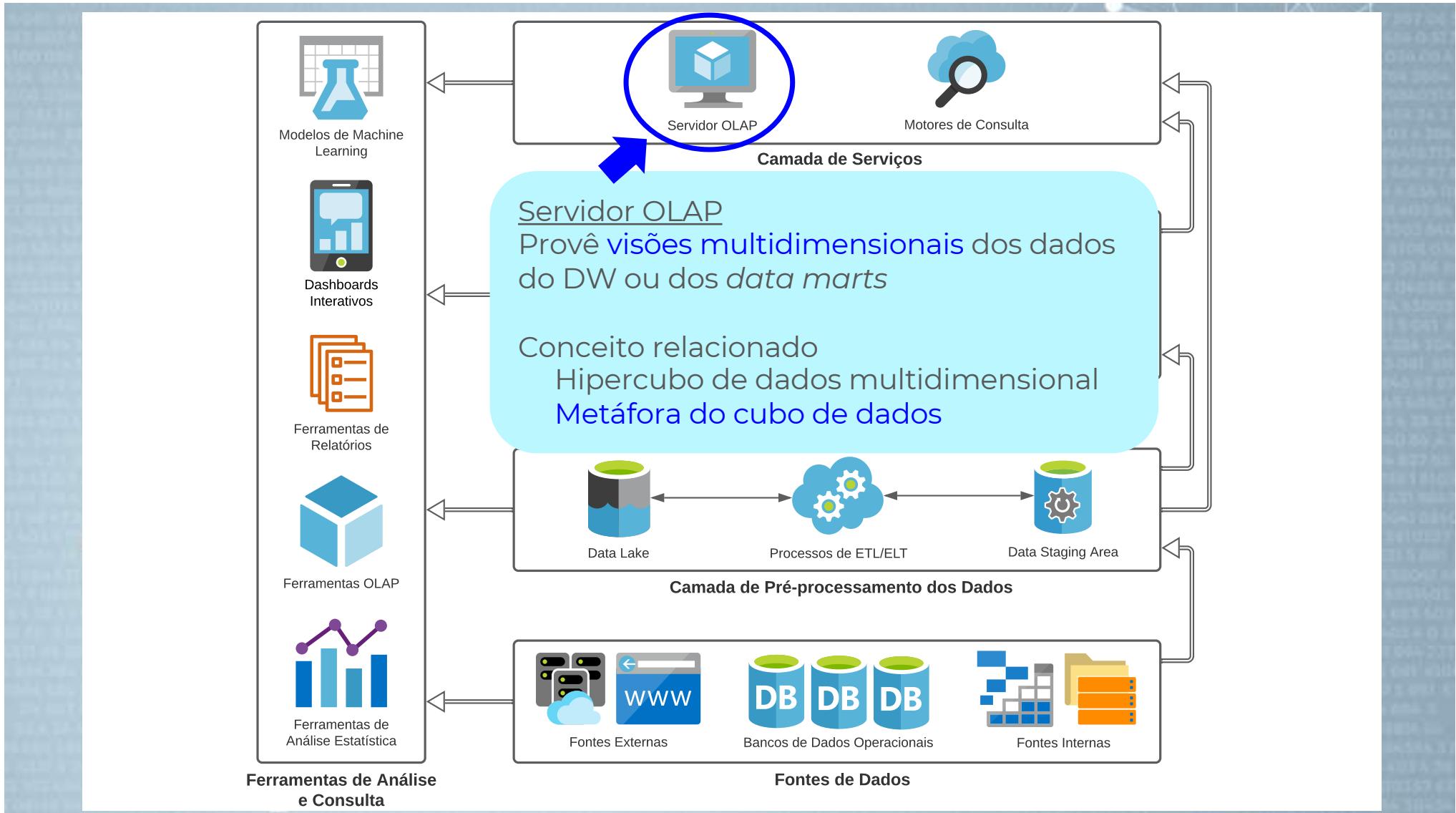


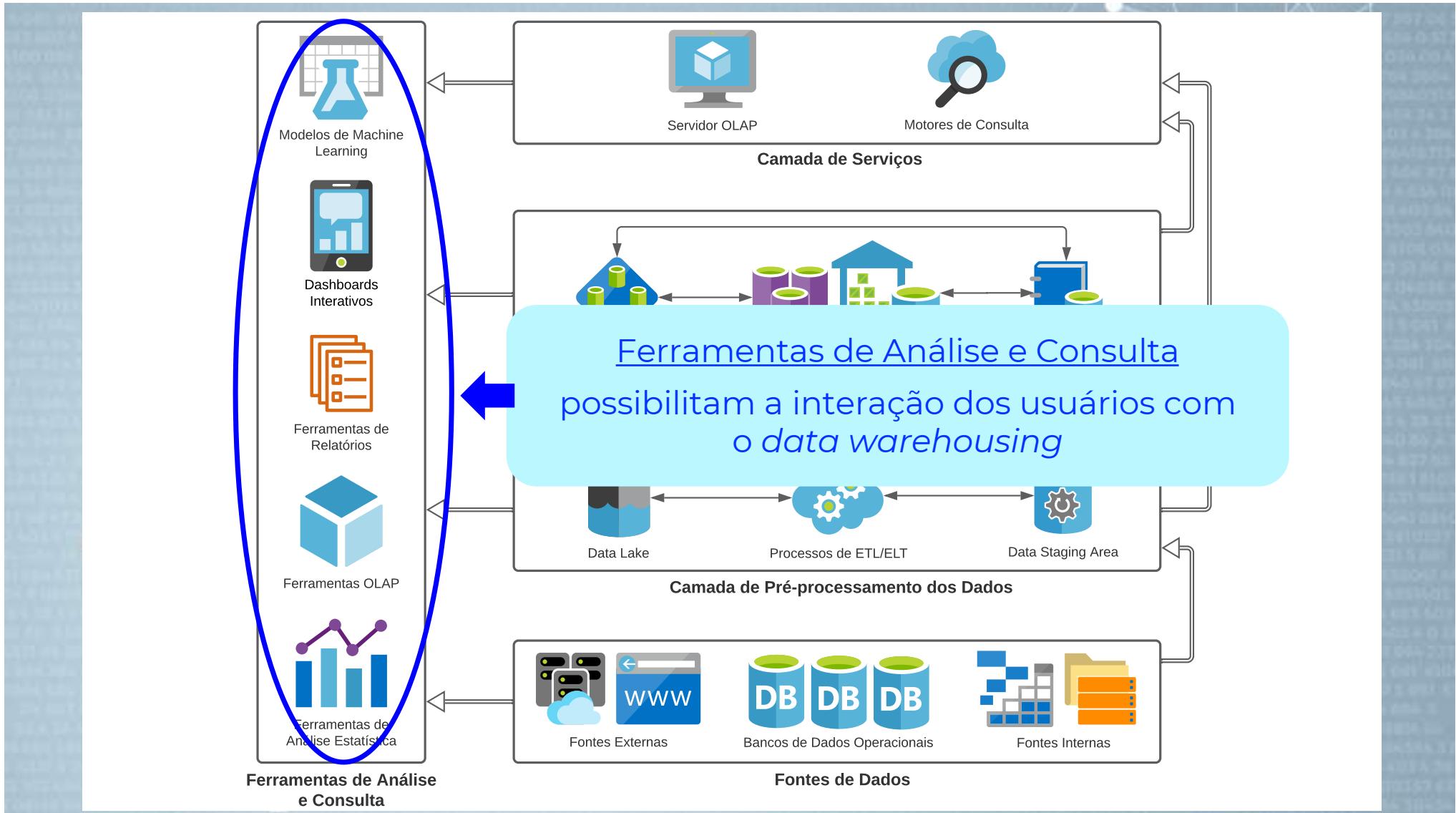


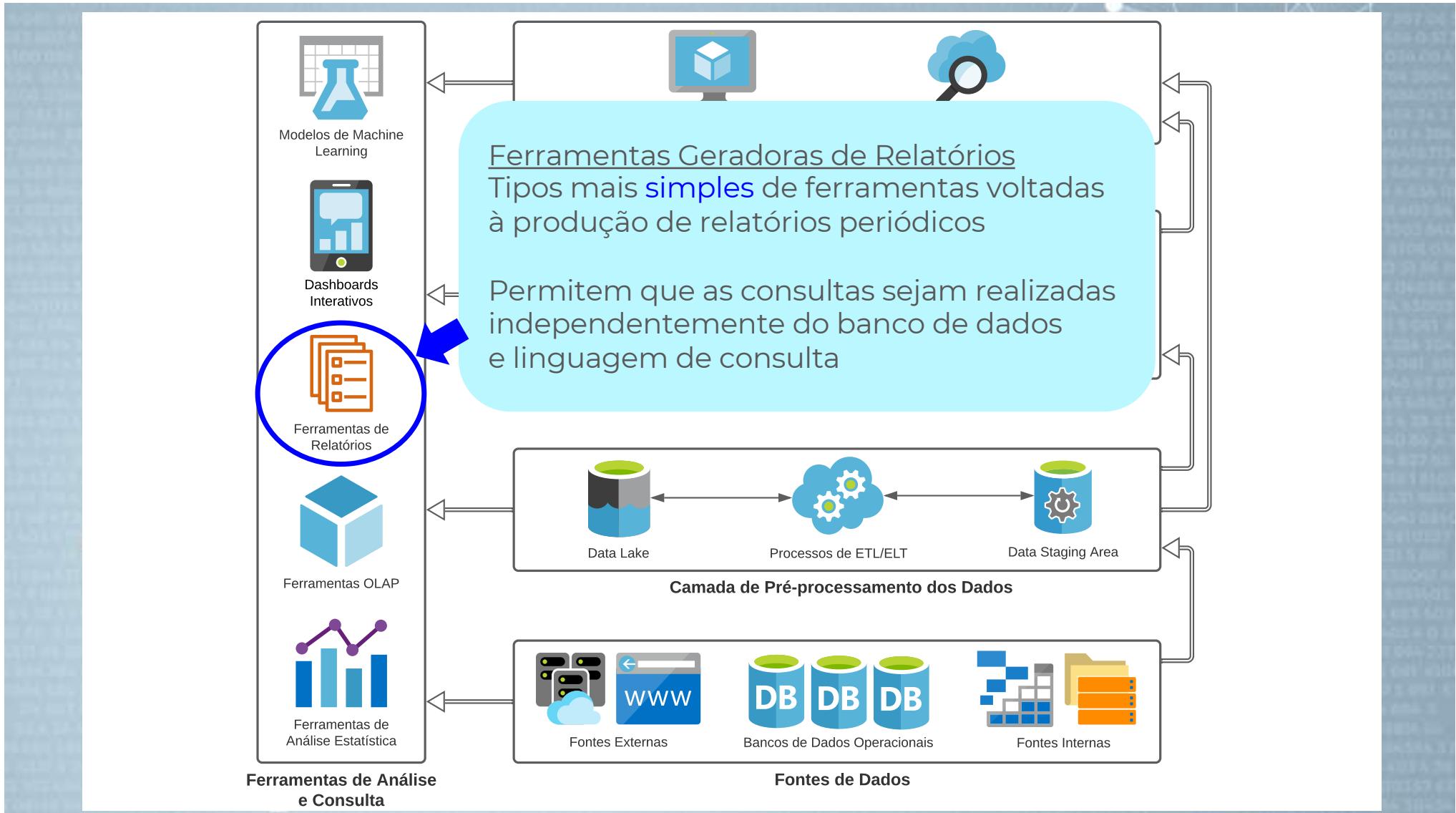




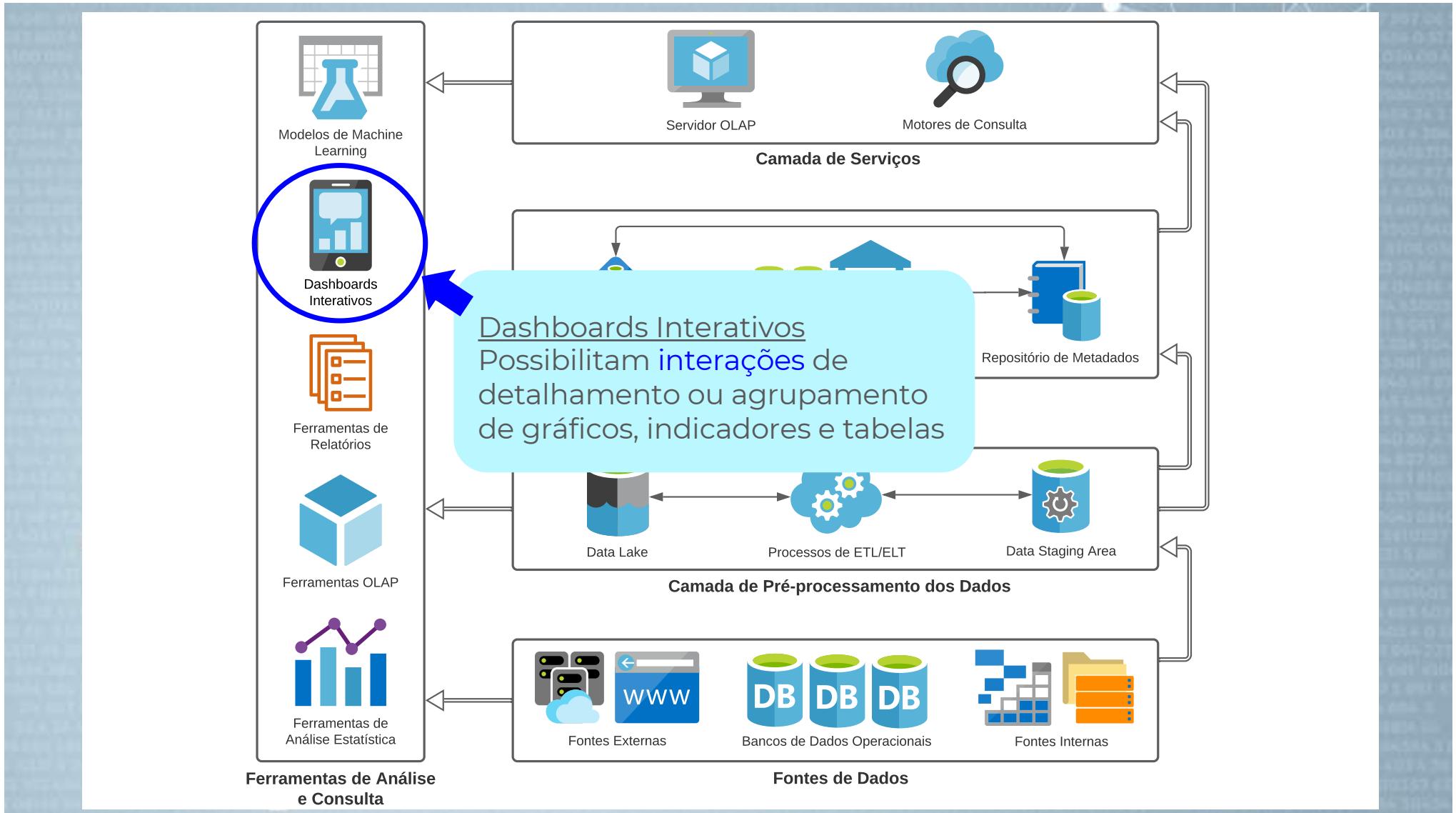


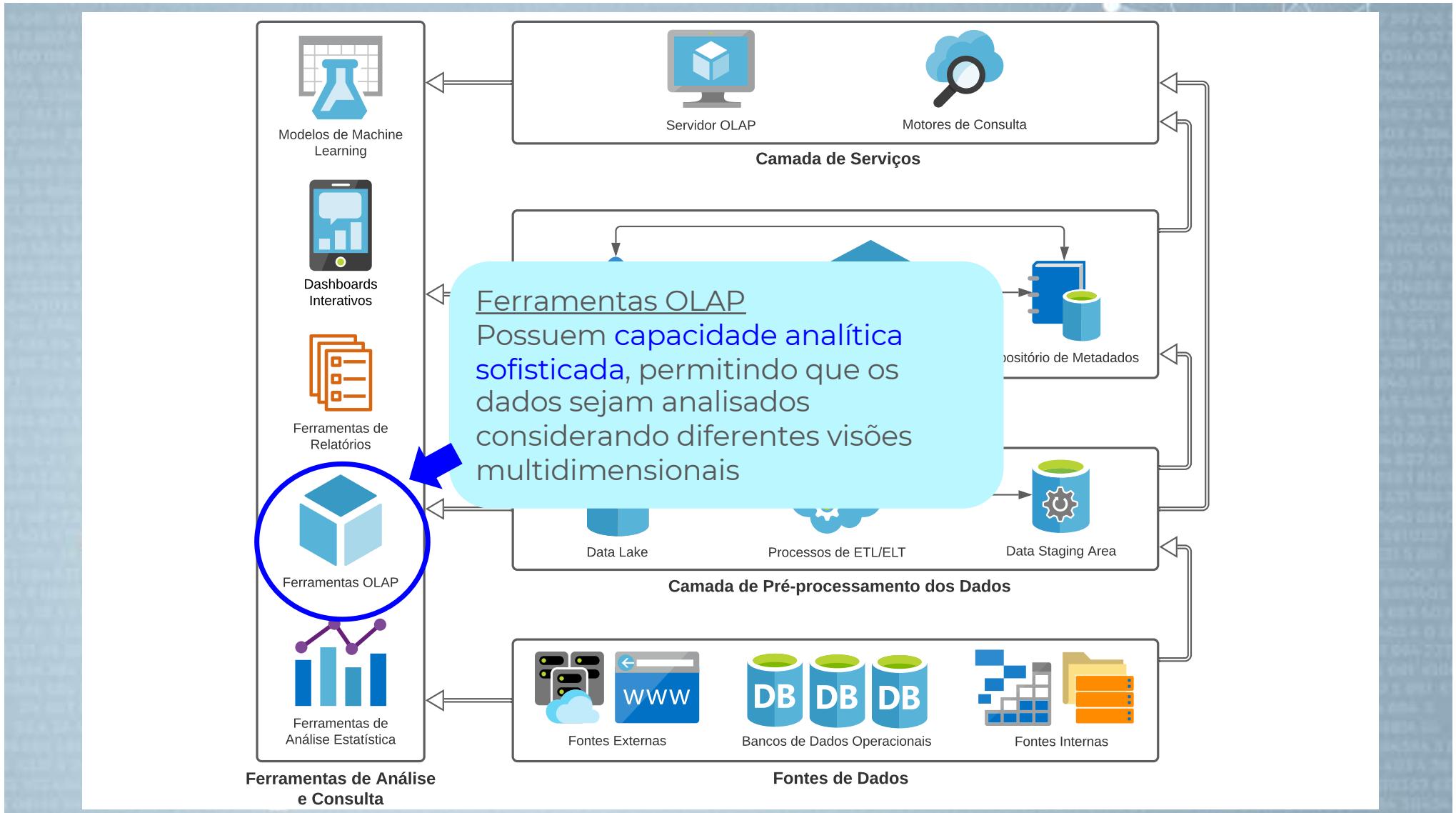


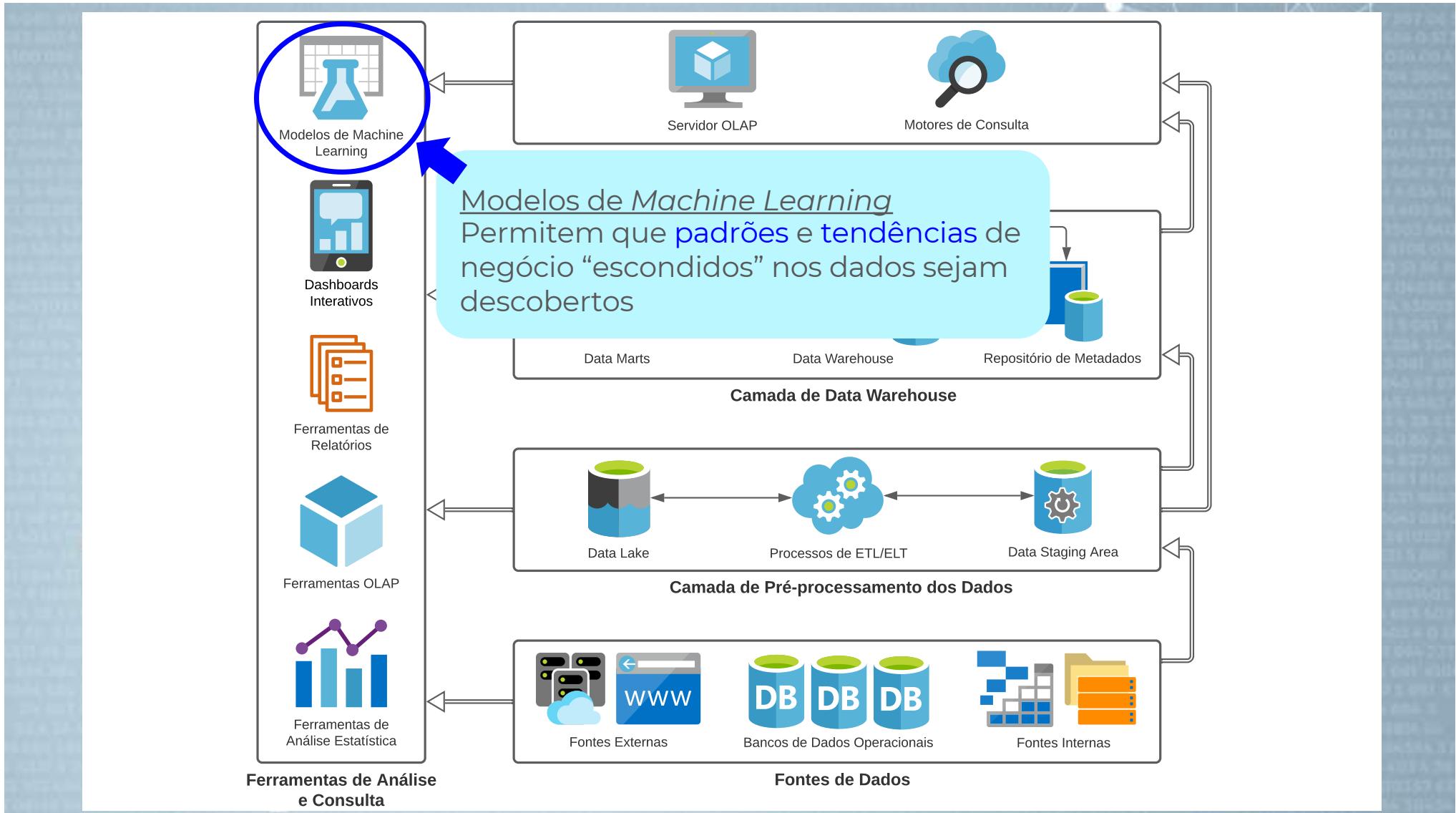


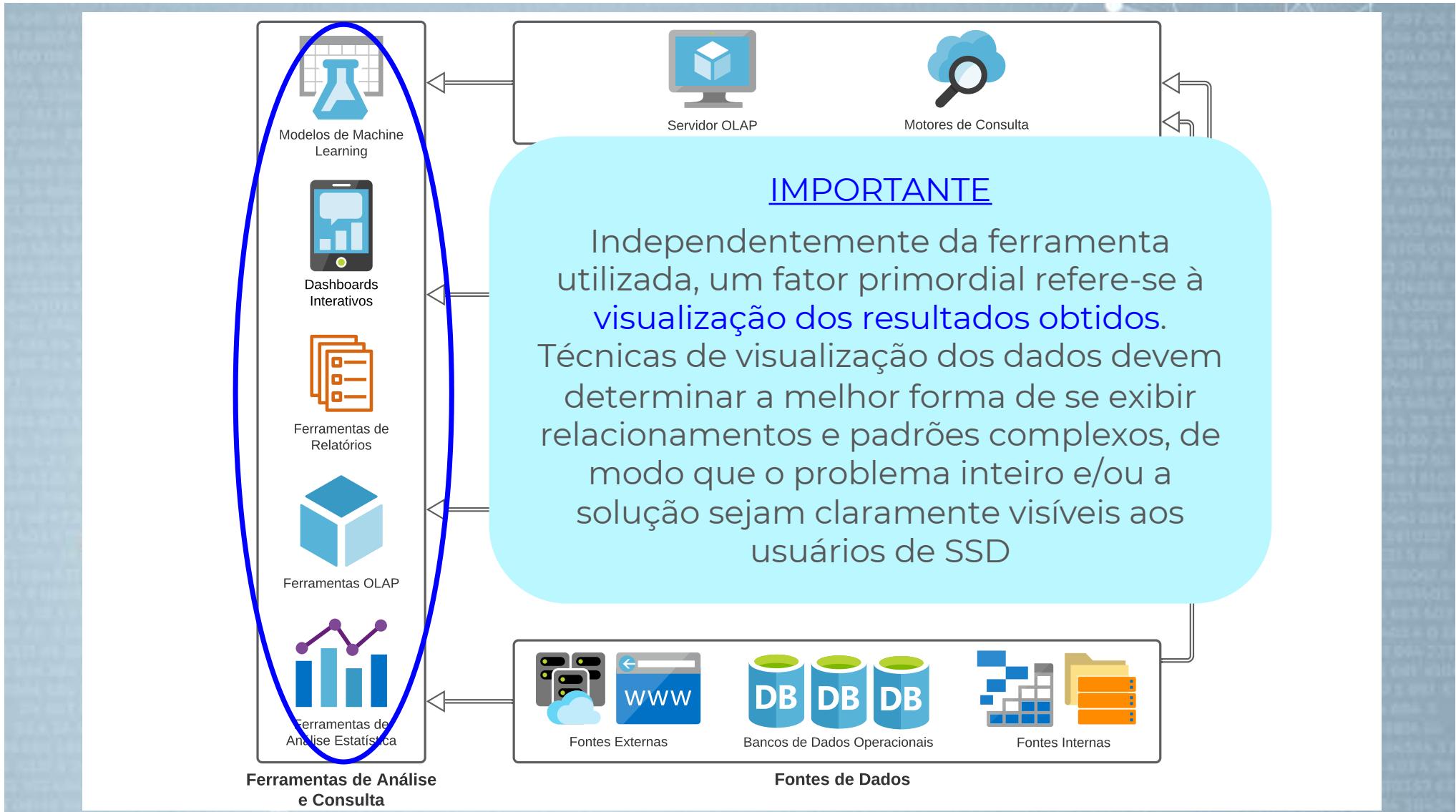






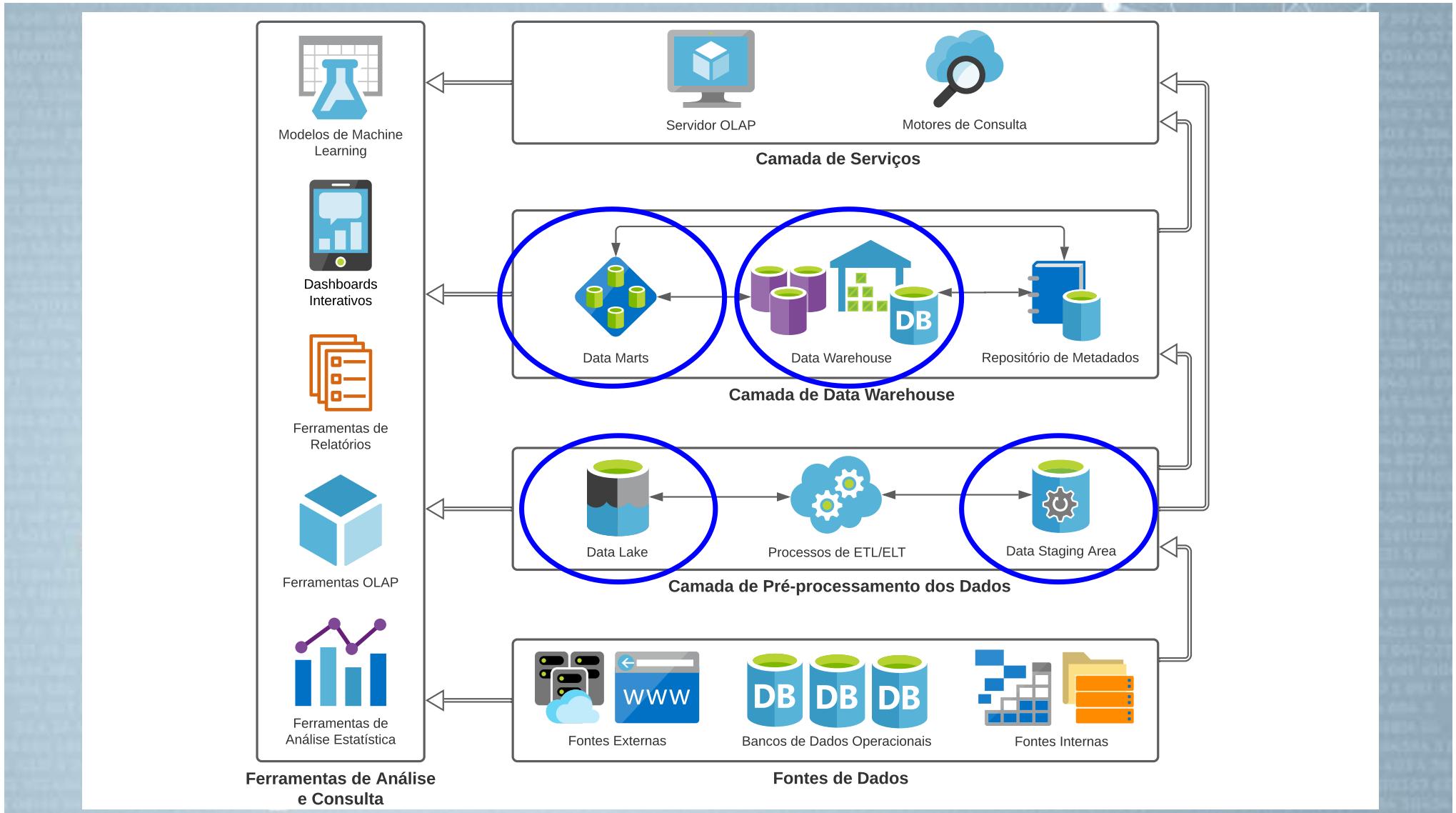


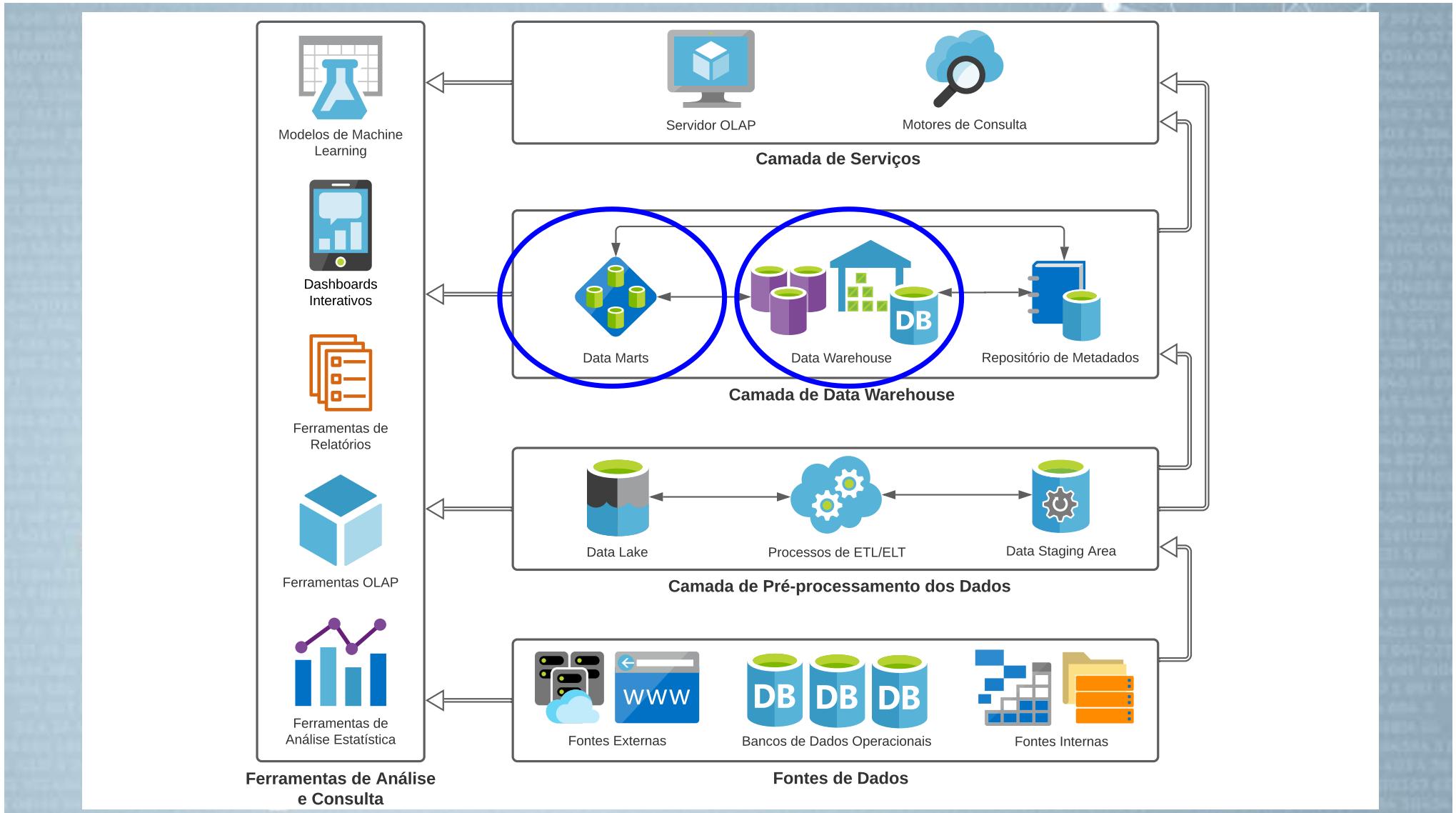




Agenda

- Visão Geral
- Diferenças entre os Locais de Armazenamento
- Big Data
- Exemplos de Pipeline





Diferenças entre Data Warehouse e Data Mart

- Data Mart
 - Consiste de um DW com **escopo limitado**
 - Armazena dados que possuem as **mesmas características** dos dados do DW
- Política de construção evolucionária de um DW corporativo
 - Processo de construção de um DW corporativo é longo, complexo e demanda alto investimento financeiro
 - Construção paulatina de vários ***data marts independentes***, cada um atendendo um assunto de interesse específico

Exemplo: BI Solutions

Demanda: analisar gastos com salários de funcionários

Data Mart 1

Foco: **salários** e
quantidadeLançamentos

Perspectivas: funcionário
cargo
filial
data

Demanda: analisar gastos com material de consumo

Data Mart 2

Foco: **gastosConsumo**
Perspectivas: material
filial
data

Demanda: analisar gastos com equipamento de infraestrutura

Data Mart 3

Foco: **gastosInfra**
Perspectivas: equipamento
filial
data
fornecedor

Exemplo: BI Solutions



Demanda: analisar receitas relativas aos cursos de treinamento dos produtos vendidos

Demanda: analisar receitas relativas às vendas dos produtos da empresa

Data Mart 4

Foco: **receitaVendas**

Perspectivas: produto
cliente
filial
data

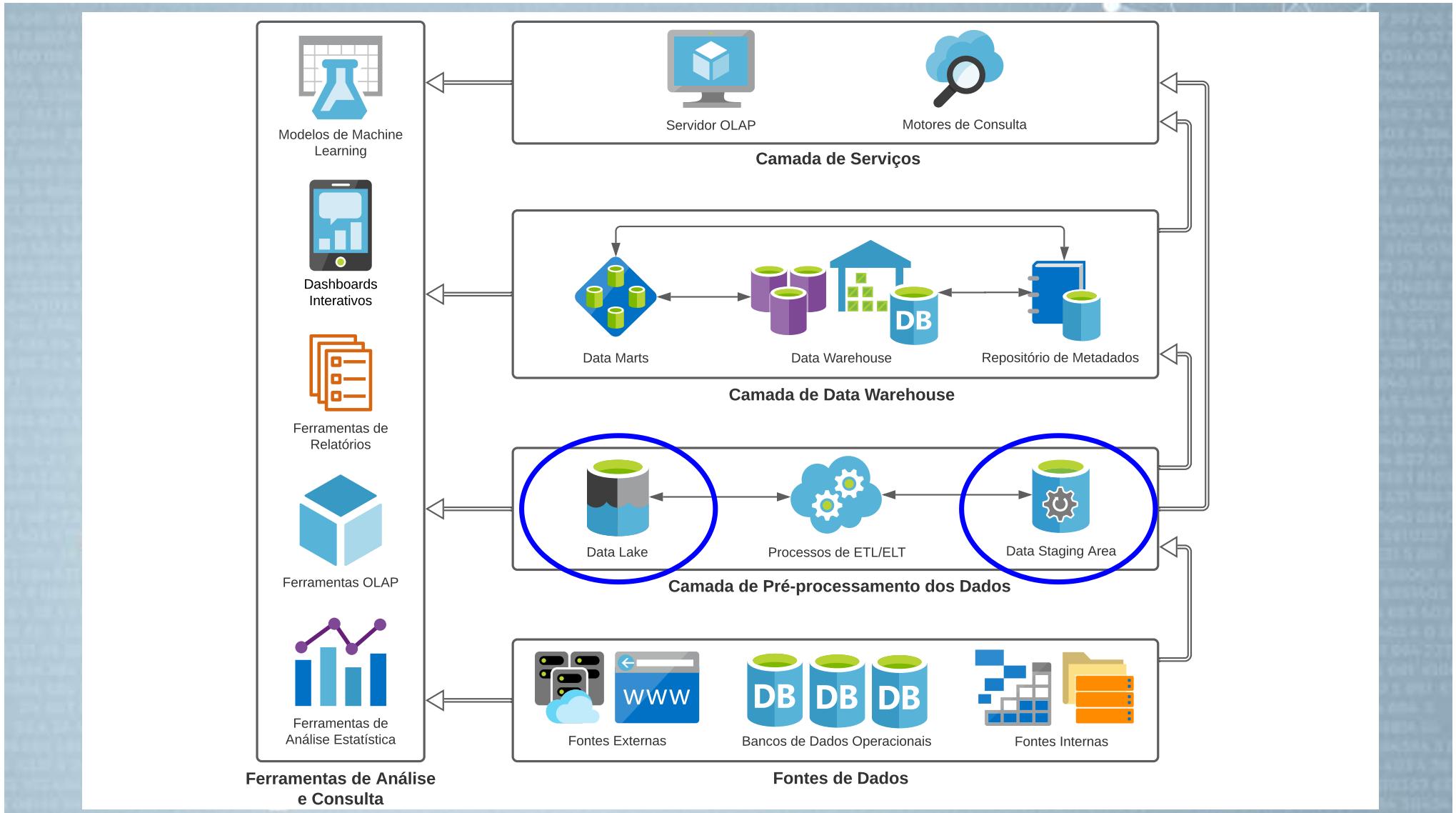
Data Mart 5

Foco: **receitaTreinamento**

Perspectivas: funcionário
cargo
data
cliente
produto

Uso de Data Marts Independentes

- Vantagens
 - Reduz gastos financeiros iniciais, desde que exige recursos monetários inferiores aos despendidos com a construção de um DW corporativo
 - Possibilita que usuários de SSD reconheçam o valor e a potencialidade da solução de *data warehousing* em um período menor de tempo
- Desvantagens
 - Pode conduzir a diferentes problemas caso um modelo de negócio completo não seja bem especificado e desenvolvido de acordo
 - Cada *data mart* independente pode tornar-se autônomo, heterogêneo e distribuído

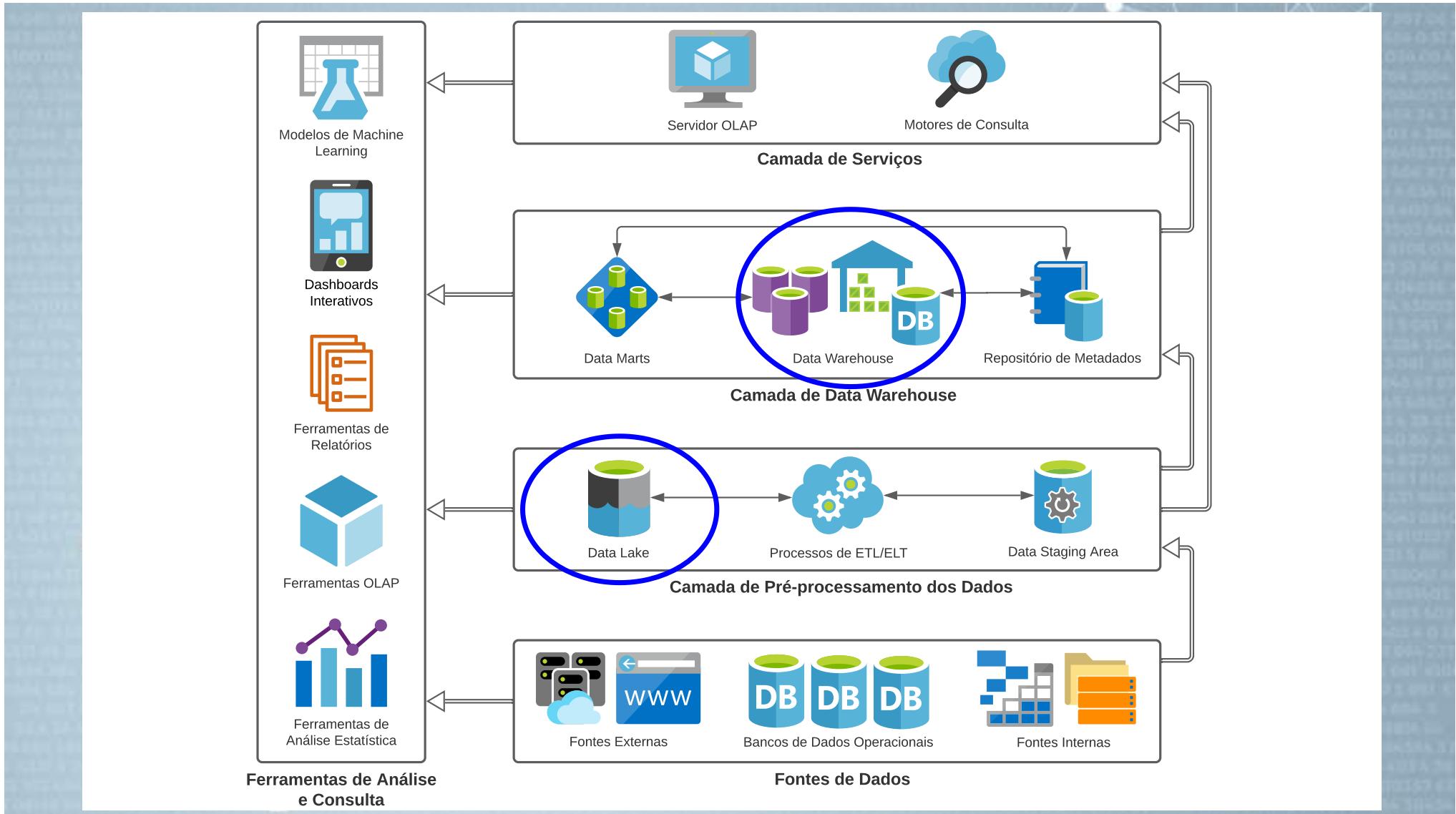


Diferenças entre Data Staging Area e Data Lake

	Data Staging Area	Data Lake
Dados Armazenados	dados que sofrem modificações sucessivas	dados no formato nativo (<i>raw data</i>), incluindo dados estruturados, semiestruturados e não estruturados
Processamento dos Dados	dados prontos para serem carregados no DW	dados processados somente quando a informação precisa ser obtida

Diferenças entre Data Staging Area e Data Lake

	Data Staging Area	Data Lake
Fluxo de Dados	<i>data staging area → DW</i>	<i>data lake → DW</i> <i>data lake → ferramentas de análise e consulta</i>
Decorrencia Histórica	processo ETL	processo ELT



Diferenças entre Data Warehouse e Data Lake

	Data Warehouse	Data Lake
Característica dos Dados	consolidados, organizados e estruturados	estruturados, semiestruturados e não estruturados
Formato dos Dados	esquema estruturado (formato bem definido)	formato nativo (diferentes formatos)
ETL/ELT	dados pré-processados antes de serem carregados	dados extraídos e carregados, sem sofrer transformações

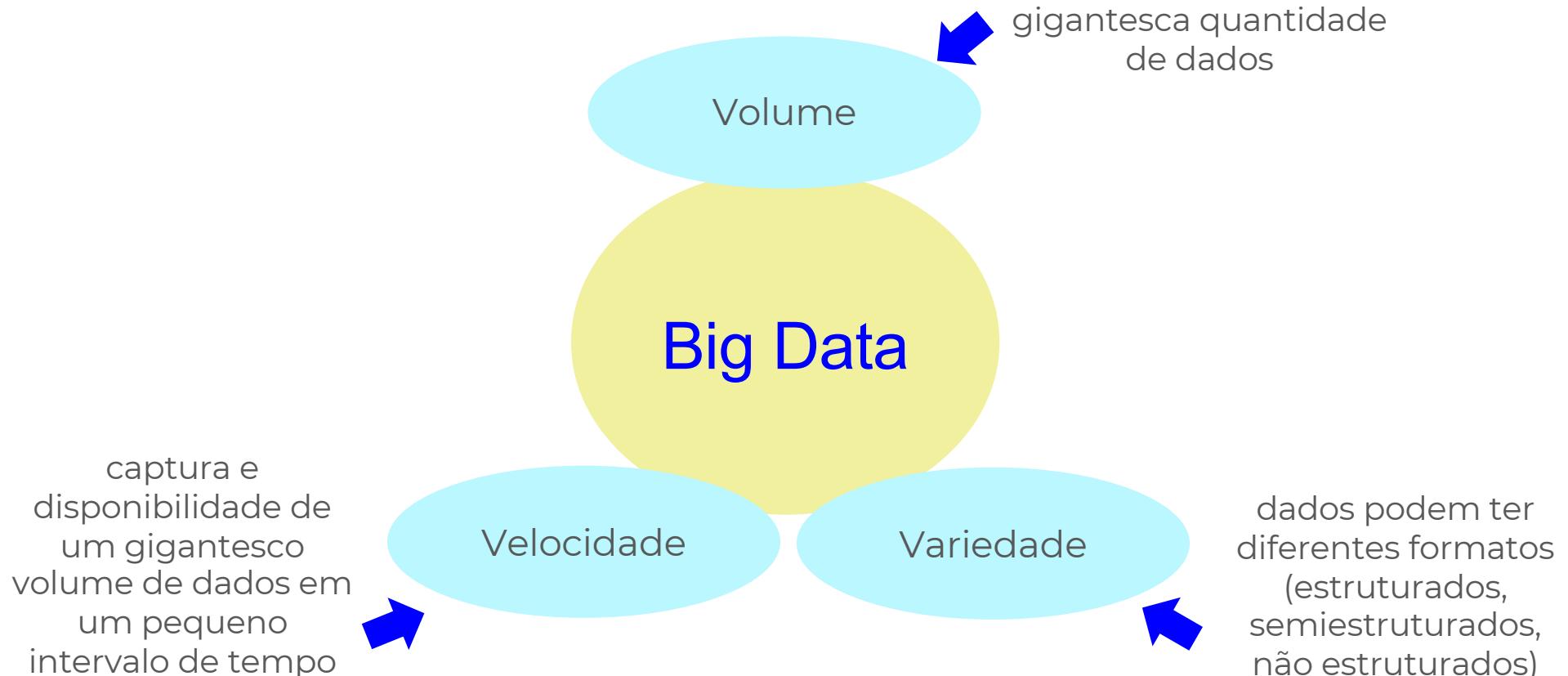
Diferenças entre Data Warehouse e Data Lake

	Data Warehouse	Data Lake
Tipos de Consulta	OLAP	variado
Latência para Disponibilizar os Dados	alta	baixa
Custo de Geração dos Dados	maior	menor
Custo de Análise dos Dados	menor	maior

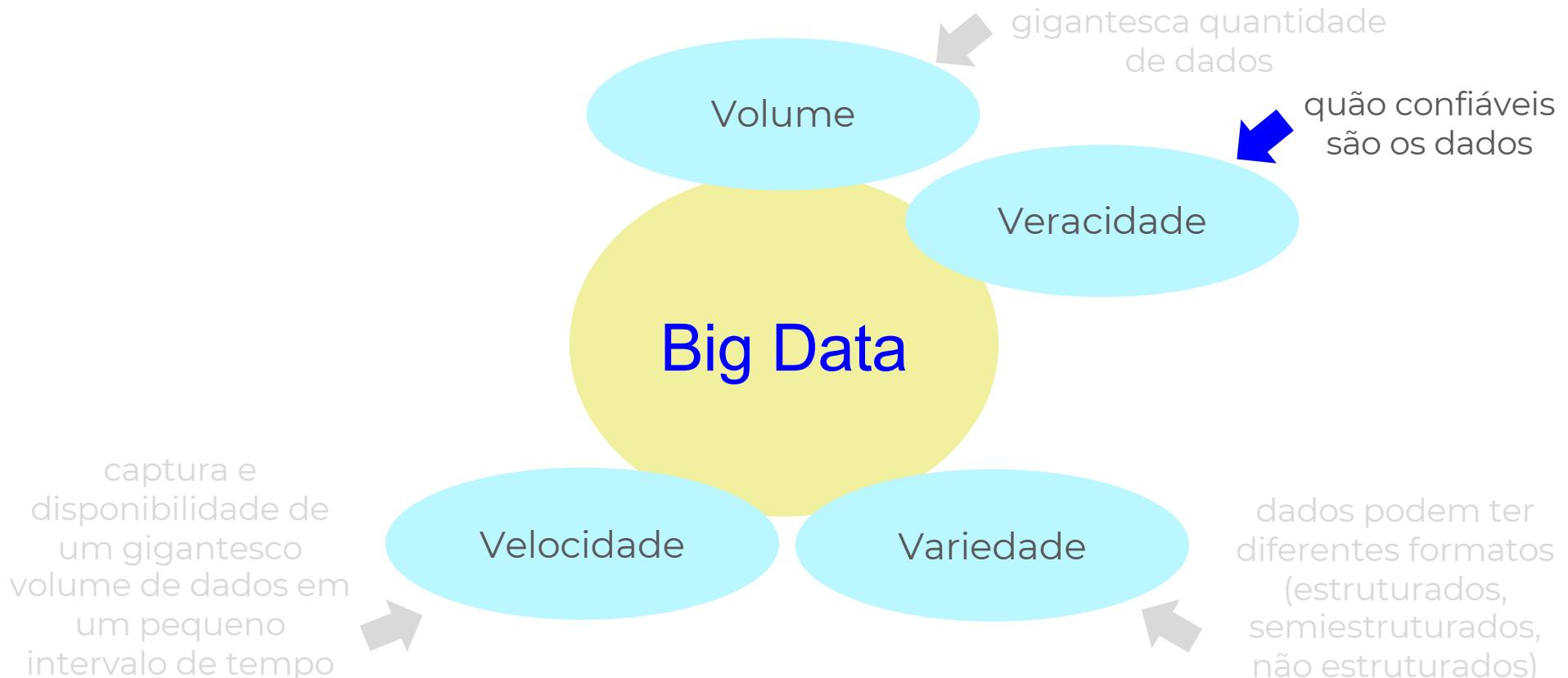
Agenda

- Visão Geral
- Diferenças entre os Locais de Armazenamento
- Big Data
- Exemplos de Pipeline

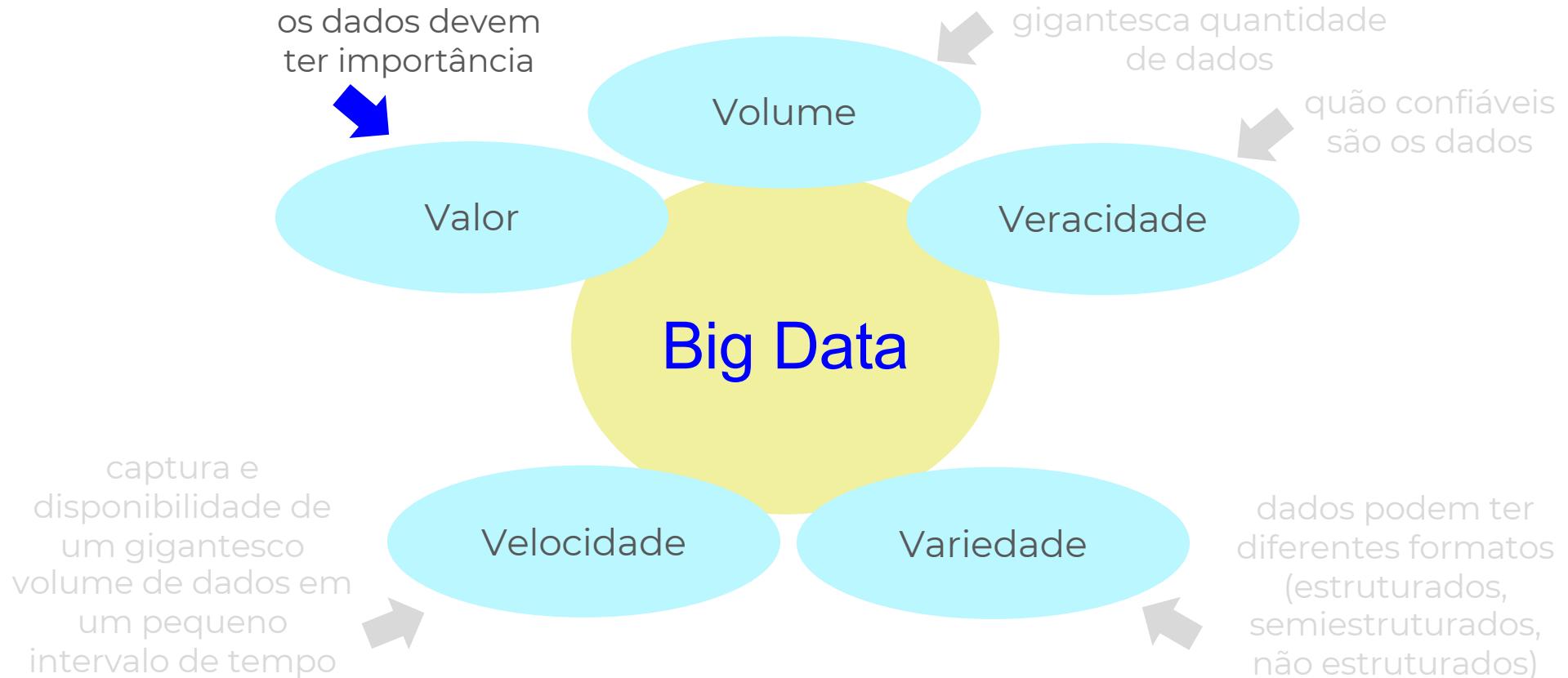
Modelo de 3Vs



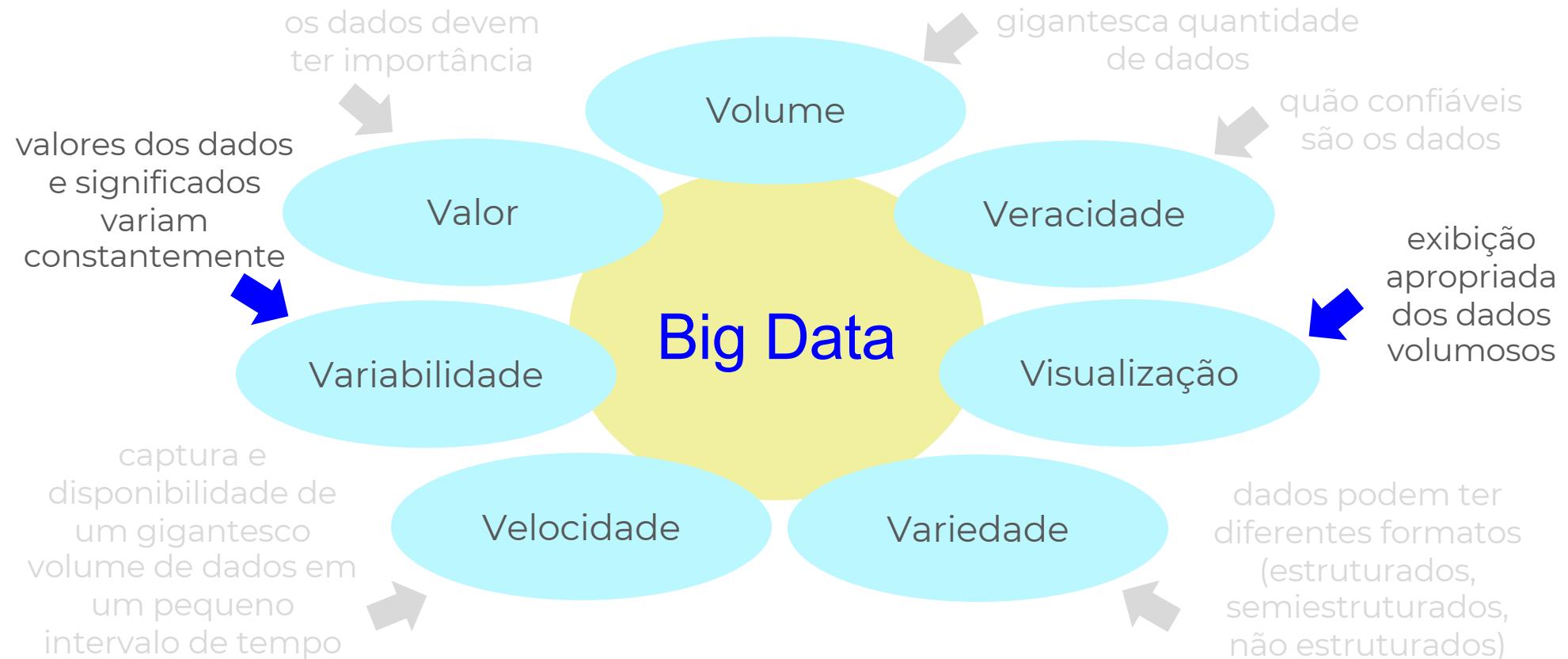
Modelo de 4Vs



Modelo de 5Vs



Modelo de 7Vs



Desafios

- Uso de **ambientes computacionais** com grande capacidade de armazenamento e processamento
 - *Clusters* de computadores
 - Computação em nuvem (*cloud computing*)
- Uso de **frameworks** de processamento paralelo e distribuído para simplificar a interação com os ambientes computacionais
 - Apache Hadoop
 - Apache Spark

Desafios

- Uso de **sistemas de arquivos distribuídos** para prover suporte para o armazenamento de grandes quantidades de dados
 - HDFS (*Hadoop Distributed File System*)
- Uso de **bases de dados NoSQL (Not only SQL)** para introduzir flexibilidade no armazenamento de diferentes tipos de dados
 - Não estruturados
 - Semiestruturados
 - Estruturados

Nuvem de Conceitos e Tecnologias

Velocidade JSON
Veracidade Cluster
Pipeline Spark HDFS
Variabilidade Databases
Druid Data Warehouse NoSQL
Kafka Hive
ELT AWS
Azure ETL Analytics Hadoop
Visualização Streaming Variedade
Valor Volume
SQL Data Lake
Petabytes Cloud Computing

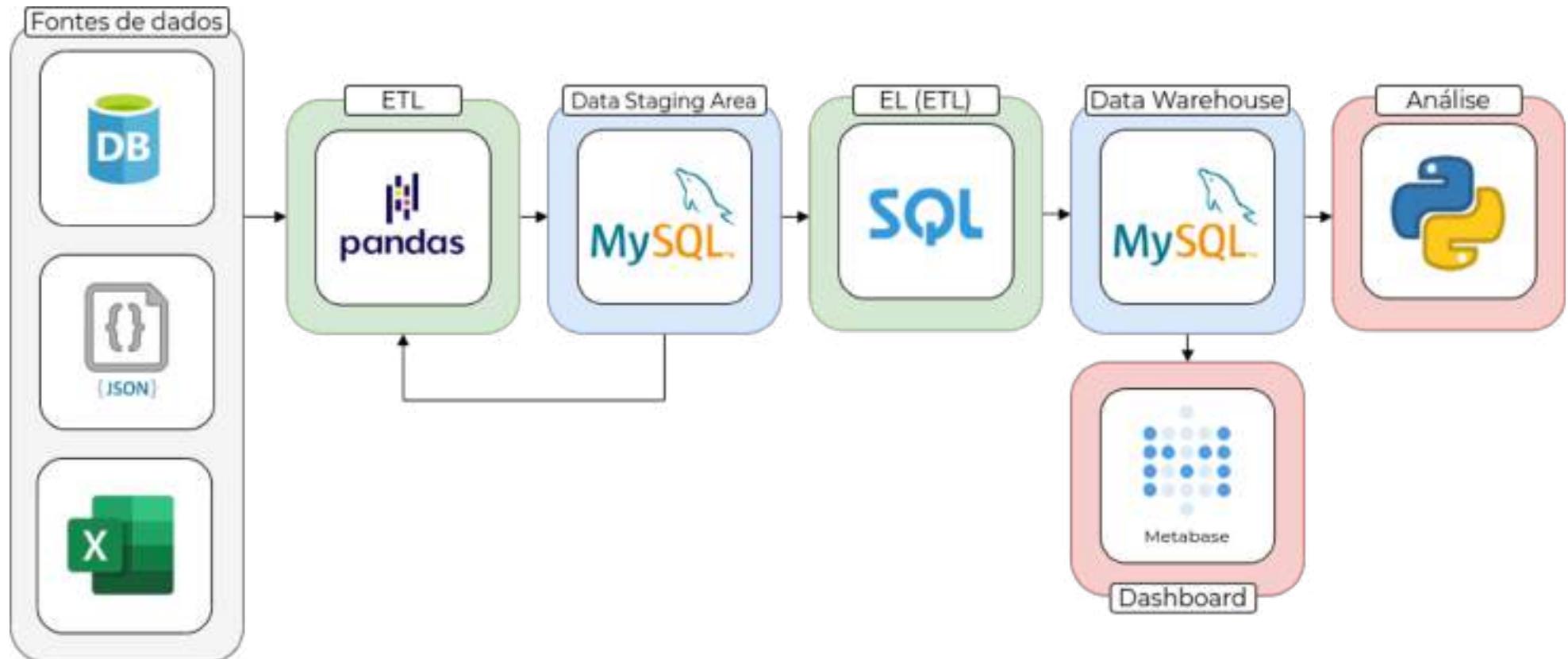
Agenda

- Visão Geral
- Diferenças entre os Locais de Armazenamento
- Big Data
- Exemplos de Pipeline

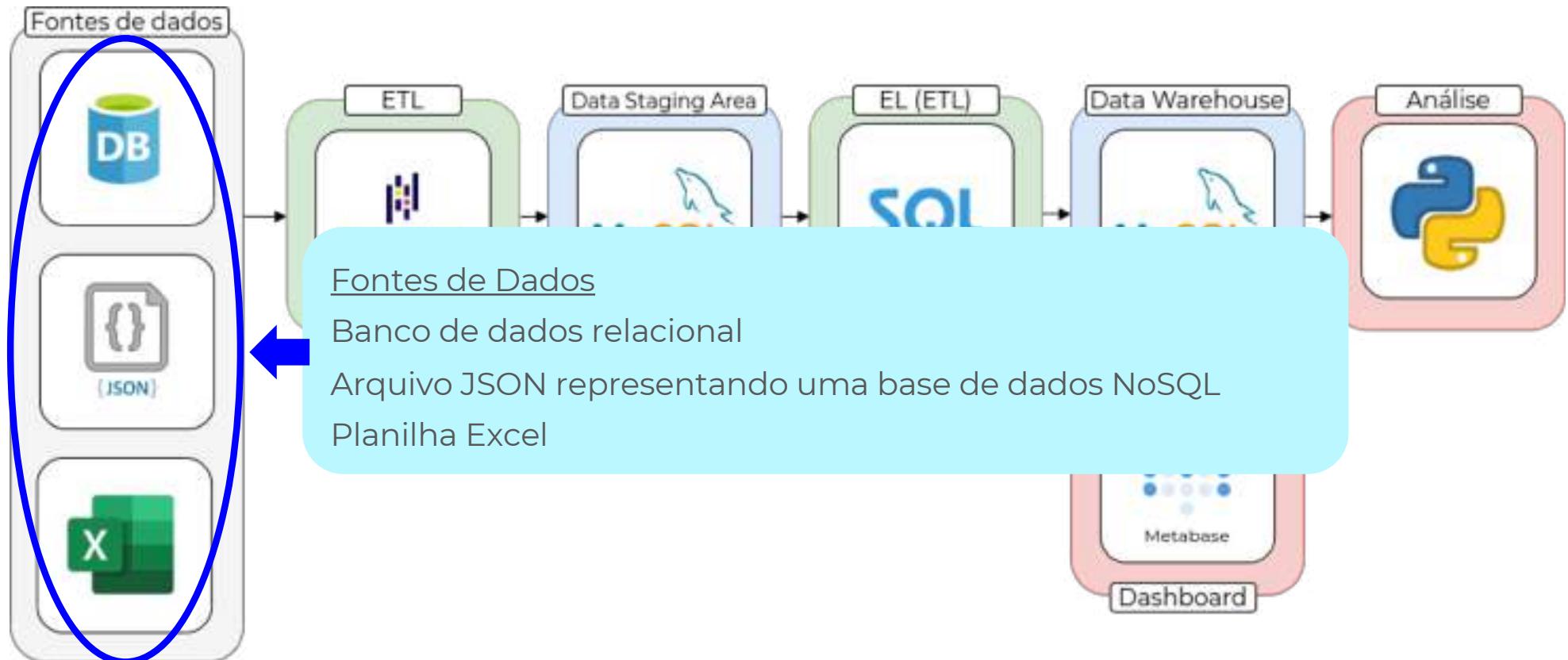
Exemplos de Pipeline

- Volumes de Dados Tradicionais
- Big Data
- Data Streaming

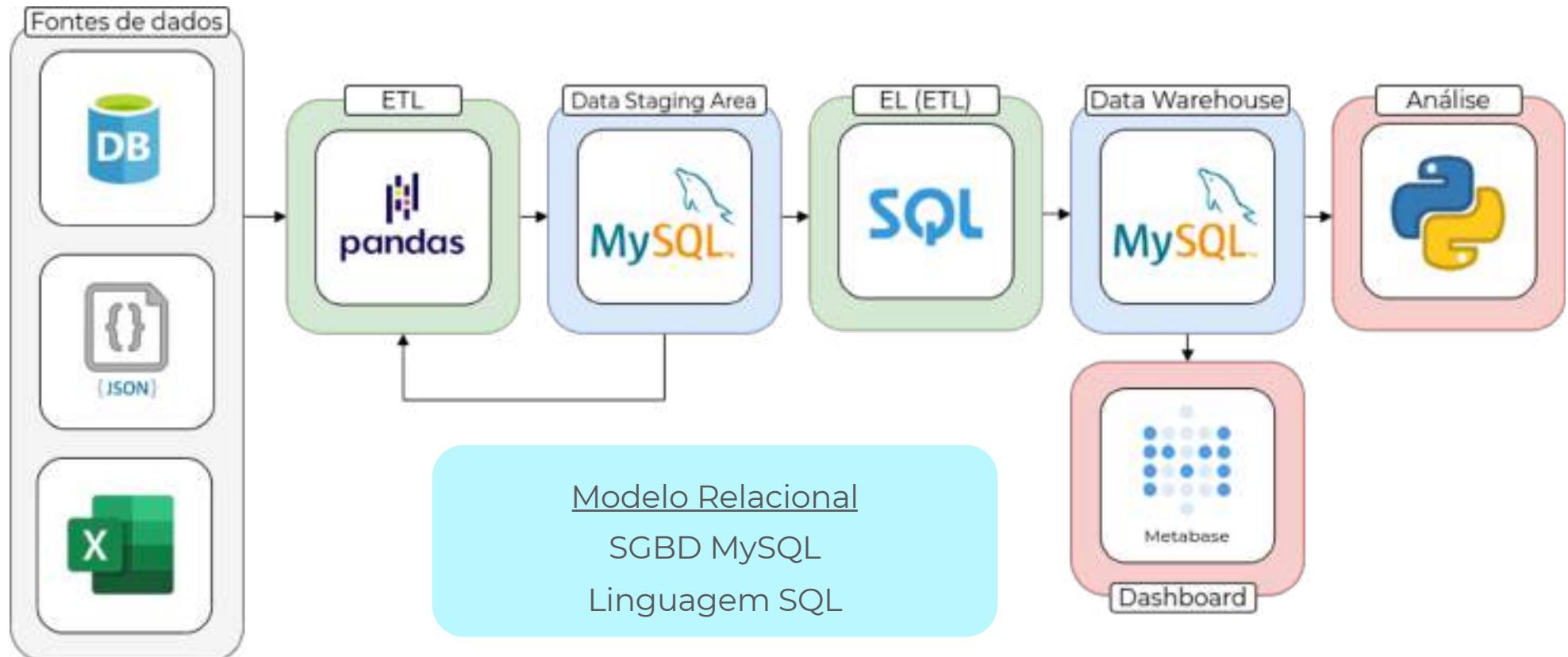
Processamento de Dados em Lote



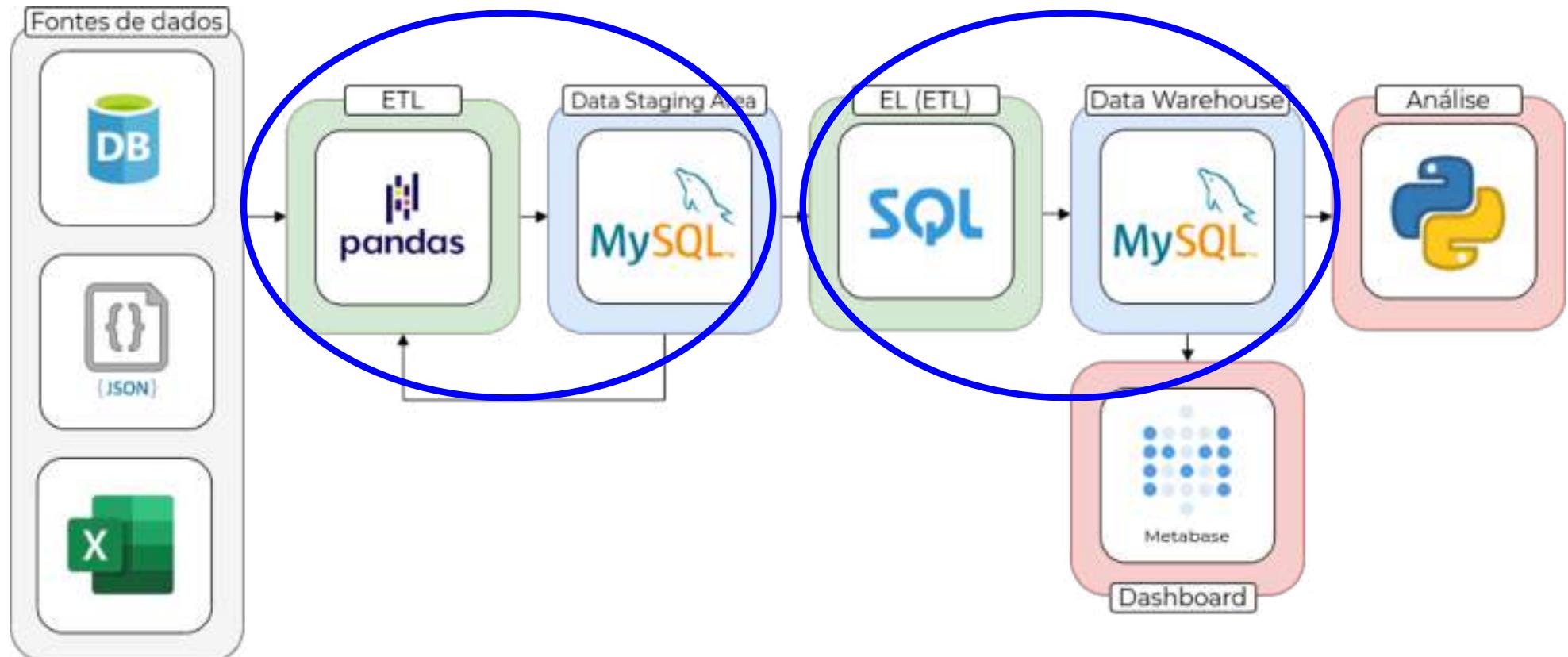
Processamento de Dados em Lote



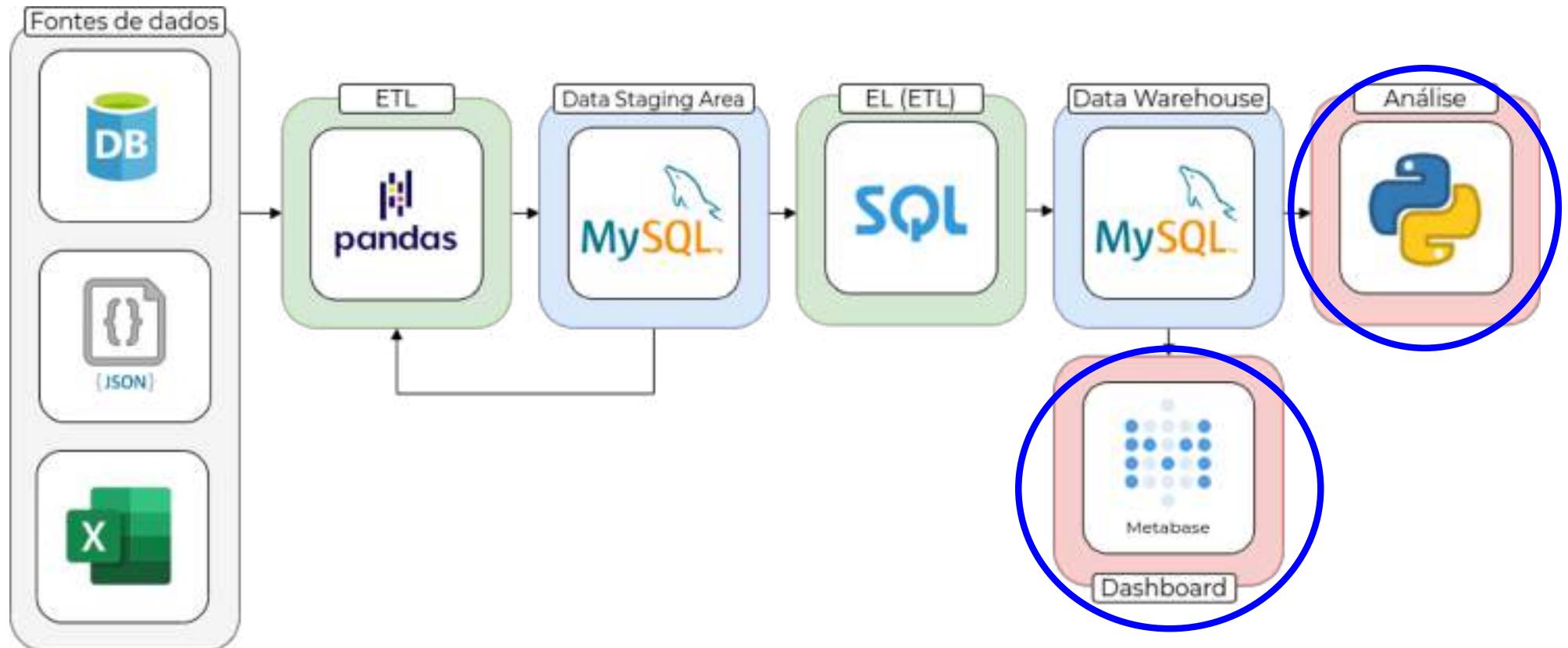
Processamento de Dados em Lote



Processamento de Dados em Lote



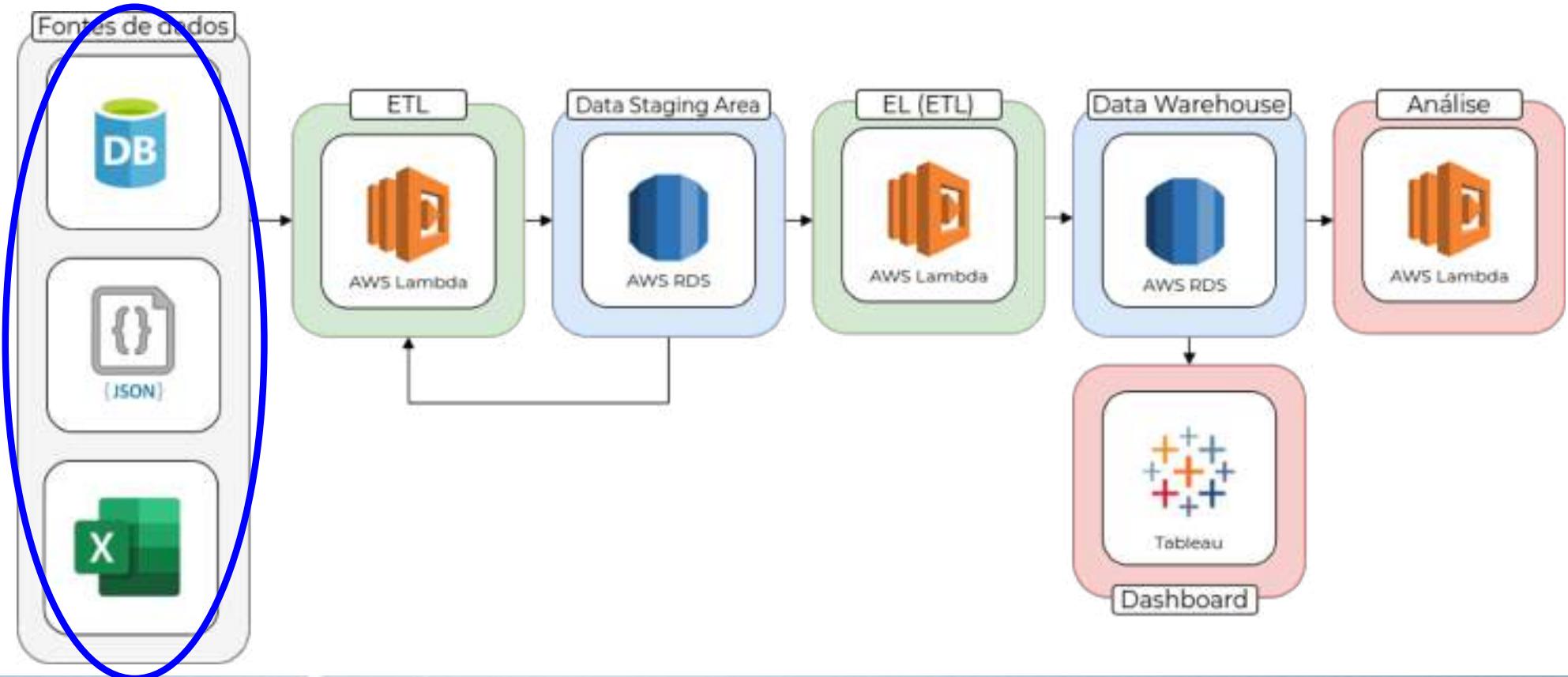
Processamento de Dados em Lote



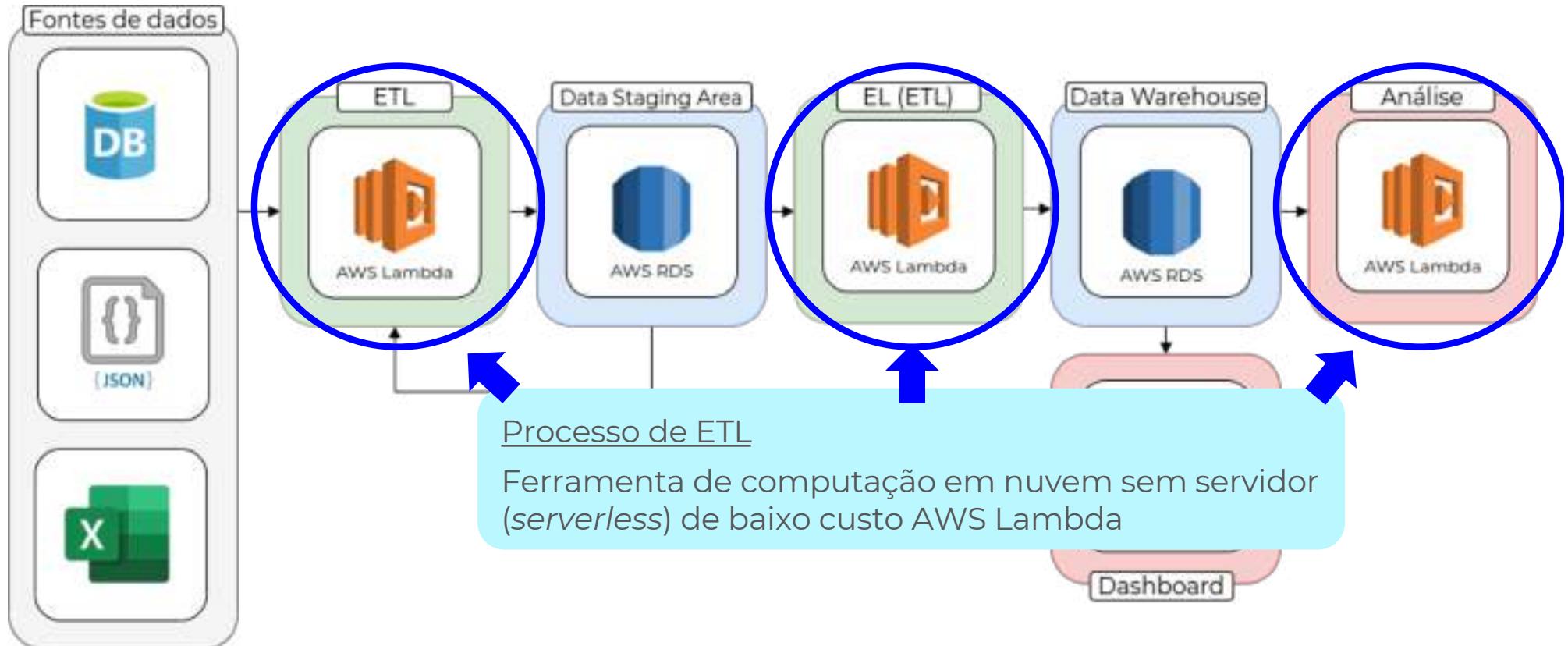
Processamento de Dados em Lote (Nuvem)



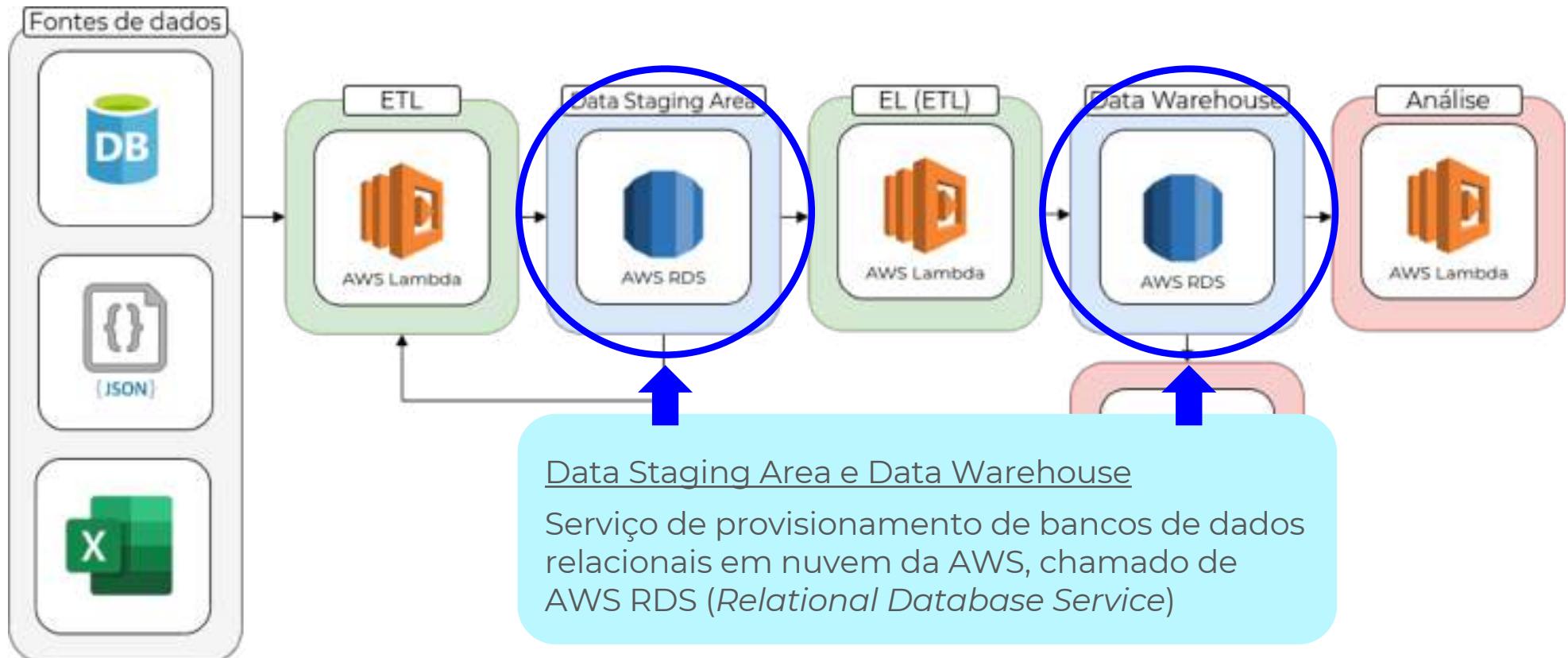
Processamento de Dados em Lote (Nuvem)



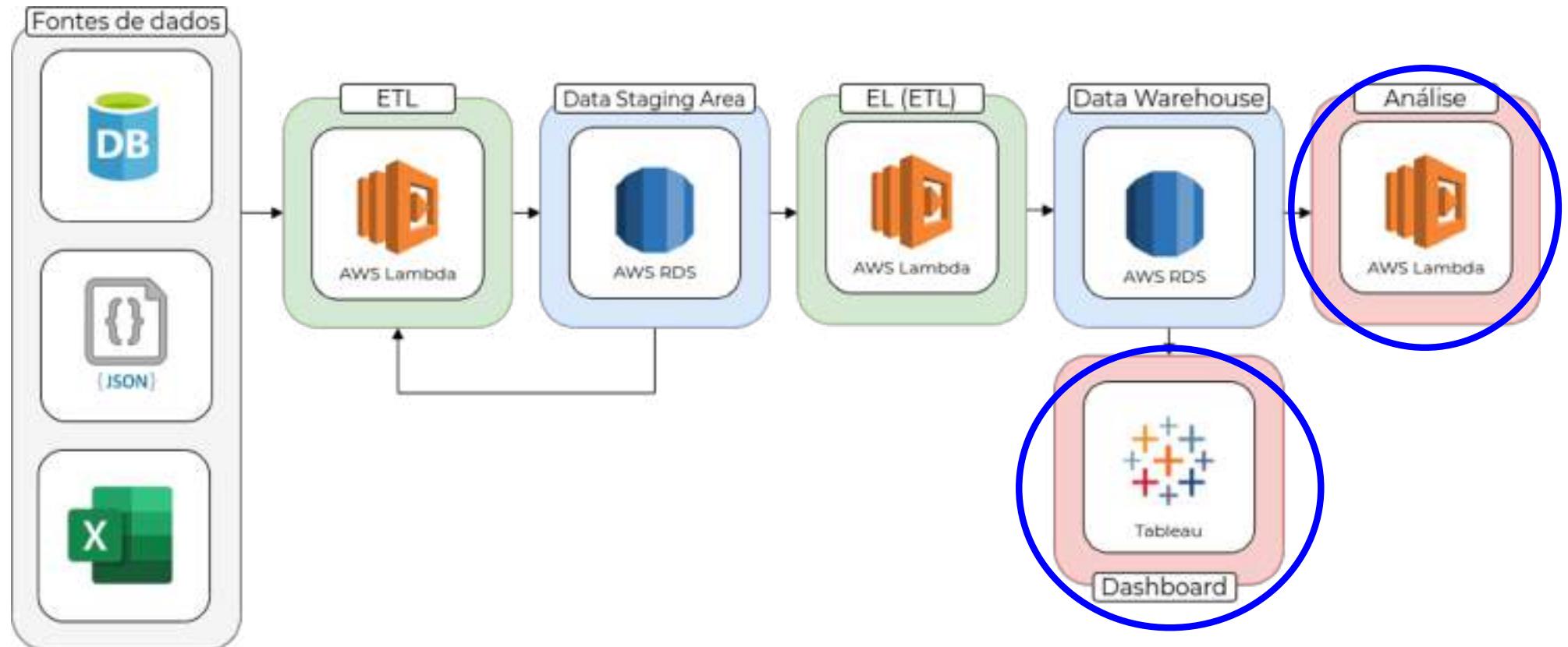
Processamento de Dados em Lote (Nuvem)



Processamento de Dados em Lote (Nuvem)



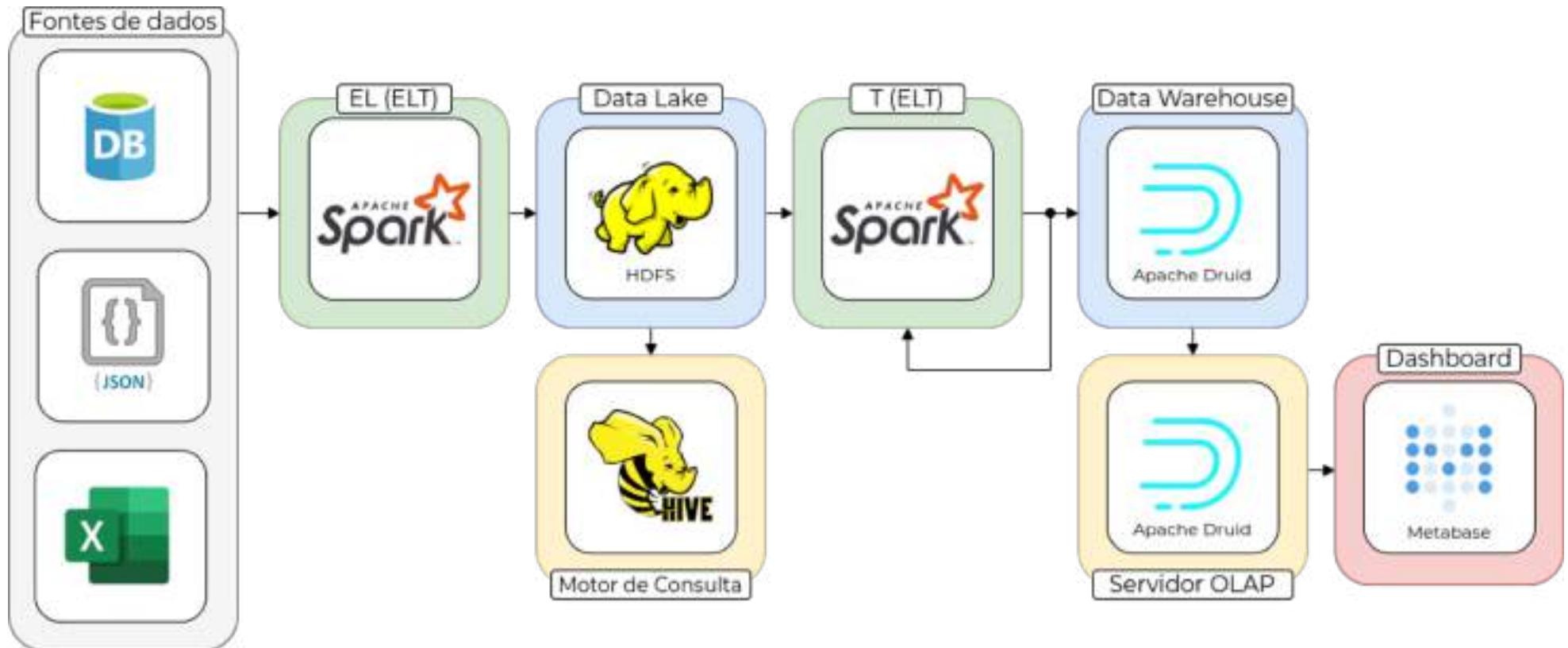
Processamento de Dados em Lote (Nuvem)



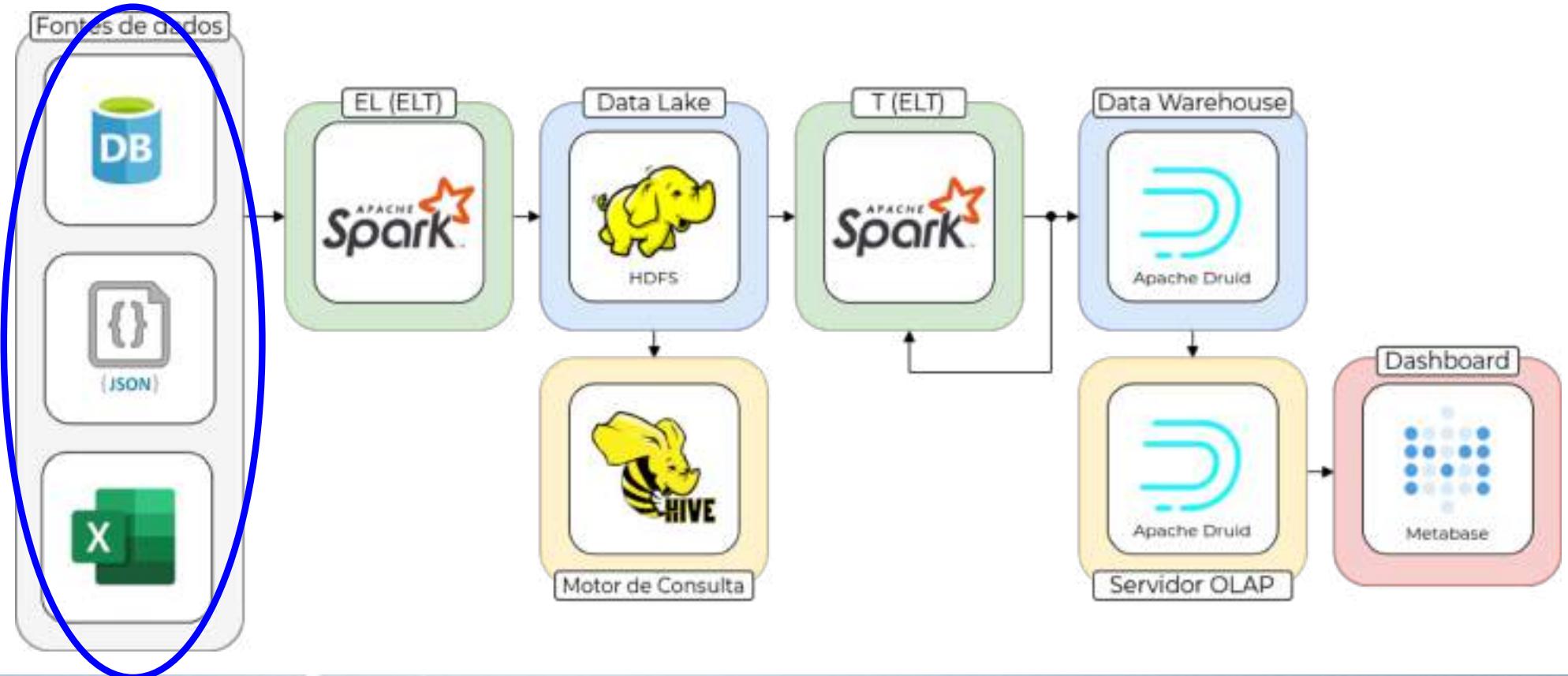
Exemplos de Pipeline

- Volumes de Dados Tradicionais
- Big Data
- Data Streaming

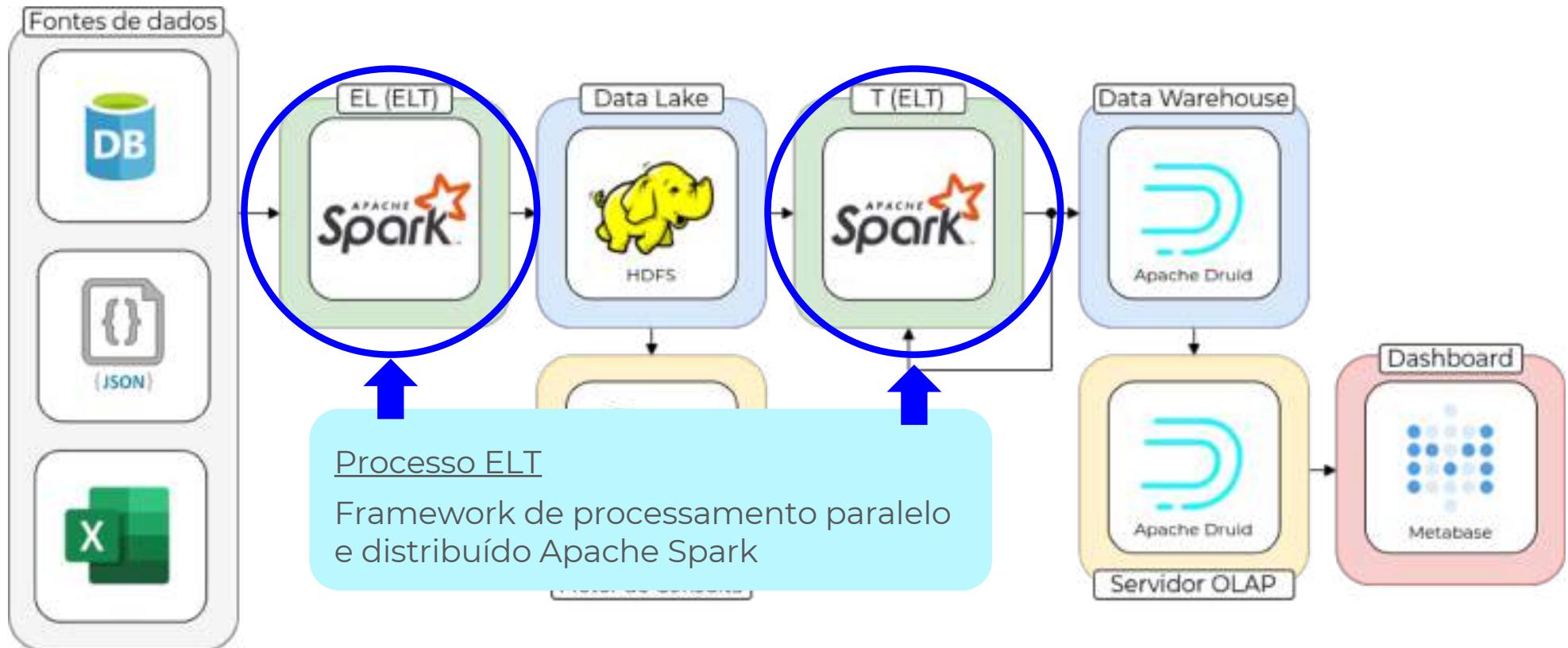
Processamento de Big Data em Lote



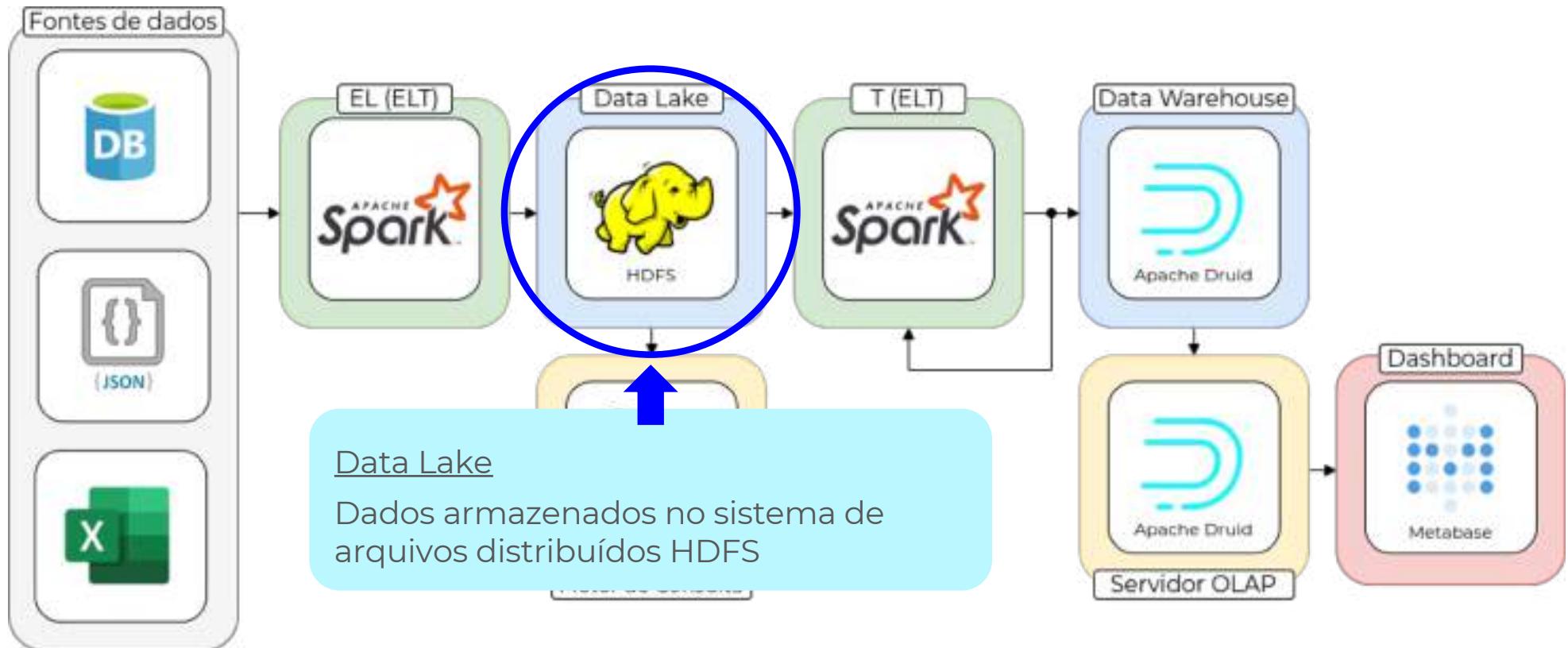
Processamento de Big Data em Lote



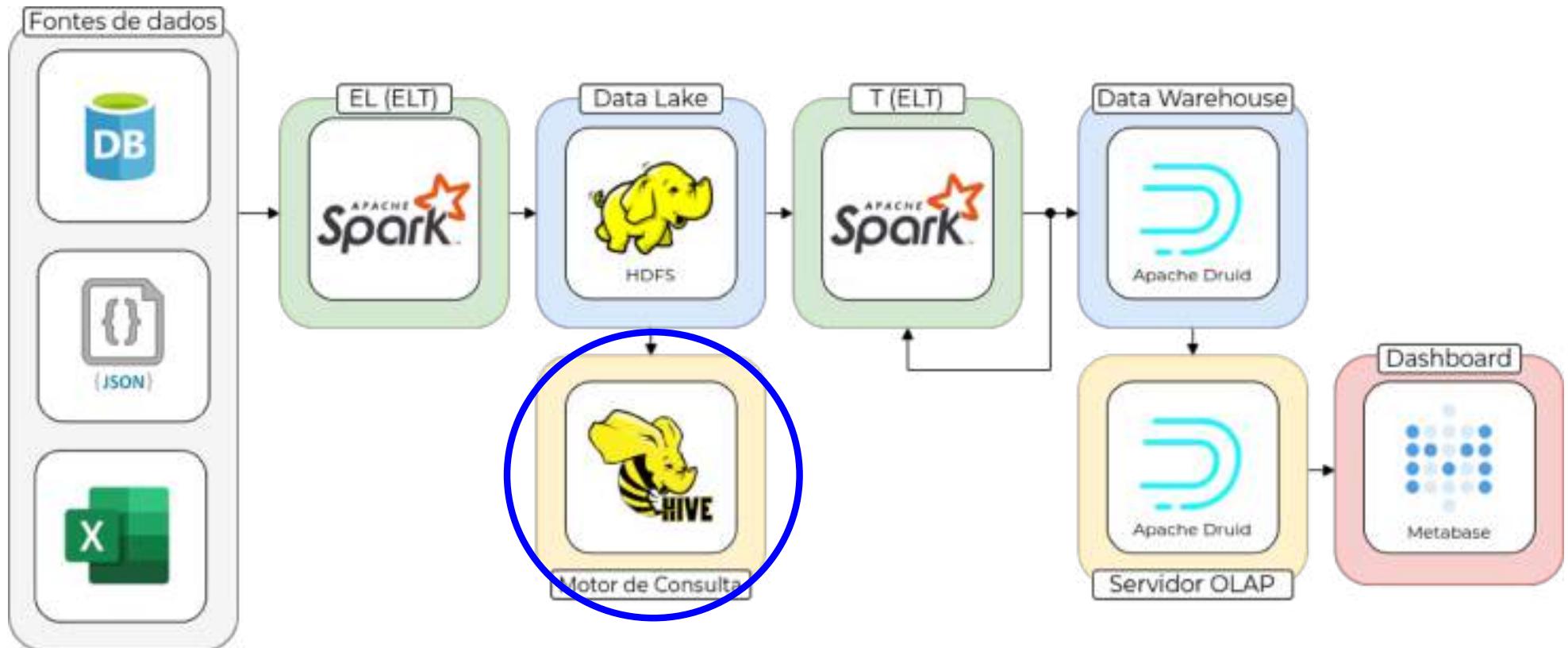
Processamento de Big Data em Lote



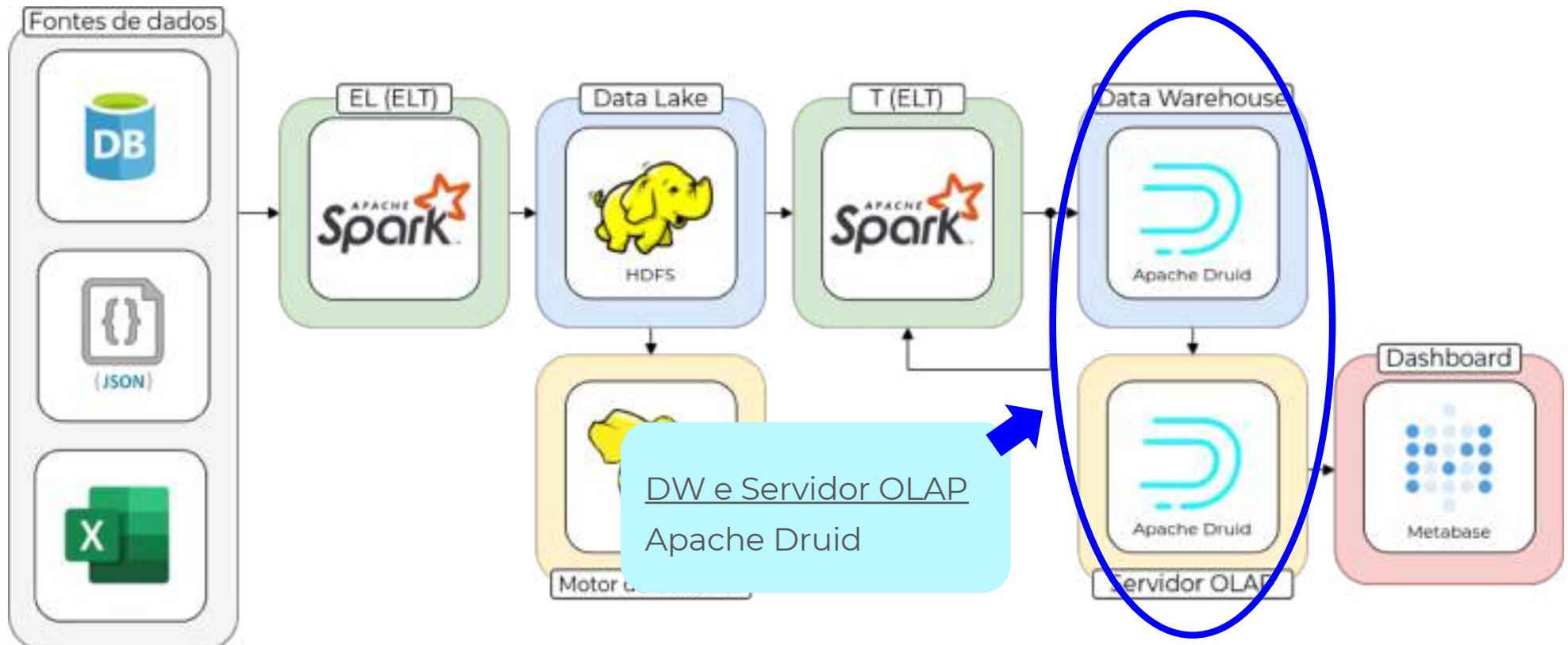
Processamento de Big Data em Lote



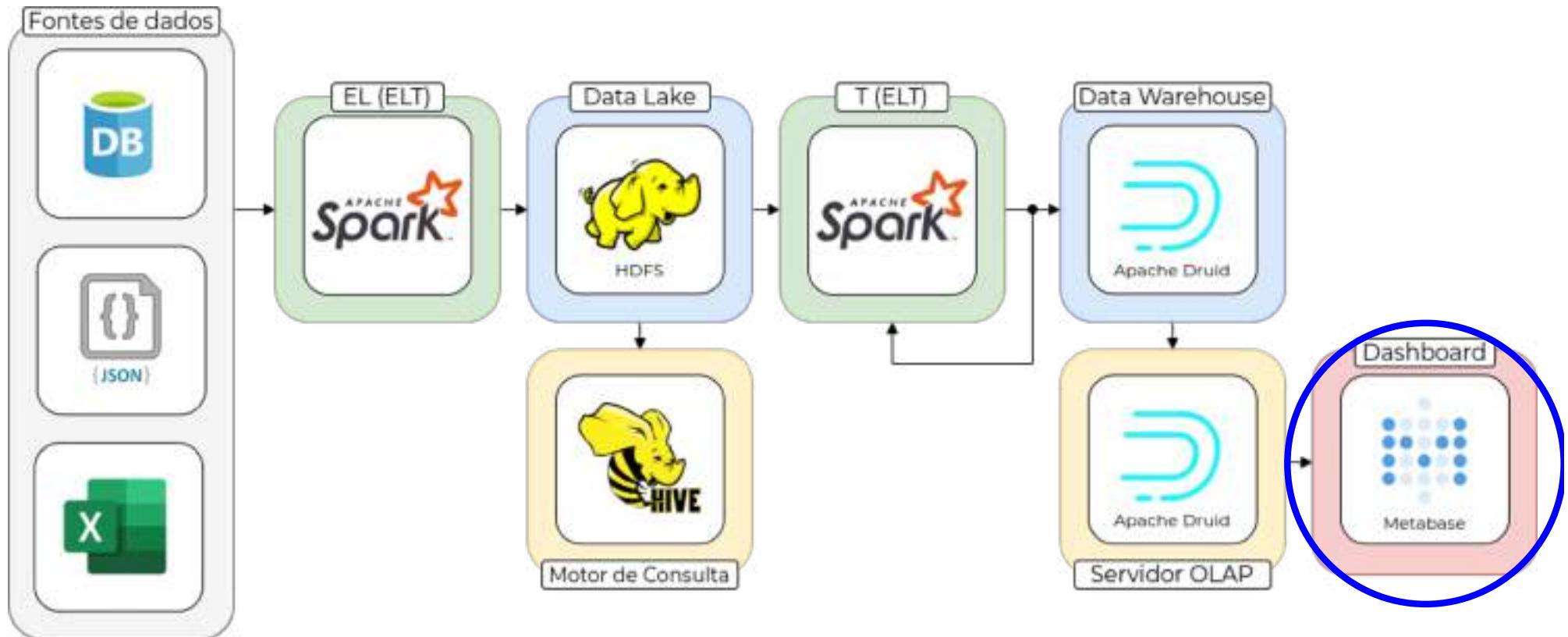
Processamento de Big Data em Lote



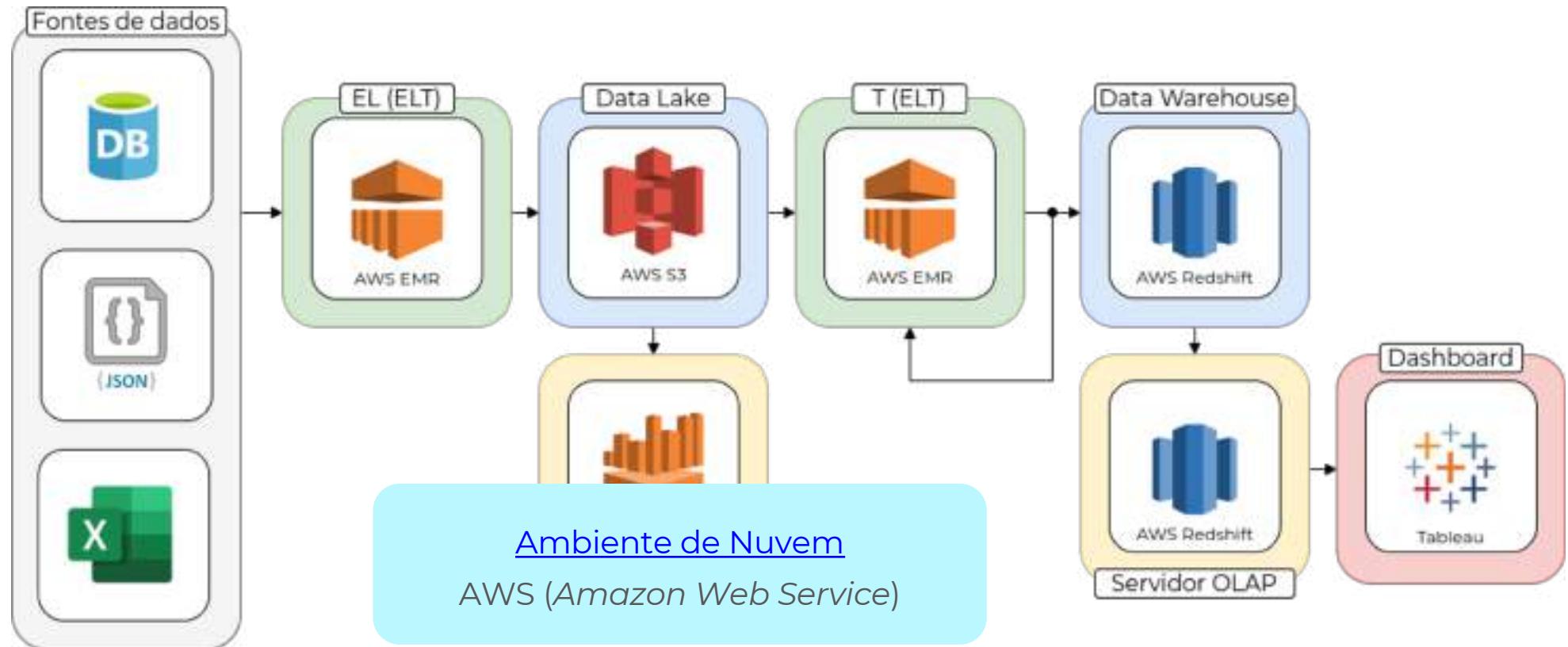
Processamento de Big Data em Lote



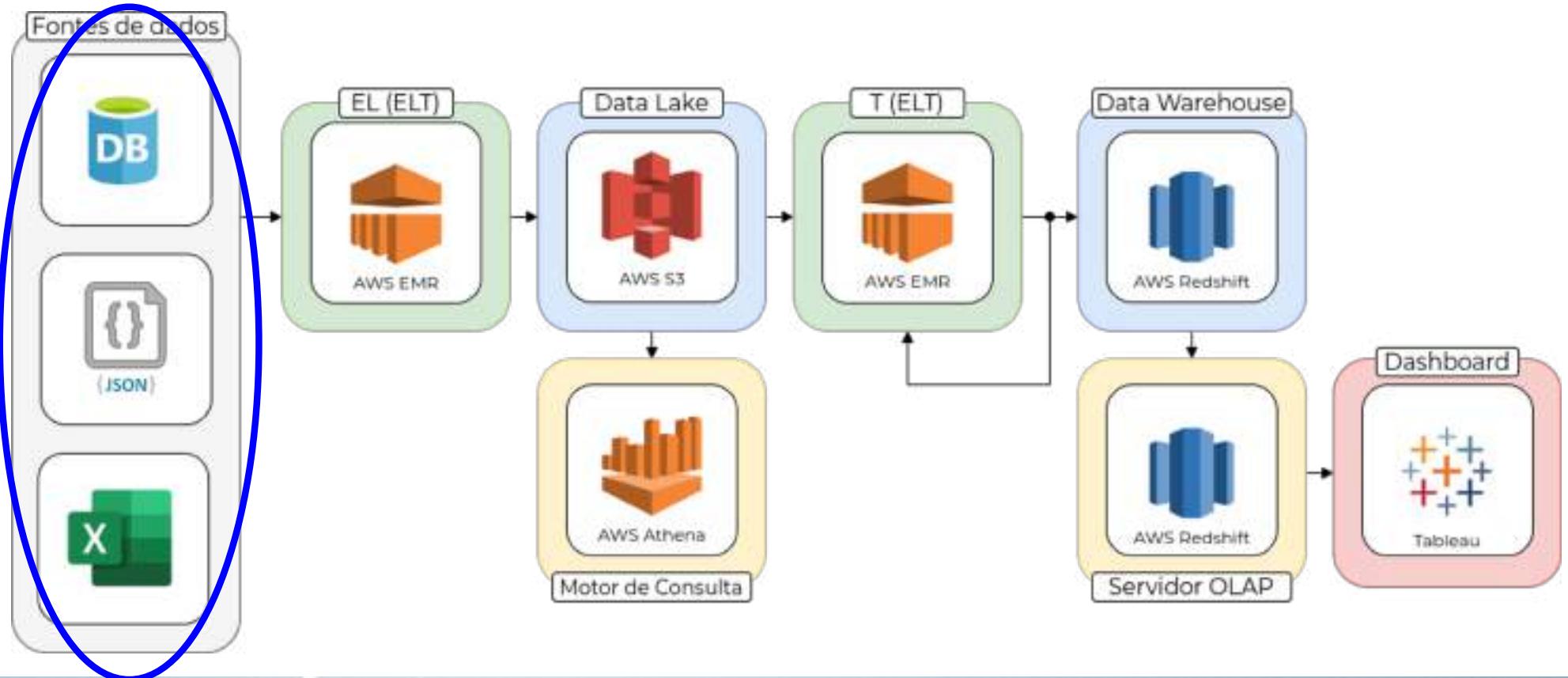
Processamento de Big Data em Lote



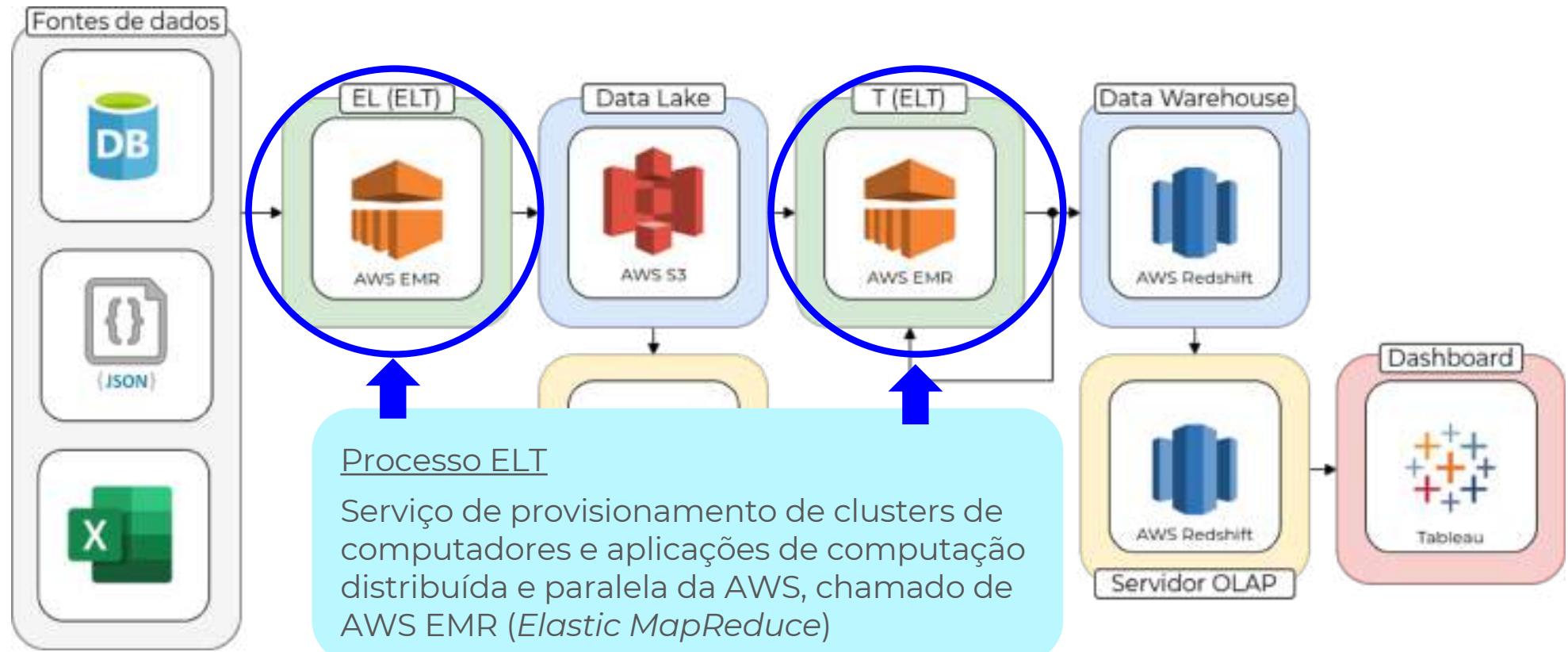
Processamento de Big Data em Lote (Nuvem)



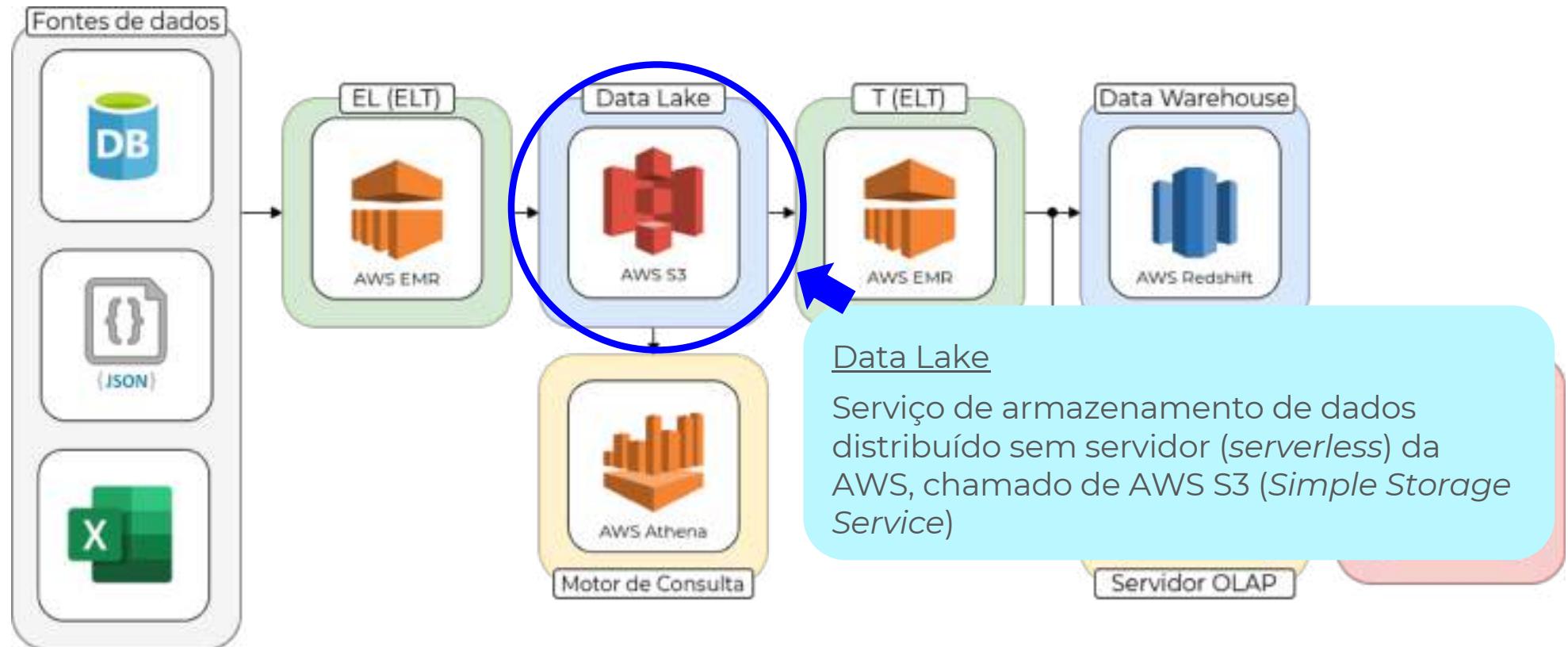
Processamento de Big Data em Lote (Nuvem)



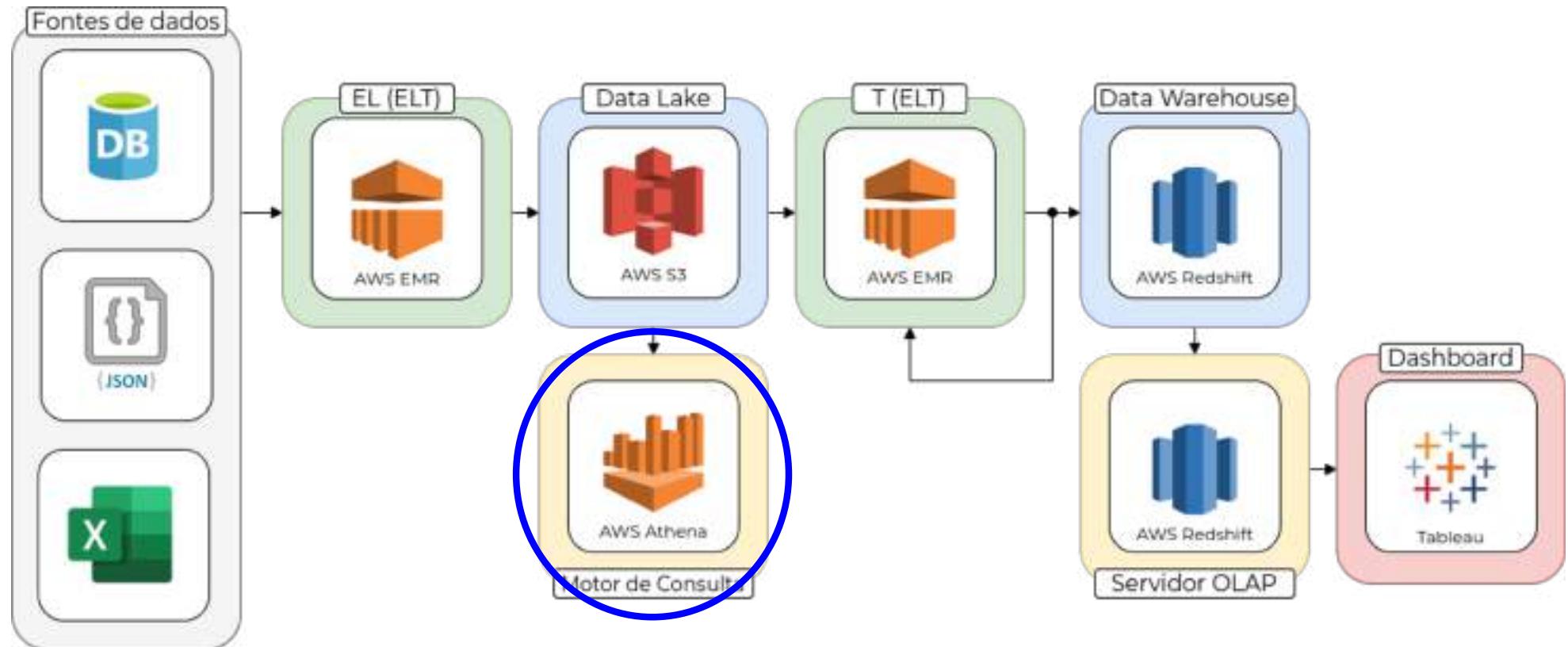
Processamento de Big Data em Lote (Nuvem)



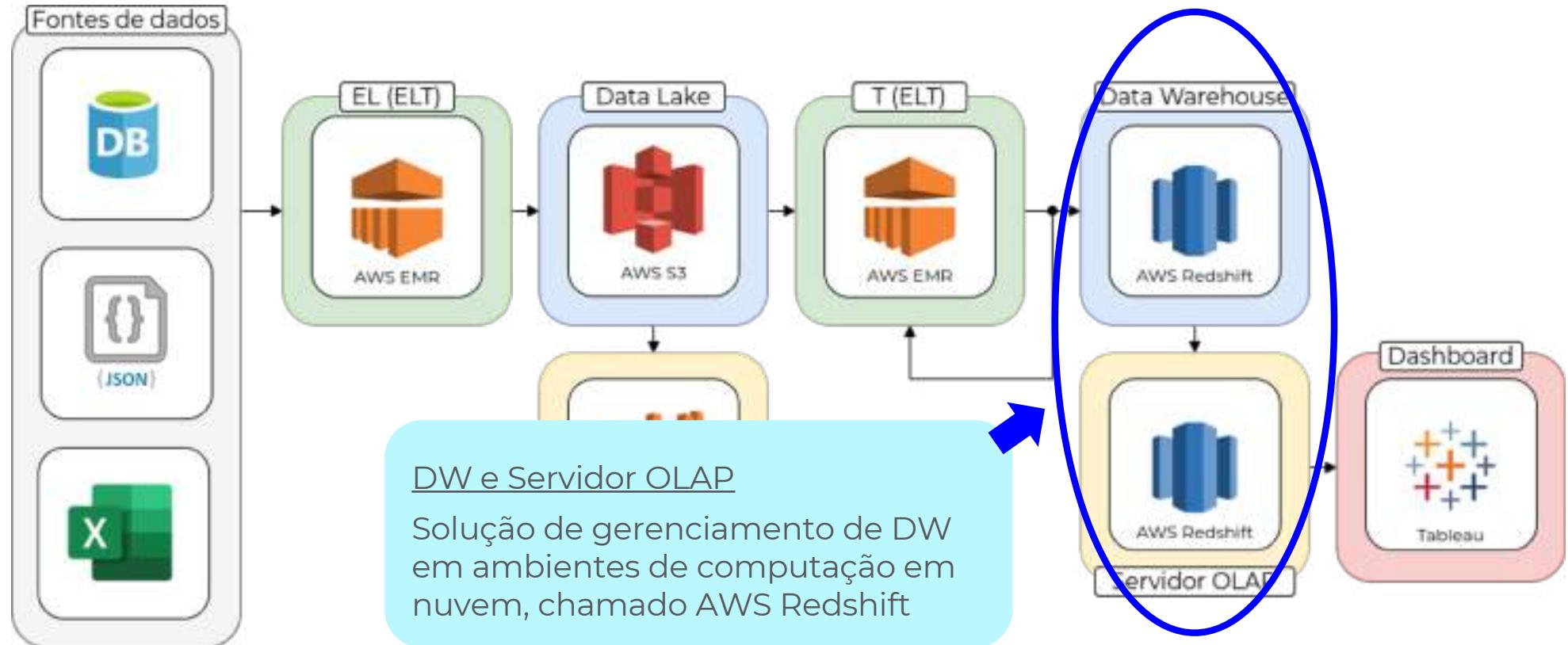
Processamento de Big Data em Lote (Nuvem)



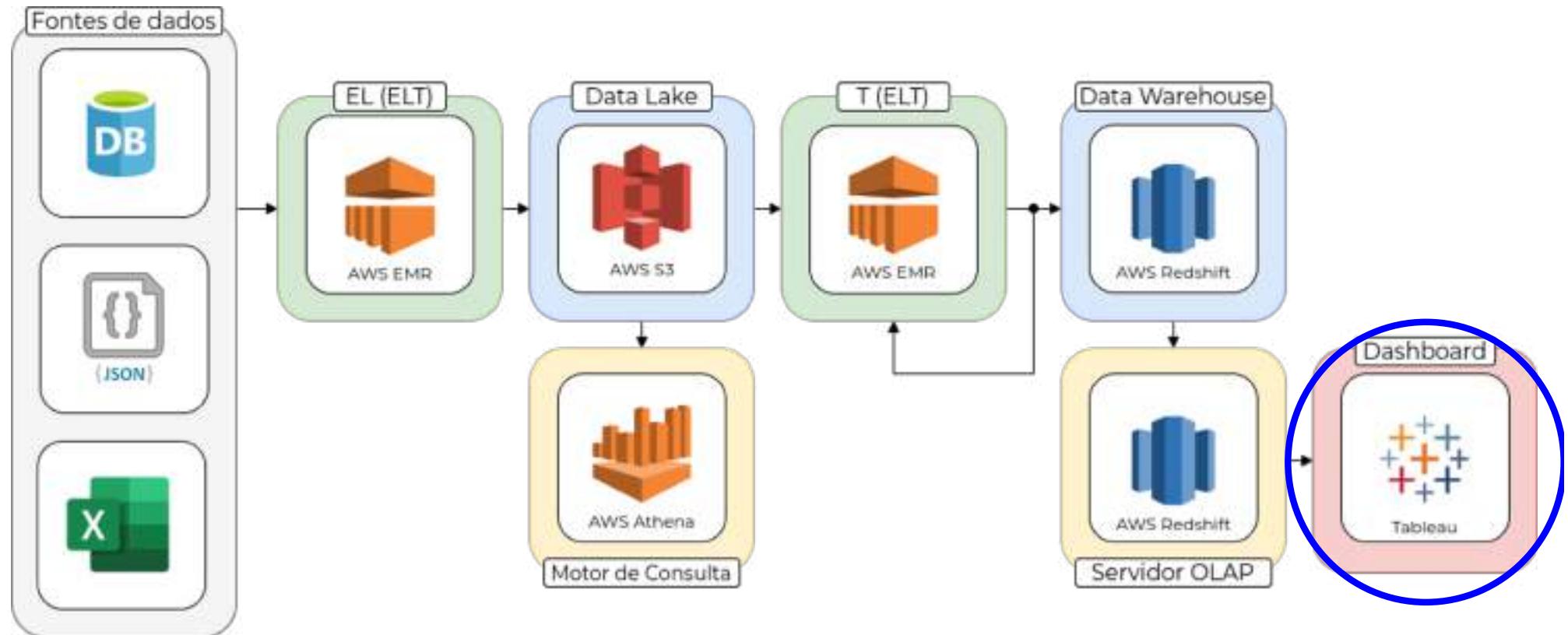
Processamento de Big Data em Lote (Nuvem)



Processamento de Big Data em Lote (Nuvem)



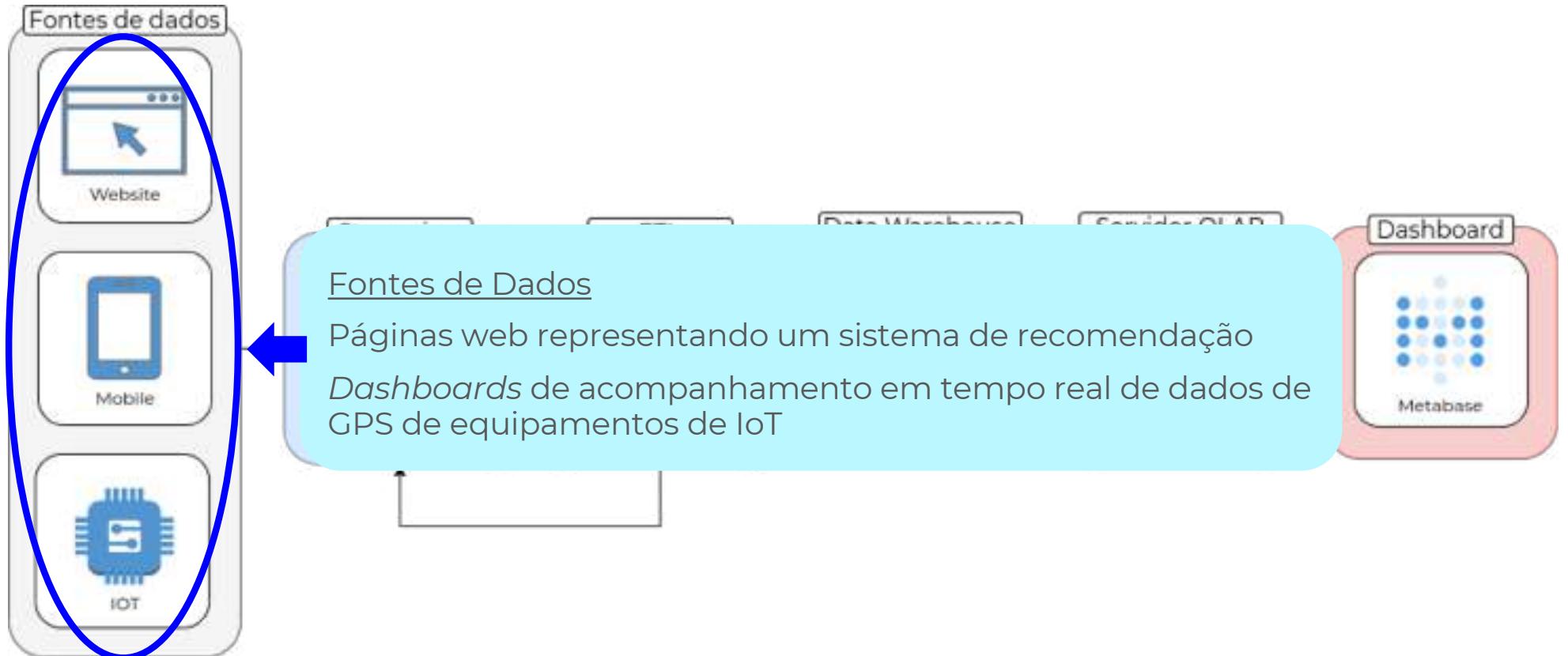
Processamento de Big Data em Lote (Nuvem)



Exemplos de Pipeline

- Volumes de Dados Tradicionais
- Big Data
- Data Streaming

Processamento de Streaming de Big Data



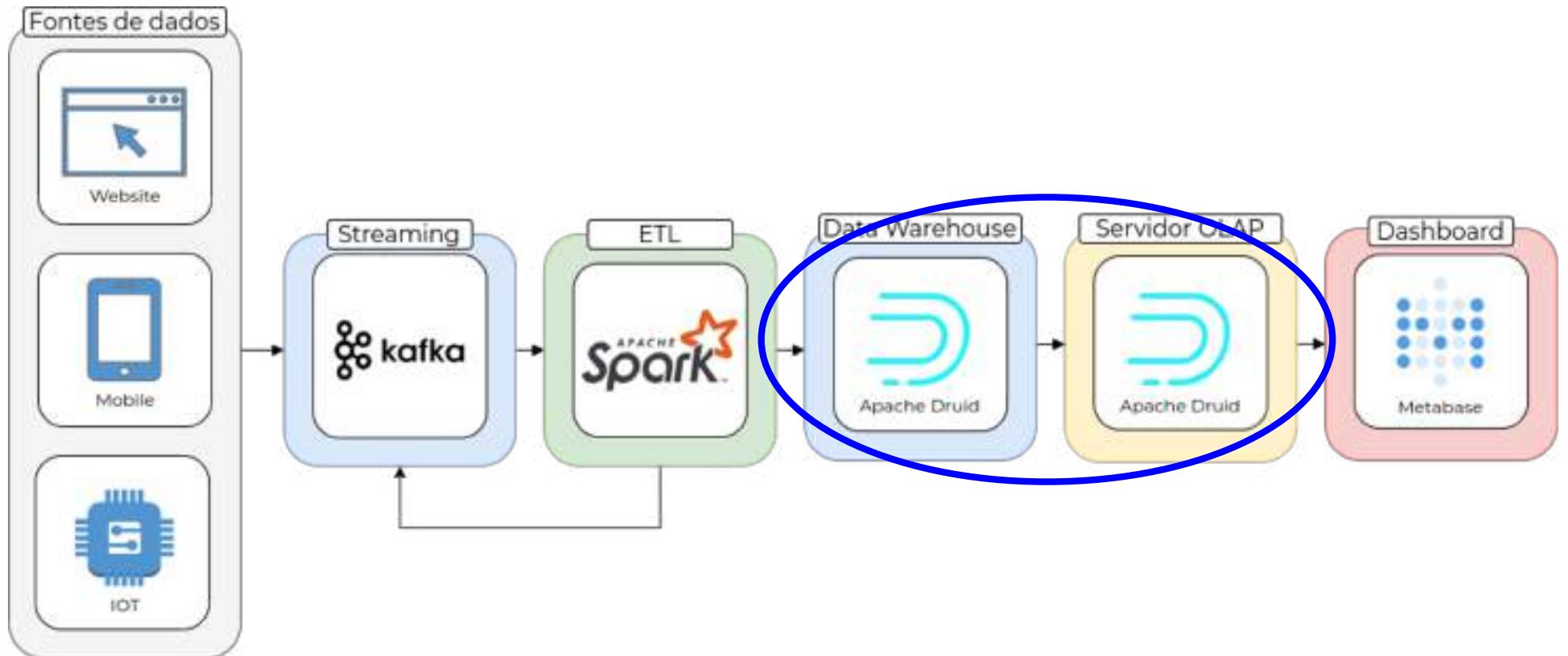
Processamento de Streaming de Big Data



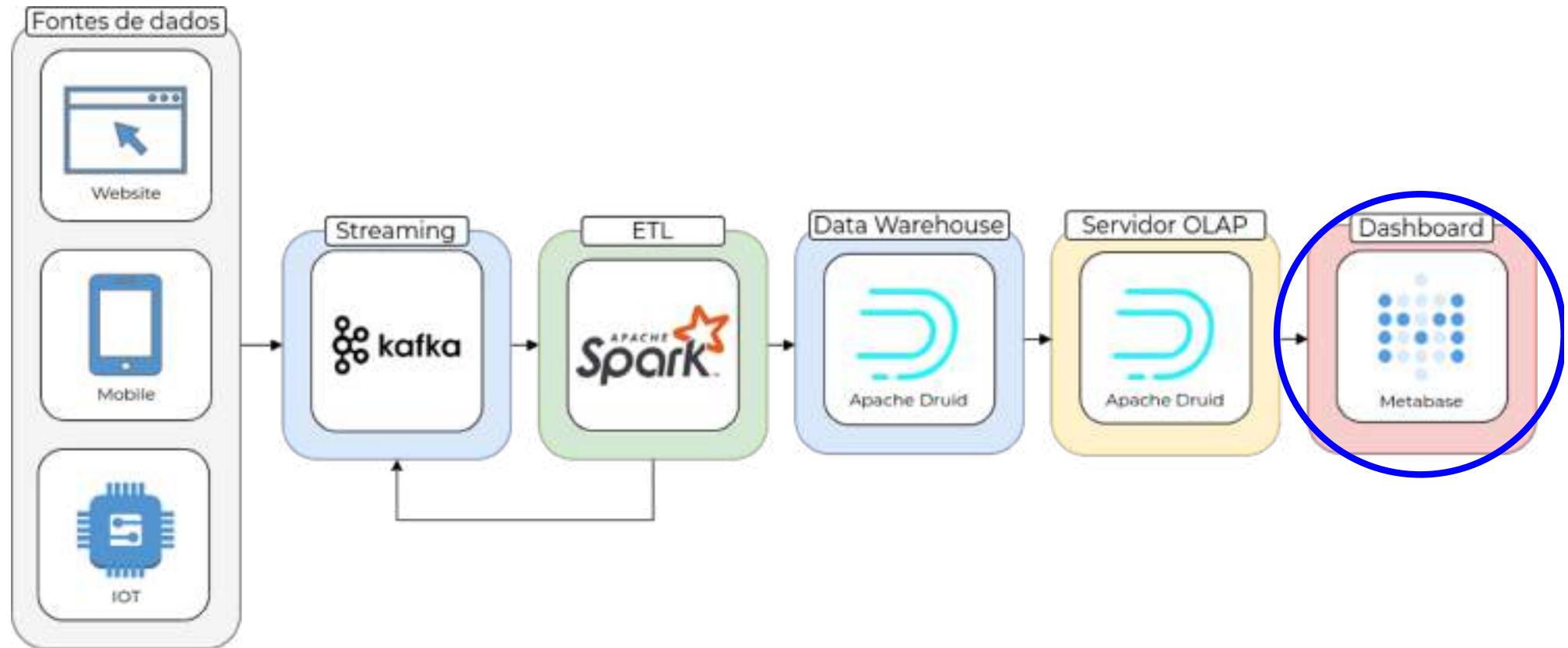
Processamento de Streaming de Big Data



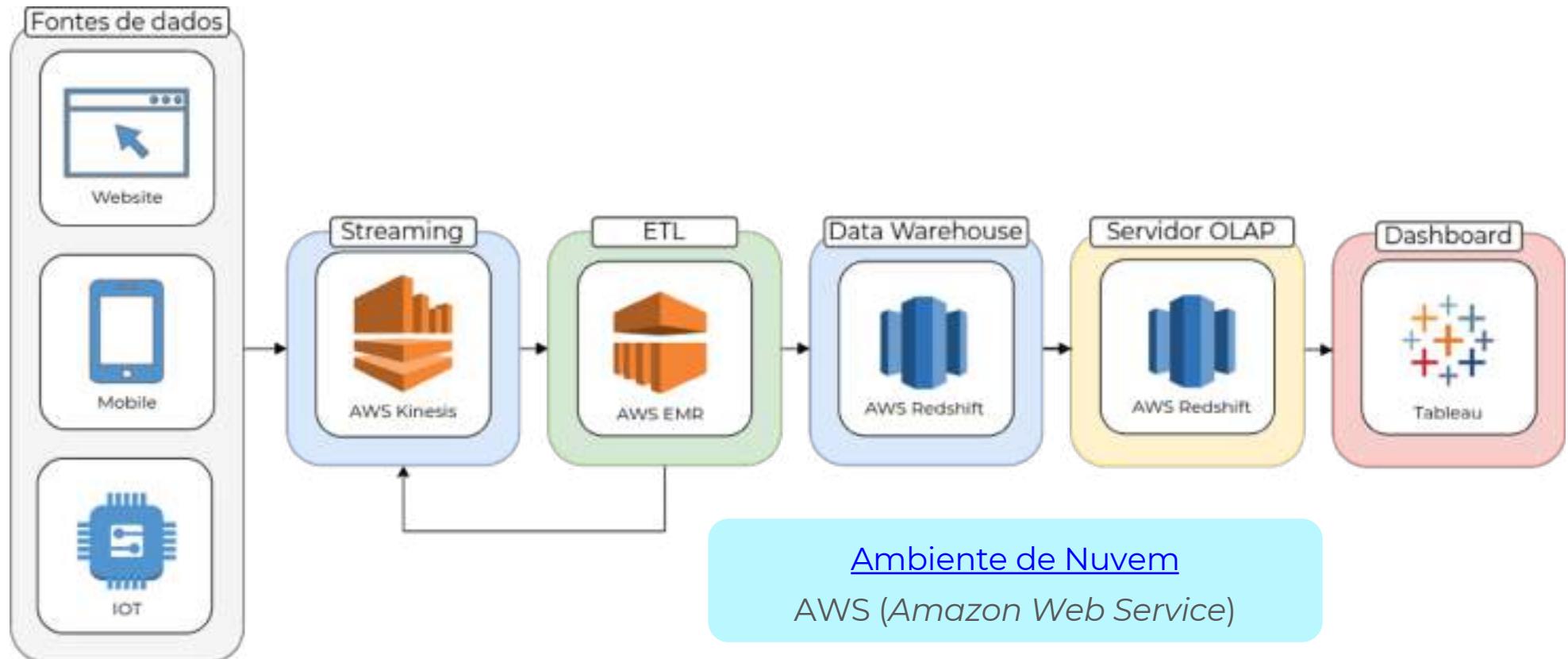
Processamento de Streaming de Big Data



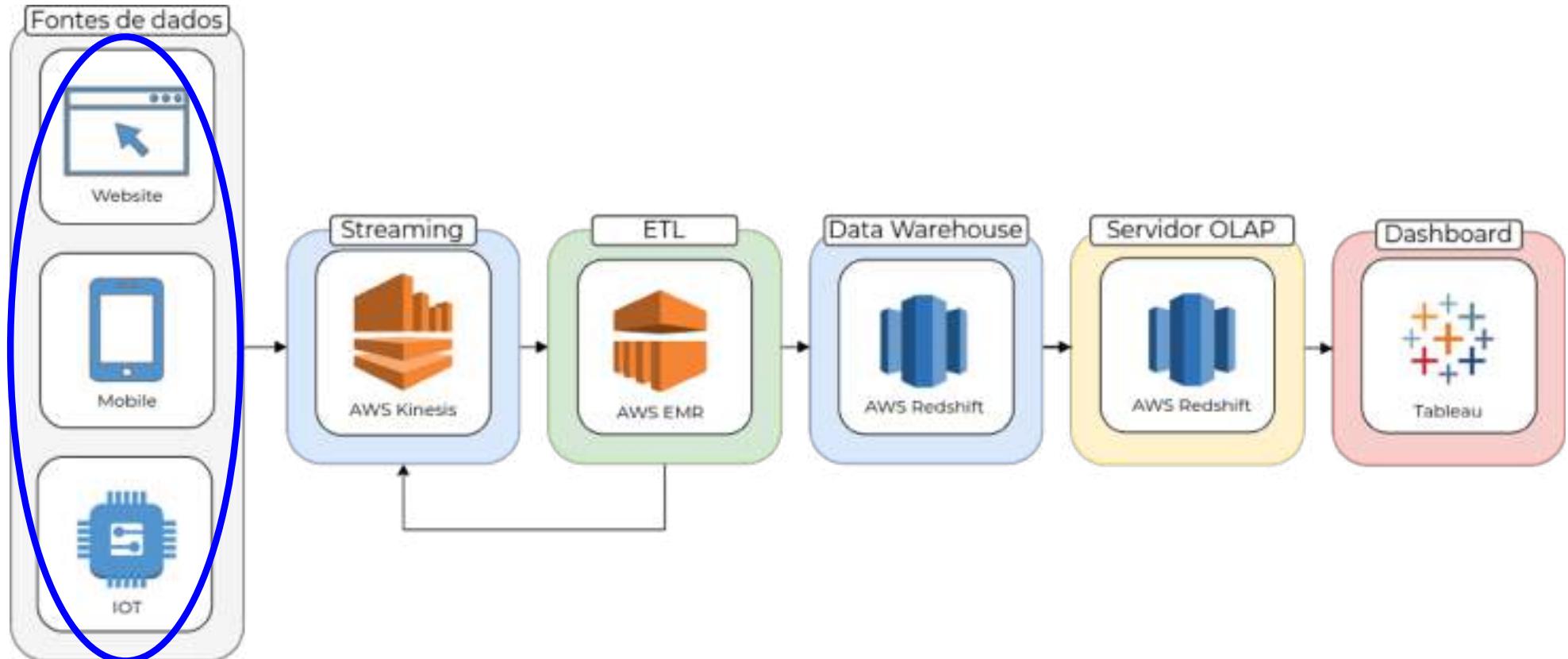
Processamento de Streaming de Big Data



Processamento de Streaming de Big Data (Nuvem)



Processamento de Streaming de Big Data (Nuvem)



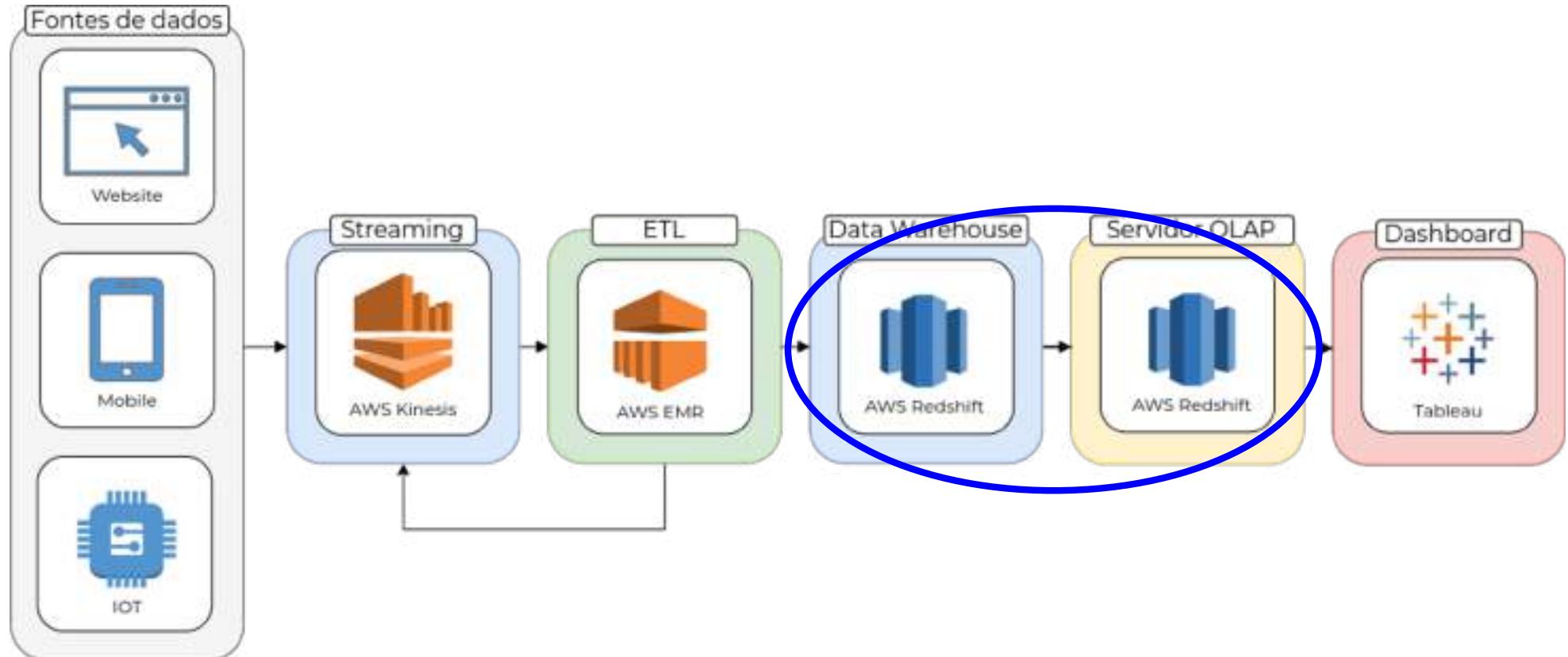
Processamento de Streaming de Big Data (Nuvem)



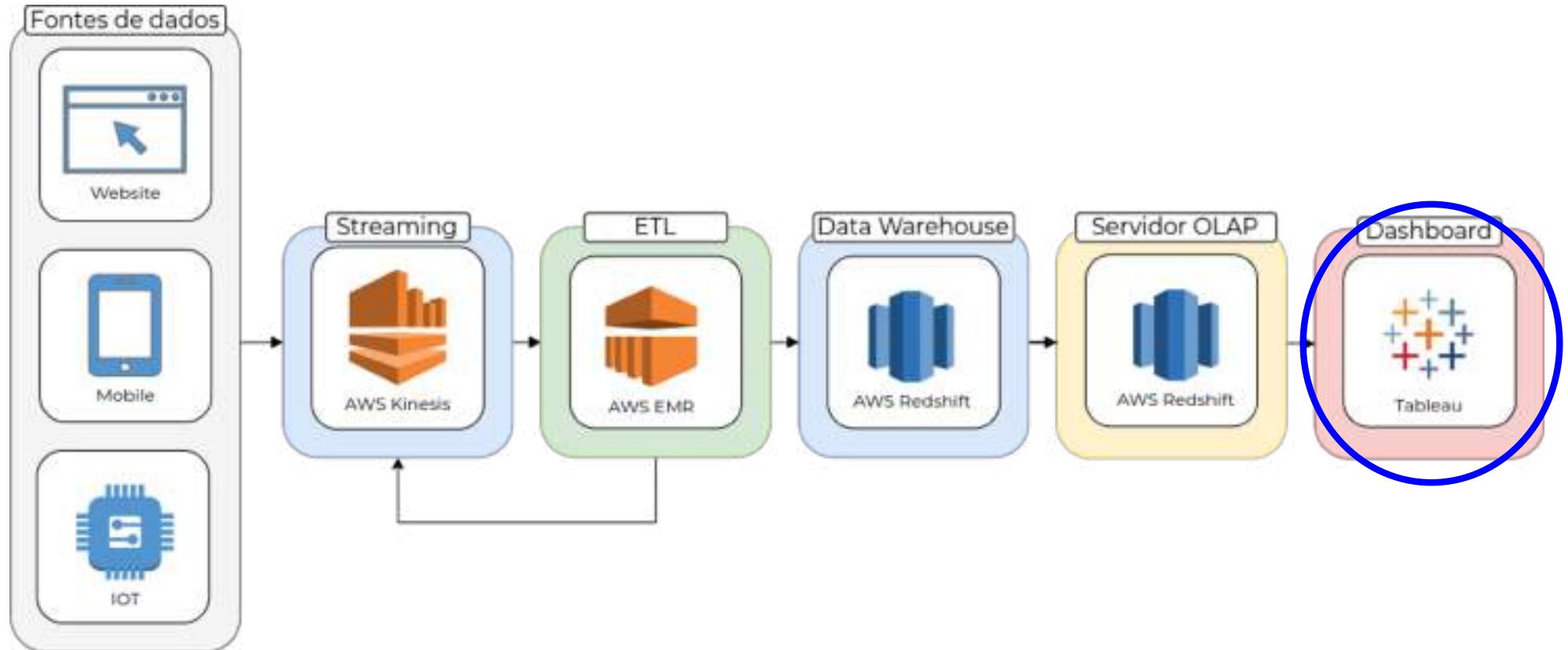
Processamento de Streaming de Big Data (Nuvem)



Processamento de Streaming de Big Data (Nuvem)



Processamento de Streaming de Big Data (Nuvem)



Análise de Dados com Base em Processamento Massivo em Paralelo

Arquitetura de Data Warehousing

Profa. Dra. Cristina Dutra de Aguiar

Resumo:

Data warehousing engloba arquiteturas, algoritmos e ferramentas que possibilitam que dados selecionados de fontes de dados autônomas, heterogêneas e distribuídas sejam integrados em um único banco de dados, conhecido como *data warehouse*. Por meio da arquitetura de *data warehousing*, é possível identificar os componentes que participam do ambiente, o relacionamento que existe entre esses componentes e as funcionalidades de cada um. Neste texto são descritos conceitos relacionados à arquitetura de *data warehousing* e às diferenças entre os locais de armazenamento de dados presentes nessa arquitetura. Também são descritos conceitos relacionados a *big data*, incluindo sua definição e os desafios relacionados. Por fim, são ilustrados exemplos de arquiteturas instanciadas por meio de tecnologias, chamadas no mercado de trabalho de *pipelines*.

Conteúdo

1 Arquitetura de <i>Data Warehousing</i>	3
1.1 Fontes de Dados	3
1.2 Camada de Pré-processamento dos Dados	4
1.3 Camada de Data Warehouse	4
1.4 Camada de Serviços	5
1.5 Camada de Ferramentas de Análise e Consulta	6
2 Diferenças entre Locais de Armazenamento	7
2.1 Data Warehouse e Data Marts	7
2.2 Data Staging Area e Data Lake	8
2.3 Data Warehouse e Data Lake	9
3 Big Data	10
3.1 Definição	10
3.2 Desafios	11
4 Instanciação da Arquitetura de Data Warehousing	11
4.1 Pipelines para Volumes de Dados Tradicionais	12
4.2 Pipelines para Gigantescos Volumes de Dados	13
4.3 Pipelines para <i>Data Streaming</i> de Gigantescos Volumes de Dados	15
5 Conclusão	16



1 ARQUITETURA DE DATA WAREHOUSING

A Figura 1 ilustra uma visão geral da arquitetura de *data warehousing*, cujo detalhamento é descrito a seguir.

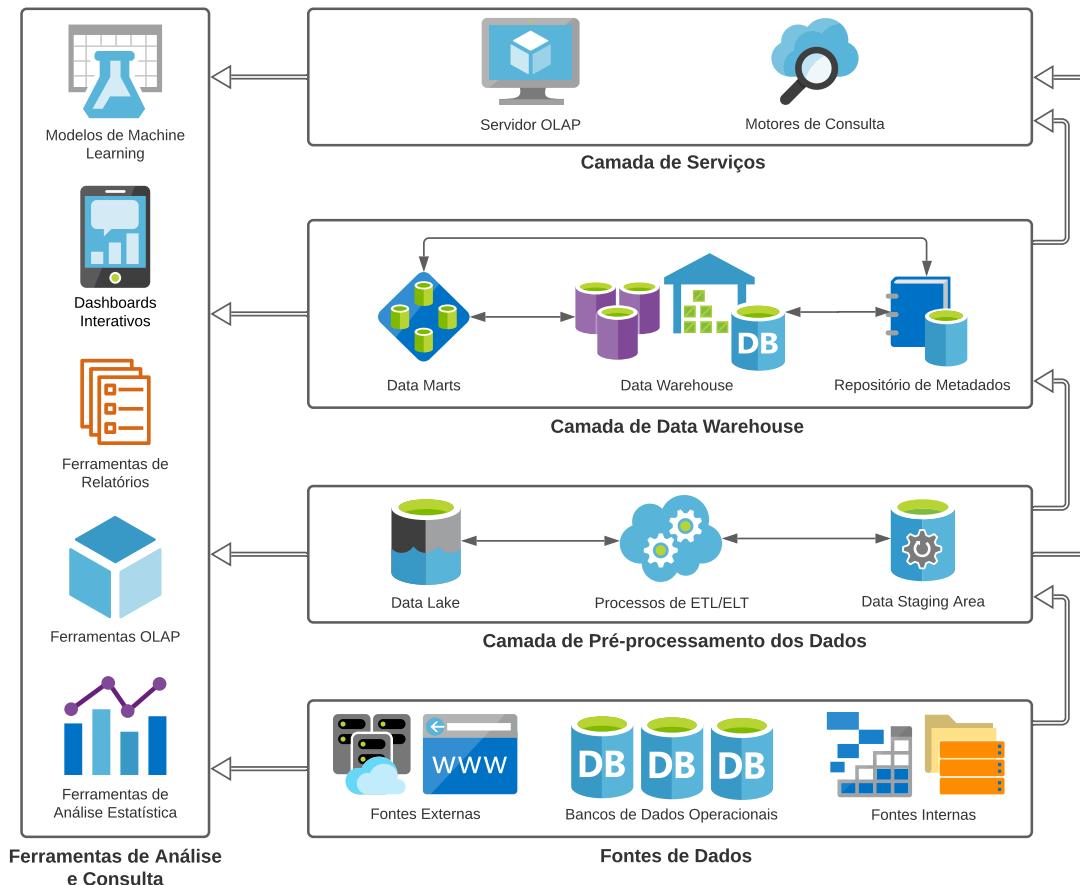


Figura 1: Visão geral da arquitetura do *data warehousing*.

1.1 FONTES DE DADOS

As **fontes de dados** contêm os dados operacionais. Elas são caracterizadas por serem autônomas, heterogêneas e distribuídas. O termo autônoma refere-se ao fato de que as fontes de dados foram desenvolvidas de forma independente, sem a perspectiva de fornecer seus dados ao *data warehousing*.

Ademais, o termo heterogênea indica que as fontes de dados podem possuir uma variedade de formatos e modelos. Exemplos de fontes heterogêneas de dados incluem:

- Sistemas gerenciadores de banco de dados (SGBDs) relacionais, orientados a objetos e objeto-relacionais.
- Bases de conhecimento e bases de dados NoSQL (*Not Only SQL*).
- Sistemas legados baseados em modelos hierárquicos e de rede.



- Documentos HTML (*Hypertext Markup Language*) e SGML (*Standard Generalized Markup Language*).
- Planilhas e arquivos.

Quanto ao termo distribuída, ele indica que as fontes de dados encontram-se usualmente localizadas em diferentes ambientes computacionais ou servidores.

1.2 CAMADA DE PRÉ-PROCESSAMENTO DOS DADOS

A **camada de pré-processamento dos dados** tem como funcionalidade possibilitar a preparação dos dados para que eles possam ser posteriormente armazenados no *data warehouse* (DW). Componentes dessa camada são processos de ETL/ELT, *data staging area* e *data lake*.

No processo **ETL/ELT**, os dados relevantes das fontes de dados são extraídos, traduzidos, filtrados e integrados para serem posteriormente armazenados no DW. Esse processo também é responsável por realizar a atualização periódica do DW de forma a refletir as alterações nos dados das fontes e realizar a expiração de dados antigos armazenados no DW.

Considerando a evolução histórica de *data warehousing*, primeiramente surgiu o termo ETL (*extract, transform, load*), traduzido como extração, transformação e carga. Desde que o processo de ETL é demorado e custoso, a *data staging area* passou a ser usada como uma área de armazenamento intermediária. Essa área de armazenamento contém os dados das fontes de dados que vão passando por sucessivas modificações até que estejam prontos para serem carregados no DW [11].

Com o avanço tecnológico, principalmente advindo da manipulação de gigantescos volumes de dados relacionados ao conceito de *big data* [3], surgiu o termo ELT (*extract, load, transform*). Isso indica que primeiro os dados precisam ser extraídos e carregados no *data lake*, para depois continuarem a ser processados e finalmente armazenados no DW. Nesse sentido, o *data lake* atua como uma área de armazenamento que contém um grande volume de dados estruturados, semiestruturados e não estruturados e que são processados somente quando a informação precisa ser obtida [5]. Desde que o DW armazena dados estruturados, semiestruturados e não estruturados, costuma-se caracterizar que o *data lake* armazena dados em seu formato nativo. Esse termo, formato nativo, é usado ao longo deste texto. Entretanto, na prática, quando se tem dados tabulares ou aninhados, é usualmente comum transformá-los em um formato mais adequado para serem explorados no *data lake* e também para serem processados quando do armazenamento no DW.

1.3 CAMADA DE DATA WAREHOUSE

Na **camada de data warehouse** são armazenados os dados que já passaram pelos processos providos pela camada de pré-processamento, bem como os metadados associados. Componentes desta camada são DW, *data marts* e repositório de metadados.



O DW, considerado o coração do ambiente de *data warehousing*, é um banco de dados voltado para o suporte aos processos de gerência e tomada de decisão. Ele armazena dados estruturados, os quais são organizados multidimensionalmente, ou seja, são organizados de acordo com as diferentes perspectivas de análise dos usuários de sistemas de suporte à decisão (usuários de SSD) [4].

Em adição ao DW, podem existir diversos *data marts*, cada um dos quais representando um pequeno DW que possui escopo limitado quando comparado ao DW propriamente dito. Os dados armazenados nos *data marts* compartilham as mesmas características que os dados do DW, ou seja, são dados organizados multidimensionalmente.

Outro componente desta camada é o **repositório de metadados**. Ele armazena os metadados de todos os dados e processos envolvidos no *data warehousing*. O conceito de metadados refere-se ao fato de que dados de nível mais baixo podem ser descritos por dados de nível mais alto. Metadados consistem em uma abstração que provê significado semântico aos dados. Por exemplo, metadados associados a uma sequência de 0s e 1s devem indicar se tais caracteres representam palavras, ou números, ou dados estatísticos sobre salários de empregados, ou ainda informações sobre a distribuição de cargos na empresa.

O armazenamento de metadados no *data warehousing* possui um nível de importância elevado, uma vez que é necessário conhecer a estrutura e o significado dos dados presentes em todos os processos envolvidos. Em outras palavras, metadados constituem-se no principal recurso para a administração dos dados no *data warehousing*. Portanto, uma grande variedade de metadados precisa ser armazenada, visando a utilização efetiva do *data warehousing*.

1.4 CAMADA DE SERVIÇOS

Componentes presentes na **camada de serviços** devem oferecer funcionalidades adequadas para facilitar o acesso aos dados com o objetivo de dar suporte à tomada de decisão estratégica.

Servidores OLAP (*on-line analytical processing*) proveem visões multidimensionais dos dados do DW ou dos *data marts*, independentemente da forma na qual os dados encontram-se armazenados [11]. Uma visão multidimensional permite a visualização dos dados sob diferentes perspectivas, de acordo com as necessidades dos usuários de suporte à decisão. É importante destacar que o servidor OLAP está relacionado ao conceito de hipercubo de dados multidimensional, ou seja, à metáfora do cubo de dados. Esse conceito está atrelado ao nível conceitual do modelo de dados multidimensional [4].

Os **motores de consulta** oferecem serviços que têm como funcionalidade executar consultas contra dados armazenados em bancos de dados. Por exemplo, um motor de consulta SQL (*structured query language*) executa consultas nessa linguagem de programação contra dados armazenados no modelo relacional.



1.5 CAMADA DE FERRAMENTAS DE ANÁLISE E CONSULTA

O principal propósito de um *data warehousing* consiste em disponibilizar informação integrada aos usuários de SSD para a tomada de decisão estratégica. Esses usuários interagem com o ambiente por meio de **ferramentas de análise e consulta** dos dados, as quais devem oferecer facilidades de navegação e de visualização. Em especial, essas ferramentas devem permitir que informações relevantes ao contexto de tomada de decisão sejam derivadas a partir da detecção de análise de tendências, da monitoração de problemas e de análises competitiva e comparativa.

Dentre os principais tipos de ferramentas existentes, pode-se citar:

- **Ferramentas de consulta gerenciáveis e geradores de relatório.** São os tipos mais simples de ferramentas e, em geral, não são voltadas especificamente ao *data warehousing*. Geradores de relatório, como o próprio nome diz, têm como principal objetivo produzir relatórios periódicos. Já ferramentas de consulta gerenciáveis oferecem aos usuários visões de negócio específicas ao domínio dos dados armazenados e permitem que esses usuários realizem consultas independentemente da estrutura e/ou da linguagem de consulta oferecida pelo banco de dados. Por exemplo, uma ferramenta desse tipo poderia permitir a criação de comandos SQL por meio da utilização de um conjunto de opções. A saída destas ferramentas é geralmente na forma de um relatório.
- **Ferramentas de análise estatística.** Essas ferramentas permitem que os usuários de SSD analisem os dados usando métodos estatísticos.
- **Dashboards interativos.** Um *dashboard* é uma ferramenta que reúne diversos dados e indicadores por meio de gráficos e tabelas, permitindo o monitoramento simultâneo de um grande número de informações, as quais são visualizadas com facilidade em um único ambiente. *Dashboards* possibilitam interações de detalhamento ou agrupamento de gráficos, indicadores e tabelas, dentre outros.
- **Ferramentas OLAP.** São caracterizadas por permitir que usuários de SSD sofisticados analisem os dados usando visões multidimensionais complexas e elaboradas, e por oferecer navegação facilitada pelas diferentes visões. Assim, os usuários podem analisar os dados sob diferentes perspectivas e/ou determinar tendências por meio da navegação entre diferentes níveis de agregação. Tais ferramentas apresentam os dados de acordo com o modelo multidimensional [2, 7, 4], independentemente da forma na qual eles estão realmente armazenados.
- **Modelos de machine learning.** De maneira geral, as ferramentas de análise e consulta possuem duas funcionalidades básicas: facilitar o acesso aos dados do DW e permitir que



informações, tendências e padrões de negócio “escondidos” nesses dados sejam descobertos. Modelos de *machine learning* vislumbram a segunda funcionalidade. Nesse sentido, aplicações de *machine learning* processam os dados para descobrir esses padrões escondidos e mostrá-los aos usuários de SSD, os quais podem inferir conhecimento útil a partir destes. A qualidade do conhecimento descoberto é altamente dependente da aplicação e tem um aspecto subjetivo inerente.

Independentemente do tipo de ferramenta utilizada, um fator primordial a ser considerado refere-se à **visualização** dos resultados obtidos. Técnicas de visualização dos dados devem determinar a melhor forma de se exibir relacionamentos e padrões complexos, de modo que o problema inteiro e a solução sejam claramente visíveis. Por exemplo, padrões podem ser mais facilmente detectados se forem expressos graficamente, melhor do que por meio de simples tabelas. Em especial, técnicas de visualização devem oferecer interação com os usuários de SSD, os quais devem ser capazes de alterar tanto o tipo de informação sendo analisada quanto o método de apresentação sendo utilizado (como histogramas, mapas hierárquicos e gráficos de dispersão).

É importante destacar que a visualização dos dados pode não ser a atividade fim das análises realizadas sobre os dados do DW ou, principalmente, sobre os dados do *data lake*. Existe a possibilidade de que as métricas extraídas por meio dessas análises alimentem automaticamente as aplicações que atuam como fontes de dados. Por exemplo, suponha uma aplicação de *data warehousing* de e-commerce. Suponha também que os dados armazenados no DW ou no *data lake* tenham sido utilizados para treinar um modelo de aprendizado de máquina. Como resultado, o conhecimento obtido pode retornar como uma recomendação para a página de e-commerce.

2 DIFERENÇAS ENTRE LOCAIS DE ARMAZENAMENTO

Excluindo as fontes de dados, a arquitetura de *data warehousing* (Figura 1) pode englobar quatro locais de armazenamento de dados: DW, *data mart*, *data lake* e *data staging area*. A seguir são destacadas as diferenças existentes entre esses locais de armazenamento.

2.1 DATA WAREHOUSE E DATA MARTS

Como visto anteriormente, um *data mart* consiste na implementação de um DW no qual o escopo do dado é limitado quando comparado ao DW propriamente dito. Entretanto, os dados armazenados em *data marts* compartilham as mesmas características que os dados do DW, ou seja, são orientados a assunto, integrados, históricos e não voláteis, além de serem organizados em níveis de agregação.



É necessário, portanto, discutir a importância dos *data marts* dentro do *data warehousing*. Em uma grande corporação, *data marts* tendem a ser utilizados como uma política de construção evolucionária do DW. Uma vez que o processo de construção de um DW sobre toda a organização é longo e complexo e os custos envolvidos são altos, *data marts* são construídos paulatinamente e, à medida que estes se consolidam, inicia-se a construção do DW corporativo. De maneira geral, tais *data marts* representam soluções fragmentadas de porções de negócio da empresa, sendo chamados de *data marts* independentes.

A utilização de *data marts* independentes tende, inicialmente, a reduzir problemas financeiros. Isso se deve ao fato de que a construção desses *data marts* exige recursos monetários inferiores do que os despendidos com a construção de um DW corporativo, fazendo com que os usuários de SSD sejam capazes de reconhecer o valor e a potencialidade da solução de *data warehousing* em um período menor de tempo. Entretanto, em longo prazo, a criação de *data marts* independentes pode conduzir a problemas de integração, caso um modelo de negócio completo não seja desenvolvido. Isso se deve ao fato de que cada *data mart* independente pode assumir formas diferentes de consolidar seus dados, gerando inconsistências.

2.2 DATA STAGING AREA E DATA LAKE

A *data staging area* contém dados extraídos das fontes de dados que vão passando por modificações sucessivas até que estejam prontos e que possam ser carregados no DW [11]. Ela consiste em uma área de armazenamento intermediária para a qual não se prevê acesso pelos componentes da camada de ferramentas de análise e consulta. Portanto, o fluxo de dados é no sentido *data staging area* → DW. Conforme descrito anteriormente, a *data staging area* é decorrente historicamente do processo de ETL.

O *data lake* contém um grande volume de dados extraídos das fontes de dados em seu formato nativo (*raw data*), incluindo dados estruturados, semiestruturados e não estruturados. Esses dados são processados somente quando a informação precisa ser obtida [5]. Existem dois fluxos de dados quando se trata de *data lake*. O primeiro deles é no sentido *data staging area* → DW, significando que o *data lake* pode atuar também como uma *data staging area* para a carga de dados no DW. O segundo fluxo de dados é no sentido *data lake* → componentes da camada de ferramentas de análise e consulta, indicando que os dados do *data lake* também podem ser usados para a descoberta de novas informações ou para a geração de valor a partir desses [8]. Isso significa que as consultas e análises podem ser realizadas diretamente sobre o *data lake*, sem a necessidade de se usar os dados do DW. Esse fluxo é necessário, por exemplo, quando se deseja analisar dados de *streaming*. Conforme descrito anteriormente, o *data lake* é decorrente historicamente do processo de ELT.



2.3 DATA WAREHOUSE E DATA LAKE

Na Tabela 1 são contrastadas diferenças existentes entre o DW e o *data lake*, considerando os seguintes aspectos relacionados aos dados: característica, formato, pré-processamento, tipos de consulta, latência de disponibilidade dos dados, custo de geração dos dados e custo de análise dos dados [1]. Essas diferenças são detalhadas a seguir.

Tabela 1: Comparativo entre as características do DW e do *data lake*, considerando aspectos relacionados aos dados.

	Data Warehouse	Data Lake
Característica	consolidados, organizados e estruturados	estruturados, semiestruturados e não estruturados
Formato	esquema estruturado (formato bem definido)	formato nativo (diferentes formatos)
ETL/ELT	dados pré-processados antes de serem carregados	dados extraídos e carregados, sem sofrer transformações
Tipos de Consulta	OLAP	variado
Latência de disponibilidade dos dados	alta	baixa
Custo de Geração	maior	menor
Custo de Análise	menor	maior

Enquanto o DW contém dados consolidados e organizados que já passaram pelo processo de ETL e que são armazenados segundo um esquema estruturado bem definido, o *data lake* deve oferecer suporte a vários formatos de dados, facilitando a aquisição dos dados das fontes de dados para prover agilidade.

Para povoar o DW, os dados precisam primeiro passar pelo processo de ETL. Em contrapartida, os dados armazenados no *data lake* são decorrentes do processo de ELT, ou seja, eles são extraídos das fontes de dados e carregados no *data lake*, sem passar por processos de transformação, os quais ocorrem somente quando necessário. O esforço necessário para extrair e carregar os dados é reduzido porque os dados não são pré-processados.

Com relação aos tipos de consulta, o DW é um banco de dados especialmente projetado para oferecer suporte eficiente ao processamento de consultas analíticas, ou seja, consultas



OLAP (*on-line analytical processing*). O *data lake*, por sua vez, oferece suporte para os mais variados tipos de consulta.

A latência de disponibilidade dos dados, o custo de geração e o custo de análise são decorrentes do pré-processamento aplicado aos dados. Devido ao processo ETL, os dados do DW demoram para serem processados e demandam processos muito custosos. Portanto, possuem latência alta e maior custo de geração dos dados. Em contrapartida, requerem menor custo relacionado à análise dos dados, desde que os dados já encontram-se preparados para serem utilizados na tomada de decisão estratégica. No caso dos dados do *data lake*, a situação é inversa. Devido ao processo ELT, os dados são extraídos e já armazenados no *data lake*, garantindo uma latência baixa de disponibilidade dos dados e um custo de geração dos dados menor. Em contrapartida, os dados precisam ser transformados e integrados para serem usados nas análises dos usuários de SSD, requerendo custos maiores para o processamento de consultas analíticas.

3 BIG DATA

Conforme descrito na arquitetura de *data warehousing*, o processo ELT e o *data lake* surgiram em resposta aos gigantescos volumes de dados relacionados ao conceito de *big data*. Nesta seção são descritos a definição de *big data* e os desafios impostos por esses.

3.1 DEFINIÇÃO

As definições de *big data* mais usadas são baseadas no modelo de diferentes Vs, sendo que na literatura existem trabalhos que definem 3Vs [3], 4Vs [6], 5Vs [9] e 7Vs [12]. O modelo de 7Vs é definido da seguinte forma.

- Volume: gigantesca quantidade de dados, a qual atualmente varia de *terabytes* a *exabytes*.
- Velocidade: captura e disponibilidade de um gigantesco volume de dados em um pequeno intervalo de tempo.
- Variedade: dados podem ser de qualquer tipo, incluindo dados semiestruturados e não-estruturados como áudio, vídeo, páginas web e texto, além de dados estruturados.
- Veracidade: quão confiáveis são os dados, considerando aspectos como abrangência, consistência, precisão e atualidade.
- Valor: os dados devem ter importância dentro do contexto da aplicação, de forma que justifique a necessidade de manipulação desses dados.
- Variabilidade: os valores dos dados e seus significados podem variar constantemente.
- Visualização: exibição apropriada dos dados volumosos.



3.2 DESAFIOS

A manipulação de *big data* introduz vários desafios [3]. O primeiro deles se refere ao uso de **ambientes computacionais com grande capacidade de armazenamento e processamento**, tais como clusters de computadores e ambientes de computação em nuvem (*cloud computing*). Esses ambientes exigem que diversos aspectos sejam considerados para que o processamento de grandes volumes de dados ocorra de forma otimizada, o que muitas vezes reflete em tarefas não triviais.

O segundo desafio consiste no uso de **frameworks de processamento paralelo e distribuído de dados**, os quais surgiram para simplificar a interação do usuário com os ambientes computacionais descritos anteriormente por meio do provimento de uma interface simplificada de programação de aplicações. Exemplos amplamente utilizados são Apache Hadoop¹ e Apache Spark².

O terceiro desafio refere-se ao uso de **sistemas de arquivos distribuídos**, dentre os quais destaca-se o HDFS (*Hadoop Distributed File System*) [10]. Ele provê suporte para o armazenamento de grandes quantidades de dados e possui alta tolerância a falhas. Adicionalmente, HDFS é capaz de ser empregado também em equipamentos de *hardware* de baixo custo. Ambos Apache Hadoop e Apache Spark utilizam o HDFS como sistema de arquivos distribuídos padrão.

Por fim, em adição ao uso de alguns SGBDs relacionais, pode-se destacar o quarto desafio como a possibilidade de se usar **bases de dados NoSQL**. Eles são caracterizados por serem baseados em diferentes formatos, usualmente não relacionais, e por garantirem alta escalabilidade. Adicionalmente, eles introduzem flexibilidade no armazenamento de diferentes tipos de dados, como dados não estruturados, semiestruturados e estruturados.

O detalhamento desses desafios não é o objetivo do presente texto. Os desafios, bem como a definição de *big data*, foram introduzidos neste texto para dar suporte à discussão de arquiteturas instanciadas por meio de tecnologias. A descrição detalhada dos desafios será realizada posteriormente.

4 INSTANCIAMENTO DA ARQUITETURA DE DATA WAREHOUSING

A arquitetura ilustrada na Figura 1 mostra todos os componentes de um *data warehousing*. Entretanto, nem todos precisam estar presentes no desenvolvimento de uma aplicação voltada à tomada de decisão estratégica. A escolha de quais componentes devem participar do *data warehousing* depende do propósito da aplicação: para o que ela serve, quais seus objetivos e quais

¹<https://hadoop.apache.org/>

²<https://spark.apache.org/>



componentes são capazes de oferecer suporte para as demandas impostas pela aplicação.

Nesta seção são exemplificadas arquiteturas de *data warehousing* instanciadas por meio de tecnologias, as quais são usualmente chamadas no mercado de trabalho de *pipelines*. Note que o objetivo é mostrar exemplos, sem ser exaustivo. Ou seja, não são listadas todas as tecnologias existentes nem todas as possíveis instanciações que podem ocorrer. Também são considerados diferentes tipos de tecnologias, incluindo soluções de *software livre* e produtos pagos.

Em todas as arquiteturas ilustradas nesta seção, os dados são armazenados no DW para propósito ilustrativo. Porém, eles poderiam estar armazenados em um ou mais *data marts* de acordo com as discussões realizadas na seção 2.3.

4.1 PIPELINES PARA VOLUMES DE DADOS TRADICIONAIS

Na Figura 2 é ilustrado um *pipeline* para uma aplicação de *data warehousing* tradicional de processamento de dados em lotes. Como exemplos de fontes de dados, tem-se um SGBD relacional, uma base de dados no formato JSON (*JavaScript Object Notation*) e uma planilha Excel. Nessa proposta de *pipeline*, os dados das fontes operacionais são extraídos, transformados e carregados (ETL) na *data staging area* usando Pandas³. Uma vez na *data staging area*, que é construída no SGBD relacional MySQL⁴, os dados passam por diversas transformações até se adequarem à forma de organização do DW. Na sequência, os dados são extraídos da *data staging area* e carregados no DW, também construído no MySQL. Desde que ambos *data staging area* e DW utilizam a tecnologia relacional para o armazenamento dos dados, utiliza-se a linguagem de programação SQL⁵ para a movimentação dos dados entre esses locais de armazenamento. O *pipeline* inclui também duas tecnologias de análise e visualização dos dados do DW: a ferramenta de construção de *dashboards* interativos Metabase⁶ e uma aplicação Python⁷ para a geração de relatórios analíticos.

O *pipeline* ilustrado na Figura 3 mostra a implementação no ambiente de nuvem da AWS⁸ (*Amazon Web Services*) para a mesma arquitetura do *pipeline* anterior (Figura 2). Para o processo de ETL utiliza-se o AWS Lambda⁹, uma ferramenta de computação em nuvem sem servidor (*serverless*) de baixo custo que permite a execução de códigos em linguagens de programação como Python e JavaScript, dentre outras. Ambos *data staging area* e DW são construídos usando o serviço de provisionamento de bancos de dados relacionais em nuvem da AWS chamado de AWS RDS¹⁰ (*Relational Database Service*). Por fim, a aplicação Python para geração de relatórios analíticos do *pipeline* utiliza o mesmo serviço do processo de ETL, ou seja, AWS

³<https://pandas.pydata.org/>

⁴<https://www.mysql.com/>

⁵<https://www.iso.org/standard/63555.html>

⁶<https://www.metabase.com/docs/latest/users-guide/07-dashboards.html>

⁷<https://www.python.org/>

⁸<https://aws.amazon.com/>

⁹<https://aws.amazon.com/lambda/>

¹⁰<https://aws.amazon.com/rds/>



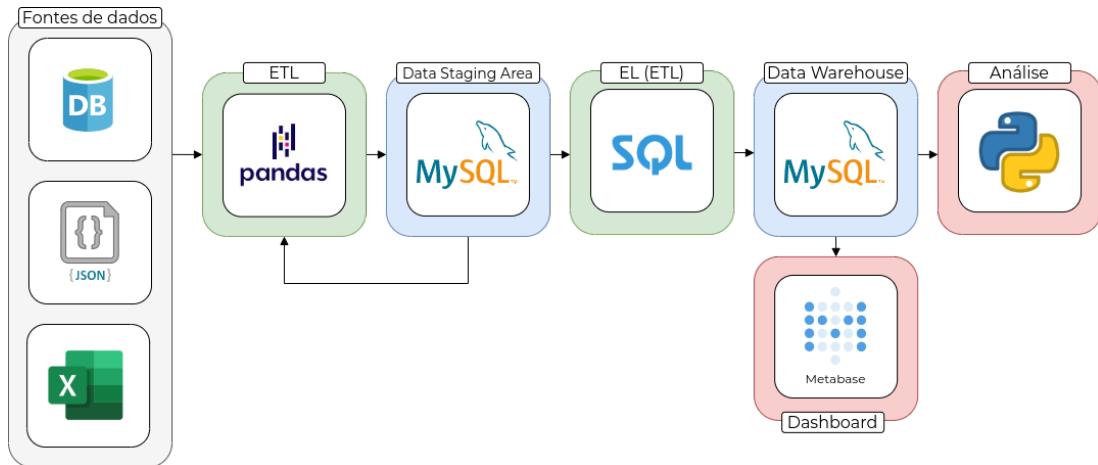


Figura 2: Pipeline de processamento de dados em lotes.

Lambda. Com relação à ferramenta geradora de *dashboards* interativos Tableau¹¹, esta usa seu próprio serviço de nuvem para visualizar os dados do DW no ambiente da AWS.

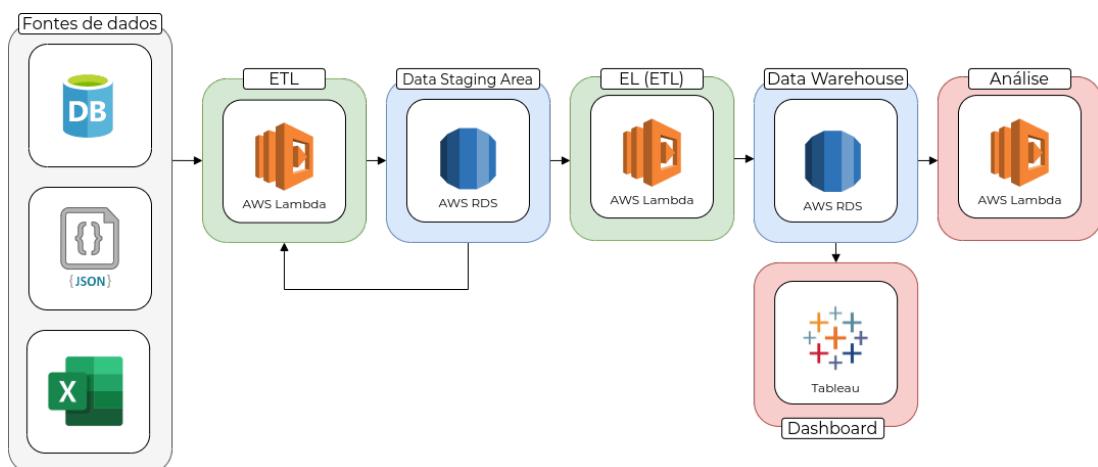


Figura 3: Pipeline de processamento de dados em lotes na nuvem.

4.2 PIPELINES PARA GIGANTESCOS VOLUMES DE DADOS

No contexto de *big data*, é proposto o *pipeline* ilustrado na Figura 4. Os dados de interesse das fontes de dados são extraídos e carregados em um *data lake* pelo Apache Spark. No *data lake*, os dados são explorados pelo motor de consulta Apache Hive¹². Na sequencia, os dados passam por vários processos de transformação usando o Apache Spark até que se adequarem à forma de organização do DW. Quando prontos, esses dados são carregados no DW construído utilizando o Apache Druid¹³, o qual também atua como um servidor OLAP no que tange às consultas analíticas. Por fim, usuários de SSD podem realizar análises e criar *dashboards* utilizando o Metabase.

¹¹<https://www.tableau.com/>

¹²<https://hive.apache.org/>

¹³<https://druid.apache.org/>



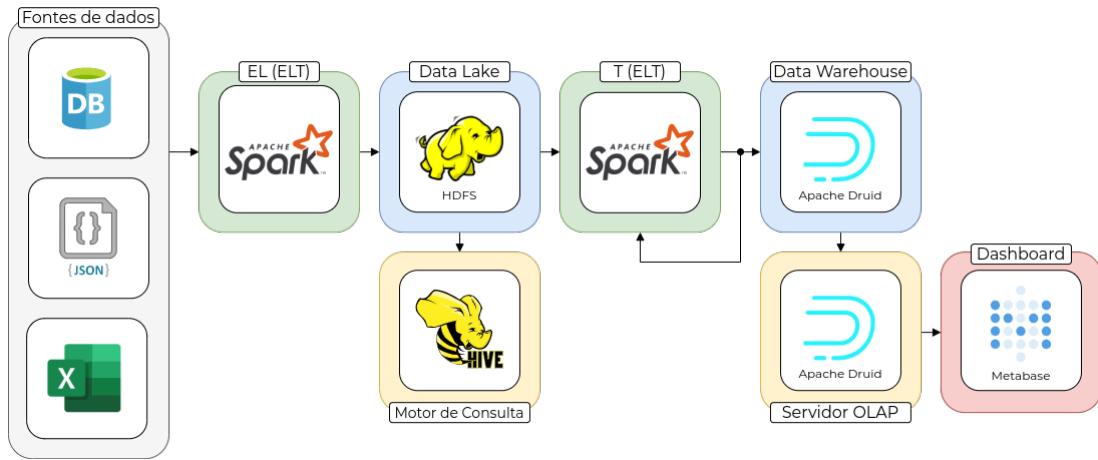


Figura 4: Pipeline de processamento de *big data* em lotes.

Na Figura 5 é ilustrada a implementação do *pipeline* da arquitetura anterior (Figura 4) no ambiente de nuvem da AWS usando as seguintes tecnologias: (i) AWS EMR¹⁴ (*Elastic Map Reduce*) para a extração e o carregamento dos dados das fontes de dados no *data lake* e para a transformação dos dados do *data lake* para o DW; (ii) AWS S3¹⁵ (*Simple Storage Service*) para o armazenamento dos dados no *data lake*; (iii) AWS Athena¹⁶ como motor de consulta para a exploração dos dados armazenados no *data lake*; e (iv) AWS Redshift¹⁷ como uma solução de armazenamento dos dados no DW e como servidor OLAP para esses dados. Usuários de SSD utilizam o Tableau, em sua própria infraestrutura em nuvem, como ferramenta de análise e criação de dashboards interativos.

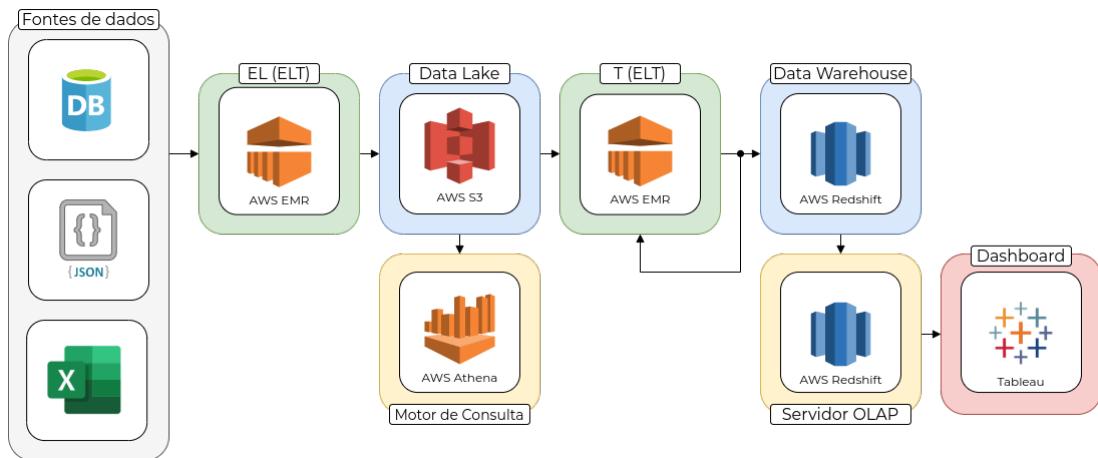


Figura 5: Pipeline de processamento de *big data* em lotes na nuvem.

¹⁴<https://aws.amazon.com/emr/>

¹⁵<https://aws.amazon.com/s3/>

¹⁶<https://aws.amazon.com/athena/>

¹⁷<https://aws.amazon.com/redshift/>

4.3 PIPELINES PARA DATA STREAMING DE GIGANTESCOS VOLUMES DE DADOS

Existe um conjunto de fontes de dados que produzem dados de maneira contínua (*data stream*) e que necessitam de monitoramento e extração de conhecimento em tempo (quase) real. Neste contexto, uma arquitetura de processamento de dados em lotes, como as ilustradas nas Figuras 2 a 5, pode não ser adequada. Por exemplo, considere um modelo de aprendizado de máquina que realiza uma recomendação a um usuário de um e-commerce enquanto ele navega pelas suas páginas. A aplicação de *data warehousing* que oferece suporte para esse modelo deve ser capaz de extrair e prover dados rapidamente. Por exemplo, no primeiro semestre de 2019, o tempo médio de navegação de um usuário dessa categoria de website foi de 4 minutos e 12 segundos¹⁸. Portanto, os dados precisam navegar por todo o *pipeline* e retornar a recomendação durante este intervalo de tempo para atender à demanda dessa aplicação.

Na Figura 6 é introduzido um exemplo de *pipeline* de processamento de dados de *data streaming* de *big data*. Como exemplos de fontes de dados, tem-se um website, uma aplicação executando em um aparelho *mobile* e dados de IoT (*Internet of Things*). Os dados das fontes de dados são enviados em fluxo contínuo para o Apache Kafka¹⁹ e são extraídos, transformados e carregados no DW, em tempo (quase) real, usando a solução de *streaming* do Apache Spark. Os dados do DW são armazenados no Apache Druid. Por fim, a combinação do servidor OLAP do Apache Druid com a ferramenta de análise Metabase proporciona aos usuários de SSD *dashboard* interativo com métricas em tempo (quase) real dos dados do DW.

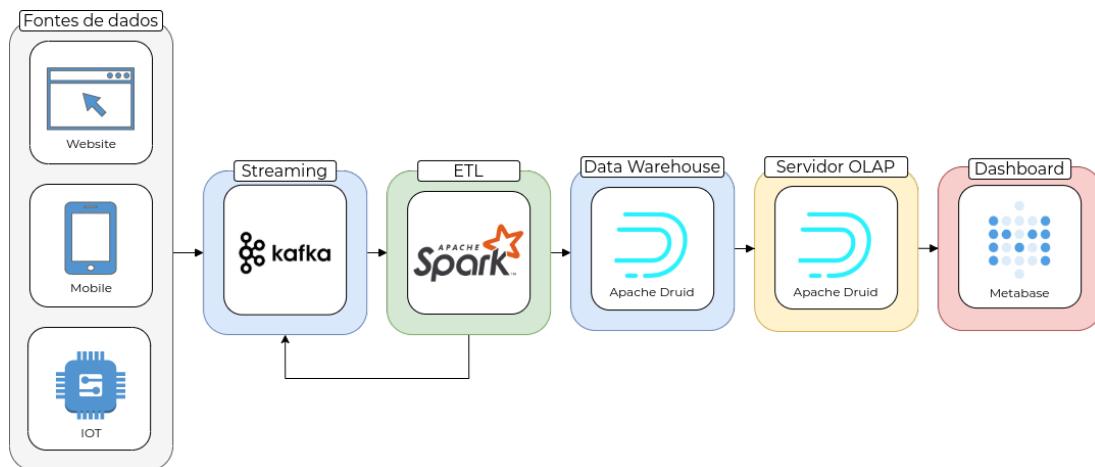


Figura 6: Pipeline de processamento de *data streaming* de *big data*.

De maneira análoga, o *pipeline* da Figura 7 propõe a implementação do *pipeline* anterior (Figura 6) no ambiente de nuvem da AWS. Os dados são extraídos, transformados e carregados no DW, armazendo no AWS Redshift, em fluxo contínuo pelo par de tecnologias AWS Kinesis²⁰

¹⁸<https://www.salesforce.com/solutions/industries/retail/shopping-index/>

¹⁹<https://kafka.apache.org/>

²⁰<https://aws.amazon.com/kinesis/>



e AWS EMR. Já o servidor OLAP do AWS Redshift e o Tableau (este em sua própria infraestrutura de nuvem) proporcionam aos usuários de SSD um *dashboard* interativo com métricas em tempo (quase) real dos dados do DW.

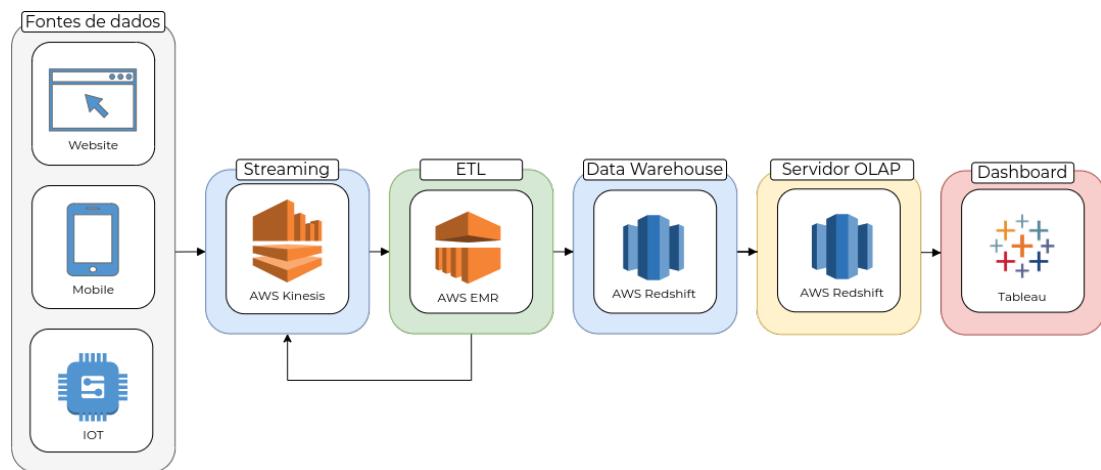


Figura 7: Pipeline de processamento de *data streaming* de *big data* na nuvem.

5 CONCLUSÃO

Neste texto, foram descritos os seguintes conceitos e aspectos relacionados:

- Arquitetura de *data warehousing*: fontes de dados, camada de pré-processamento dos dados, camada de DW, camada de servidores de aplicação e camada de ferramentas de análise e consulta.
- Diferenças entre os locais de armazenamento de dados: DW e *data marts*, *data staging area* e *data lake* e DW e *data lake*.
- *Big data*: definição e principais desafios.
- Instanciação da arquitetura de *data warehousing*: exemplificação de diferentes *pipelines* considerando aplicações de *data warehousing* tradicionais, no contexto de *big data* e no contexto de *data streaming*.

Referências

- [1] J. J. Brito. *Data Warehouses in the era of Big Data: efficient processing of Star Joins in Hadoop*. PhD thesis, Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional, Instituto de Ciências Matemáticas e da Computação, Universidade de São Paulo, 2017.
- [2] S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *SIGMOD Record*, 26(1):65–74, 1997.
- [3] M. Chen, S. Mao, , and Y. Liu. Big data: A survey. *Mobile Networks and Applications*, 19(2):171–209, 2014.
- [4] C. D. A. Ciferri, R. R. Ciferri, L. Gómez, M. Schneider, A. Vaisman, and E. Zimányi. Cube algebra: A generic user-centric model and query language for OLAP cubes. *Journal of Data Warehousing and Mining*, 9(2):39–65, 2013.
- [5] J. Couto, O. Borges, D. Ruiz, S. Marczak, and R. Prikladnicki. A mapping study about data lakes: An improved definition and possible architectures. In *Proceedings of the 31st International Conference on Software Engineering and Knowledge Engineering*, pages 453–458, 2019.
- [6] X. L. Dong and D. Srivastava. Big data integration. *Proceedings of the VLDB Endowment*, 6(11):1188–1189, 2013.
- [7] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. Wiley, 2nd edition, 2002.
- [8] C. Mathis. Data lakes. *Datenbank-Spektrum*, 17(3):289–293, 2017.
- [9] S. Sharma and V. Mangat. Technology and trends to handle big data: Survey. In *Proceedings of the Fifth International Conference on Advanced Computing & Communication Technologies*, pages 266–271, 2015.
- [10] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. The Hadoop distributed file. In *Proceedings of the IEEE 26th Symposium on Mass Storage Systems and Technologies*, pages 1–10, 2015.
- [11] A. Vaisman and E. Zimányi. *Data Warehouse Systems: Design and Implementation*. Springer, 2014.
- [12] R. Wrembel. Novel big data integration techniques: Painel discussion at BigNovelTI 2017@ADBIS2017. 2017.



Análise de Dados com Base em Processamento Massivo em Paralelo

Aula 3: Processo de ETL/ELT

Cristina Dutra de Aguiar
ICMC/USP
cdac@icmc.usp.br

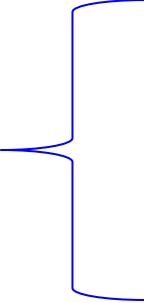


CEPID - Centro de Ciências
Matemáticas Aplicadas à Indústria

Agenda

- Contextualização
- Operações
- Exemplo usando Pandas

Processo de ETL/ELT

- Representa a atividade mais complexa, cara e demorada do *data warehousing*
 - 75% do custo
- Operações
 - Extração (Extract)
 - Transformação (Transform)
 - Carga (Load)
 - Tradução
 - Limpeza
 - Integração

Projeto do Data Warehousing

- Deve ser realizado **antes** do processo de ETL/ELT
 - Indispensável para um bom desenvolvimento do *data warehousing*
- Aspectos a serem considerados
 - **Objetivo** da aplicação de *data warehousing*
 - **Recursos** disponíveis
 - **Hardware** e **software** apropriados
 - Pessoas envolvidas
 - Planejamento da **capacidade** do ambiente

Principais Atividades Envolvidas

- Identificar o **propósito** da aplicação de *data warehousing* e o **volume** de dados manipulado
- Identificar a **arquitetura** de *data warehousing* e seus **componentes**
- **Instanciar** a arquitetura, por meio da integração de servidores, ferramentas e tecnologias
- Realizar o **projeto** do *data warehouse*, dos *data marts* e do *data lake*
- Identificar as **fontes** de dados que possuem dados relevantes
- **Integrar** as fontes de dados ao *data warehousing*

LGPD (Lei Geral de Proteção de Dados)

- Características
 - Introduz diversas **mudanças jurídicas**
 - Proteção de dados pessoais
 - Responsabilidade civil dos responsáveis pelo **tratamento dos dados**
- Possui **grande** impacto no *data warehousing*
 - Coleta, processamento, armazenamento, extração, utilização, modificação, ...

Base das Explicações

- Carga dos dados no *data warehouse*
 - Operações podem ser aplicadas aos *data marts* e ao *data lake*, respeitando-se as particularidades de cada um
- Processo de ETL/ELT como um todo
 - Não considera a abordagem na qual projeta-se apenas as operações do processo de EL para a carga dos dados no *data lake* para pré-exploração dos dados
- Manipulação de grandes volumes de dados
 - Não discute especificamente *big data* e *data streaming*

Processo de ELT e Big Data

- Introduz complexidade adicional
 - A **quantidade** de fontes de dados é muito maior
 - A variedade de **domínios** é muito maior
 - Muitas fontes de dados são **dinâmicas**
 - As fontes de dados são **extremamente heterogêneas** com relação ao formato dos dados
 - Os dados apresentam grande **variabilidade**, dificultando a identificação de mesmas entidades do mundo real presentes em diferentes fontes
 - Os dados das fontes de dados apresentam muita variação de **qualidade**
 - O **tratamento incipiente** do aspecto temporal é muito mais emergente

Diferença entre Instância e Esquema

- Instância
 - Coleção de dados armazenados no banco de dados em um determinado momento, ou seja, são os **dados** propriamente ditos
 - Sinônimos: **extensão** do banco de dados, **linhas** (ou **tuplas**) de tabelas relacionais e **registros** de arquivos
- Esquema
 - **Projeto** do banco de dados, incluindo as entidades e os relacionamentos entre essas entidades
 - Sinônimos: **intenção** do banco de dados, **definição de tabelas** relacionais e **definição da estrutura** (campos) dos registros de arquivos

Aplicação de Data Warehousing da BI Solutions



- Propósito

Foco: **salário** e
quantidadeLançamentos

Perspectivas: funcionário
cargo
filial
data

- Volume de dados

- Pequeno
- Foco em **funcionário**

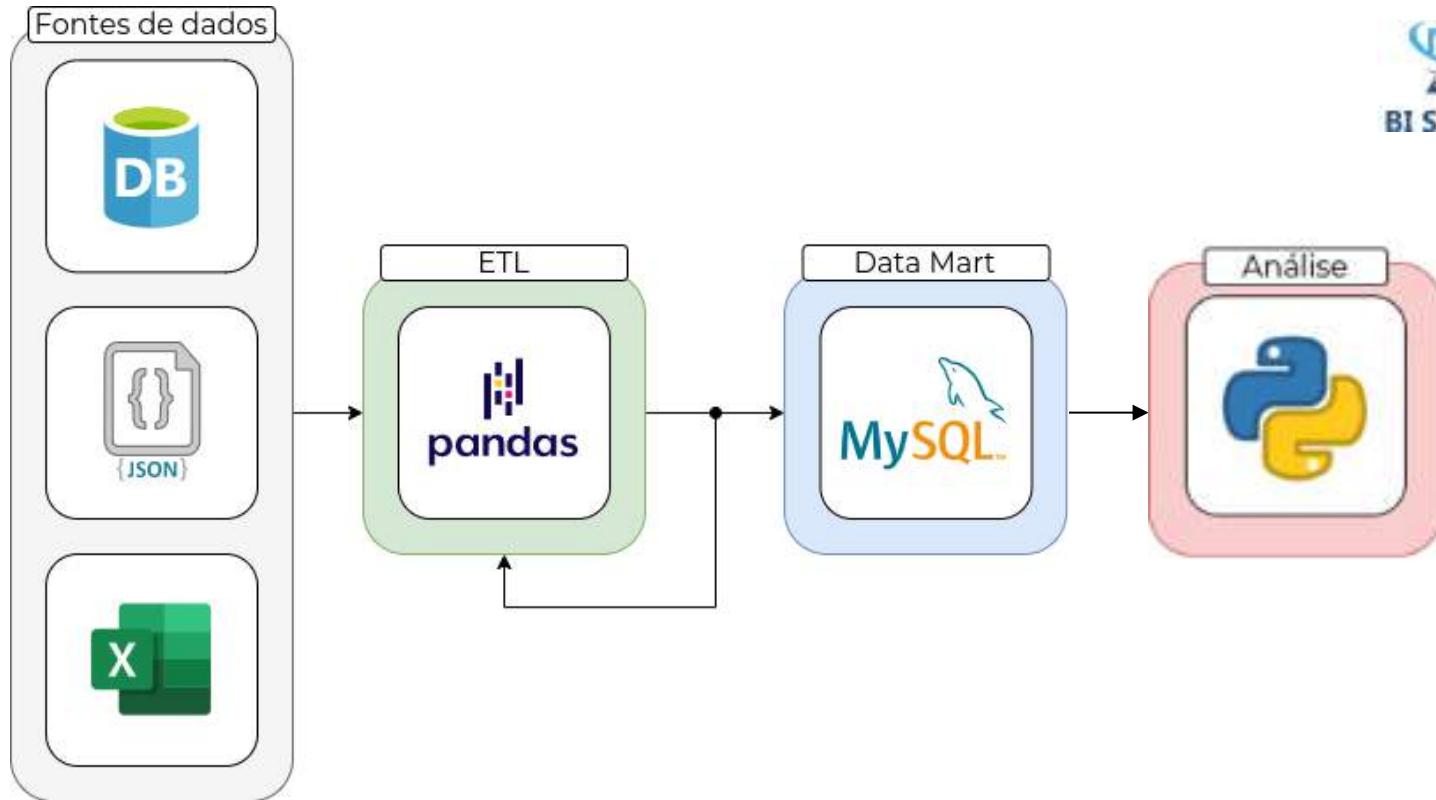
- Arquitetura

- Processamento em **lote**
- Uso de **Data Mart**
- Baseado do **modelo relacional**

- Fontes de dados

- Banco de dados **relacional**
- Arquivo **JSON** (NoSQL)
- Planilha **Excel**

Instanciação da Arquitetura: Pipeline



Projeto do Data Mart: Funcionário



Relação funcionário

funcionario (funcPK, funcMatricula, funcNome, funcSexo, funcDataNascimento,
funcDiaNascimento, funcMesNascimento, funcAnoNascimento,
funcCidade, funcEstadoNome, funcEstadoSigla, funcRegiaoNome,
funcRegiaoSigla, funcPaisNome, funcPaisSigla)

Tabela relacional

funcPK	funcMatricula	funcNome	funcSexo	funcDataNascimento	funcDiaNascimento	funcMesNascimento	...
1	40	Abdiel Lima	M	9/4/1990	09	04	...
2	1	Aline Almeida	F	1/1/1990	01	01	...
...

} esquema
} instância

Fontes de Dados

funcionário (funcMatricula, funcNome, funcSexo, funcDataNasc, funcCidade, funcEstado, funcPaís)

(Fonte 1)
Tabela
Relacional

funcMatricula	funcNome	funcSexo	funcDataNasc	funcCidade	funcEstado	funcPaís
3	ARON ANDRADE	M	03/03/90	Santos	SP	Brasil
10	ADETE CARVALHO	M	10/10/90	Osasco	SP	Brasil
...

} esquema
} instância

(Fonte 2)
Arquivo
JSON

```
[{"colab_matricula":71,"colab_nome":"Ademil Castro","colab_sexo":0,  
"colab_data_nasc":"1970-11-10",  
"colab_cidade":"Belo Horizonte","colab_estado":"MG","colab_pais":"BRA"},  
 {"colab_matricula":72,"colab_nome":"Ademor Costa","colab_sexo":0,"colab_data_nasc":"1971-12-11",  
"colab_cidade":"Araguari","colab_estado":"MG","colab_pais":"BRA"}, ...]
```

(Fonte 3)
Planilha
Excel

	A	B	C	D	E	F
1	Matrícula do Empregado	Nome do Empregado	Sexo do Empregado	Data de Nascimento	Cidade de Residência	Estado de Residência
2						
3	78	Garcia, Adenir	Masculino	17/06/77	Morretes	Parana
4	79	Gomes, Adenil	Masculino	18/07/78	Recife	Pernambuco
5

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

Fontes de Dados

(Fonte 1)
Tabela
Relacional

funcMatricula	funcNome	funcSexo	funcDataNasc	funcCidade	funcEstado	funcPais
3	ARON ANDRADE	M	03/03/90	Santos	SP	Brasil
10	ADETE CARVALHO	M	10/10/90	Osasco	SP	Brasil
...

(Fonte 2)
Arquivo
JSON

```
[{"colab_matricula":71,"colab_nome":"Ademil Castro","colab_sexo":0,  
"colab_data_nasc":"1970-11-10",  
"colab_cidade":"Belo Horizonte","colab_estado":"MG","colab_pais":"BRA"},  
 {"colab_matricula":72,"colab_nome":"Ademor Costa","colab_sexo":0,"colab_data_nasc":"1971-12-11",  
"colab_cidade":"Araguari","colab_estado":"MG","colab_pais":"BRA"}, ...]
```

esquema

instância

(Fonte 3)
Planilha
Excel

	A	B	C	D	E	F
1	Matrícula do Empregado	Nome do Empregado	Sexo do Empregado	Data de Nascimento	Cidade de Residência	Estado de Residência
2						
3	78	Garcia, Adenir	Masculino	17/06/77	Morretes	Parana
4	79	Gomes, Adenil	Masculino	18/07/78	Recife	Pernambuco
5

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

Fontes de Dados

(Fonte 1)
Tabela
Relacional

funcMatricula	funcNome	funcSexo	funcDataNasc	funcCidade	funcEstado	funcPais
3	ARON ANDRADE	M	03/03/90	Santos	SP	Brasil
10	ADETE CARVALHO	M	10/10/90	Osasco	SP	Brasil
...

(Fonte 2)
Arquivo
JSON

```
[{"colab_matricula":71,"colab_nome":"Ademil Castro","colab_sexo":0,  
"colab_data_nasc":"1970-11-10",  
"colab_cidade":"Belo Horizonte","colab_estado":"MG","colab_pais":"BRA"},  
 {"colab_matricula":72,"colab_nome":"Ademor Costa","colab_sexo":0,"colab_data_nasc":"1971-12-11",  
"colab_cidade":"Araguari","colab_estado":"MG","colab_pais":"BRA"}, ...]
```

(Fonte 3)
Planilha
Excel

	A	B	C	D	E	F
1	Matrícula do Empregado	Nome do Empregado	Sexo do Empregado	Data de Nascimento	Cidade de Residência	Estado de Residência
2	78	Garcia, Adenir	Masculino	17/06/77	Morretes	Parana
3	79	Gomes, Adenil	Masculino	18/07/78	Recife	Pernambuco
4
5

} esquema
} instância

Agenda

- Contextualização
- Operações
- Exemplo usando Pandas

Operações

- Extração
- Tradução
- Limpeza
- Integração
- Carga

Extração

- Objetivo
 - Extração dos dados de interesse das fontes
 - Encaminhamento dos dados para as demais operações
- Tarefas
 - Quais dados são extraídos de quais fontes de dados
 - Como esses dados são extraídos
 - Com qual frequência esses dados devem ser periodicamente extraídos
 - Qual técnica empregar para identificar dados das fontes que foram alterados

Tipos de Extração

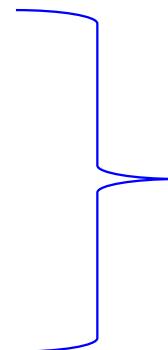
- Extração inicial
 - Carga inicial dos dados no DW
 - Quanto maior o volume de dados, mais tempo é consumido
- Demais extrações
 - Ocorrem devido às alterações nos dados das fontes
 - Técnicas que podem ser aplicadas
 - Extração completa: todos os dados são extraídos e carregados novamente
 - Extração incremental: apenas os dados que sofreram alteração são extraídos e utilizados para determinar o novo conteúdo do DW

Autonomia das Fontes de Dados

- Intervenção mínima
 - Aplicações devem prover seus dados com a **menor intervenção** possível
- Operação não intrusiva
 - Extração **não** pode **impactar negativamente** na execução das aplicações
 - **Janela de manutenção** para processamento em lote
 - Período no qual os sistemas operacionais geralmente ficam mais ociosos
 - **Extração contínua** para processamento de *data streaming*

Exemplos de Abordagens de Extração

- Processo de extração
 - cliente: aplicação que realiza a extração dos dados
 - servidor de dados: fonte de dados
- Abordagens
 - Interface de comandos padronizados
 - Protocolo comum de acesso aos dados
 - Conversor de comandos



podem ser utilizadas conjuntamente ou separadamente

Interface de Comandos Padronizados

- Interface do cliente e do servidor como um elemento comum
 - Interface genérica, ou seja, *Application Programming Interface* ([API](#))
- Aplicação identifica qual a fonte de dados e um **driver** específico **converte** os formatos e comandos
- Exemplos
 - ODBC (*Open Database Connectivity*)
 - DBE (*Borland Database Engine*)
 - JDBC (*Java Database Connectivity*)

Código da aplicação cliente **não precisa ser alterado** quando esta for redirecionada para outro tipo de servidor

Protocolo Comum de Acesso aos Dados

- Protocolo bem definido para conectar aplicações clientes aos vários tipos de servidores
 - Interface **cliente**: recebe requisições de aplicações clientes traduz no protocolo comum
 - Interface **servidora**: identifica e processa as requisições traduz os resultados para o protocolo comum
- Exemplos
 - DRDA (*Distributed Relational Database Architecture*)
 - RDA (*Remote Data Access*)
 - REST (*Representational State Transfer*)

Garante
interoperabilidade entre
tipos de servidores
mesmo que diferentes
APIs tenham sido
usadas

Conversor de Comandos

- Converte comandos de manipulação de dados e de formato de dados (*gateway*)
- Provê a funcionalidade de tradutor
 - Realiza o **mapeamento** de dados e de comandos entre os vários clientes e servidores
- Exemplos
 - *Database Gateway* para DB2
 - *Informix Enterprise Gateway*

Usualmente **complementa** ou **estende** as outras abordagens

Detecção e Propagação de Alterações

- Rotinas
 - Monitorar as **modificações** ocorridas nas fontes de dados
 - **Identificar** quais modificações ocorreram em quais dados
 - Extrair somente os **dados necessários**
- **Frequência** (periodicidade, latência)
 - Depende das necessidades das análises e do nível de consistência desejado
 - Exemplos: assim que o dado é gerado, a cada hora, diária, semanal

Técnicas Empregadas

- Chamadas de *Change Data Capture (CDC)*
- Dependem das facilidades oferecidas pelas fontes de dados
- Abordagens
 - *Timestamp* (marcadores de tempo)
 - *Triggers* (gatilhos)
 - *Logs*
 - *Snapshots* (instantâneos)

Timestamp

- Armazenado em uma **coluna de auditoria**
- CDC
 - **Compara** o *timestamp* com a data e o horário da extração mais recente
 - Extrai dados que possuem **data de alteração maior** do que a **data dessa extração**
- Exemplo
 - Kafka Connect (Confluent)
 - Somente **poucos** dados operacionais possuem *timestamp*
 - Não identifica dados **removidos**
 - Pode ser **intrusiva**

Triggers

- Presentes em sistemas gerenciadores de banco de dados relacionais
 - CDC
 - Usa triggers para a detecção
 - Realiza a notificação automática de alterações
 - Comando CREATE TRIGGER
 - Oracle
 - PostgreSQL
 - DB2
- Somente poucas fontes oferecem recursos de gatilhos
 - Podem ser intrusivos
 - Podem onerar o servidor de dados

Logs

- Armazenam **todas as transações** que ocorrem na aplicação
 - Inserções, remoções e atualizações
 - Consultas
- CDC
 - **Percorre** o arquivo de *log*
 - Identifica as **diferenças** que devem ser extraídas
- Exemplo
 - Logstash (Elastic)

- Requerem **privilégio** de administrador de banco de dados
- Possuem **formatos proprietários**
- **Protegidos** pelo sistema

Snapshots

- Foto dos valores de dados armazenados em um certo momento no banco de dados
- CDC
 - Compara o *snapshot* da extração anterior e com o *snapshot* da extração atual
 - Gera um *arquivo delta* com as atualizações
- Comparação de *snapshots*
 - Solução comumente usada

Comparações cada vez maiores precisam ser realizadas à medida que o volume de dados da fonte cresce

Projeto do Data Mart: **funcionario**



funcionario

funcionario (funcPK, funcMatricula, funcNome, funcSexo, funcDataNascimento,
funcDiaNascimento, funcMesNascimento, funcAnoNascimento,
funcCidade, funcEstadoNome, funcEstadoSigla, funcRegiaoNome,
funcRegiaoSigla, funcPaisNome, funcPaisSigla)

Tabela relacional

funcPK	funcMatricula	funcNome	funcSexo	funcDataNascimento	funcDiaNascimento	funcMesNascimento	...
--------	---------------	----------	----------	--------------------	-------------------	-------------------	-----

Extração para o Exemplo da BI Solutions

- Quais dados são extraídos de quais fontes de dados?
 - Carga inicial: todos os dados de interesse



Tabela **funcionarioRelacional**
Relacional

funcMatricula funcNome funcSexo funcDataNasc funcCidade funcEstado funcPais

colaboradorJSON

Arquivo
JSON

```
[{"colab_matricula": " ", "colab_nome": " ", "colab_sexo": , "colab_data_nasc": " ",  
"colab_cidade": " ", "colab_estado": " ", "colab_pais": " "}, ...]
```

Planilha
Excel

empregadoPlanilha

	A	B	C	D	E	F
1	Matrícula do Empregado	Nome do Empregado	Sexo do Empregado	Data de Nascimento	Cidade de Residência	Estado de Residência
2						

Extração para o Exemplo da BI Solutions

- Como os dados de interesse são extraídos?
 - Por meio de APIs
- Com qual frequência os dados de interesse são extraídos?
 - Processamento: em lote
 - Frequência da extração incremental: mensal



Extração para o Exemplo da BI Solutions

- Qual **técnica** empregar para identificar dados das fontes que foram alterados?
 - `funcionarioRelacional`: *Triggers*
 - `colaboradorJSON`: Comparação de *Snapshots*
 - `empregadoPlanilha`: Comparação de *Snapshots*



Operações

- Extração
- Tradução
- Limpeza
- Integração
- Carga

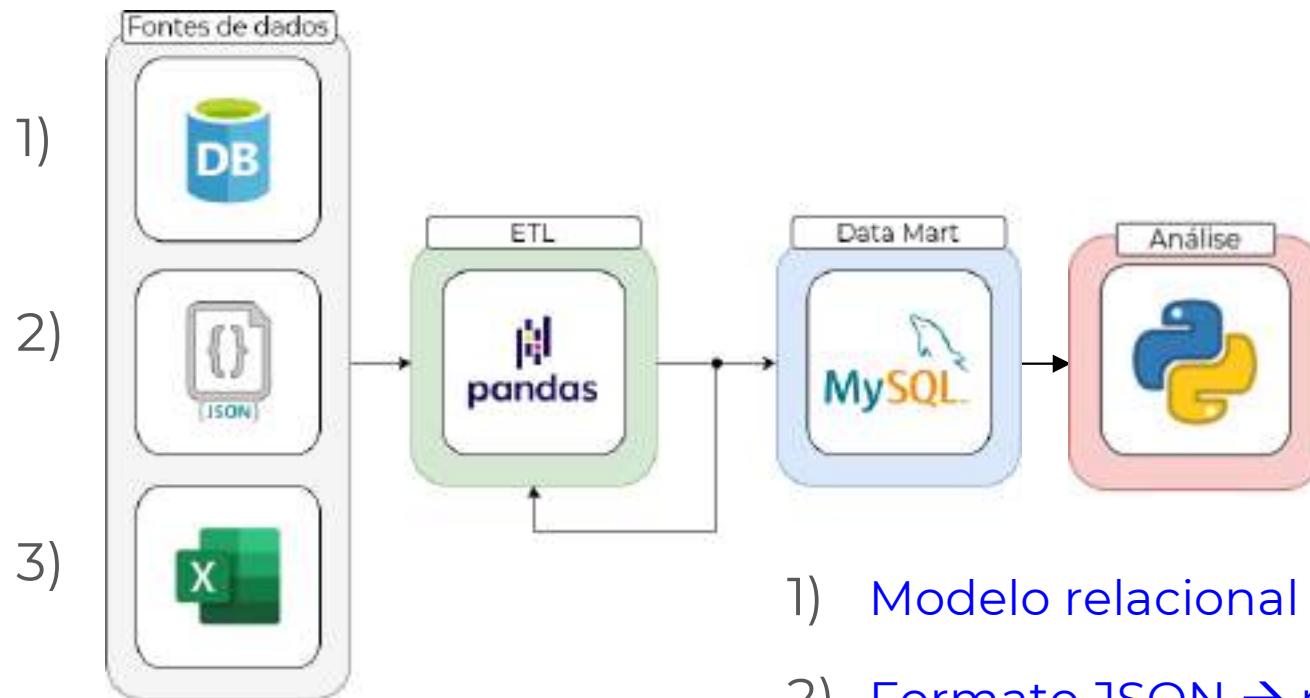
Problema

- Dados armazenados nas fontes de dados
 - Heterogêneos
 - Seguem diferentes modelos de dados
 - São representados por conceitos diferentes
 - Possuem diferentes formatos
 - Redundantes, inconsistentes e até mesmo complementares
- Dados armazenados no *data warehouse*
 - Devem seguir um projeto e uma forma de organização específica

Objetivos

- Realizar a **conversão** entre o formato nativo das fontes de dados e o formato do *data warehouse*
 - Esquema
 - Instância: valores dos dados e tipos de dados
- Garantir a manutenção da temporalidade
 - Maioria das fontes de dados **não é histórica**, mas o *data warehouse* sempre deve armazenar dados históricos
 - Dados temporais podem ser adicionados indicando o **momento de atualização dos dados nas fontes** de dados ou o **momento de armazenamento no data warehouse**

Rotinas para o Exemplo da BI Solutions



- 1) Modelo relacional → modelo relacional
- 2) Formato JSON → modelo relacional
- 3) Formato de planilha Excel → modelo relacional

Operações

- Extração
- Tradução
- Limpeza
- Integração
- Carga

Limpeza

- Garante a **acurácia** e a **qualidade** dos dados
 - Dados devem atender às restrições de integridade impostas pelas regras de negócio
- Exemplos
 - Comprimentos de campos inválidos e uso de caracteres inválidos
 - Dados incompletos, em branco, ou usando abreviações não padronizadas
 - Duplicações dos mesmos dados (ou seja, redundância)
 - Descrições inconsistentes, violação de restrições de integridade, associação de valores inconsistentes

Deve ser realizada durante todas as atividades do processo de ETL/ELT

Limpeza para o Exemplo da BI Solutions

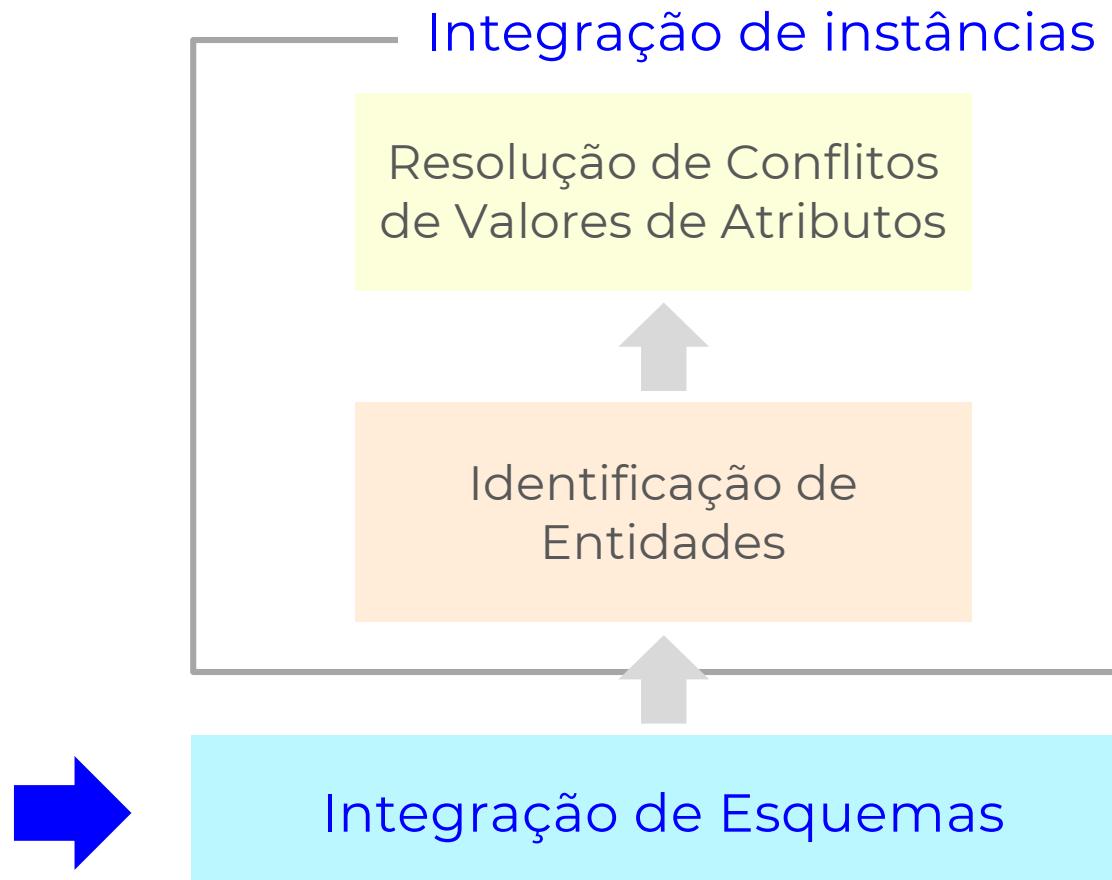
- Uso de estratégias aprendidas em disciplinas anteriores
 - Técnicas Avançadas de Captura e Tratamento de Dados
- Exemplos
 - Redundância
 - Detecção de *outliers*
 - Tratamento de informações errôneas
 - Manutenção da acurácia do **número de matrícula** dos funcionários



Operações

- Extração
- Tradução
- Limpeza
- Integração
- Carga

Visão Geral



Integração de Esquemas

- Definição
 - Especificação de **mapeamentos** que descrevem os relacionamentos semânticos entre os esquemas das fontes de dados e o esquema do data warehouse
- Relativismo semântico
 - **Conflitos** existentes entre duas ou mais representações
 - **Diferentes usuários** modelam o mesmo pedaço do mundo real de **diferentes formas**, de acordo com as suas percepções

Conflito

- Surge quando duas ou mais representações do mesmo conceito **não são idênticas**
 - Usam construtores diferentes
 - Aplicam diferentes restrições de integridade
- Tipos
 - **de nome**
 - **semântico**
 - **estrutural**

discrepâncias existentes entre os esquemas
apresentam mais do que um tipo de conflito

Conflito de Nome

- Relacionado aos **nomes** que representam os diferentes elementos nos esquemas a serem integrados
- **Sinônimos**
 - Diferentes nomes são aplicados ao mesmo elemento
 - `funcionarioRelacional`, `colaboradorJSON` e `empregadoPlanilha` indicam funcionários
- **Homônimos**
 - Mesmo nome é aplicado a diferentes elementos
 - Data representa `data de contratação` em um esquema e `data de aniversário` em outro



Conflito Semântico

- Mesmo elemento é modelado em diferentes esquemas, porém representando conjuntos que se sobrepõem
- Exemplo
 - **funcionarioRelacional**: funcionários da área de *Engenharia*
 - **colaboradorJSON**: funcionários da área de *Marketing*
 - **empregadoPlanilha**: funcionários da área de *Recursos Humanos*
 - funcionários podem ser diferentes entre si
 - o mesmo funcionário pode estar em mais do que uma fonte de dados, desde que ele mudou de área de atuação durante a sua trajetória



Conflito Estrutural

- Diferentes **construtores estruturais** são utilizados para modelar o mesmo conceito representado em diferentes fontes de dados
- Exemplo
 - **Endereço**
 - **Atributo** do esquema **funcionário** em uma fonte de dados
 - Esquema **endereço** composto de outros atributos: **nome da rua**, **número** e **complemento**



Mapeamentos (Data Mart e Fonte Relacional)

■ ■ funcionario ≡ funcionarioRelacional

funcMatricula = funcMatricula

funcNome = funcNome

funcSexo = funcSexo

■ funcDataNascimento = funcDataNasc

funcCidade = funcCidade

■ funcEstadoSigla = funcEstado

funcPais = funcPais

■ conflito de nome
(sinônimos)

■ conflito semântico

Mapeamentos (Data Mart e Fonte JSON)



funcionario ≡ colaboradorJSON

- funcMatricula = colab_matricula
 - funcNome = colab_nome
 - funcSexo = colab_sexo
 - funcDataNascimento = colab_data_nasc
 - funcCidade = colab_cidade
 - funcEstadoSigla = colab_estado
 - funcPais = colab_pais
- conflito de nome (sinônimos)
 - conflito semântico

Mapeamentos (Data Mart e Fonte Planilha)



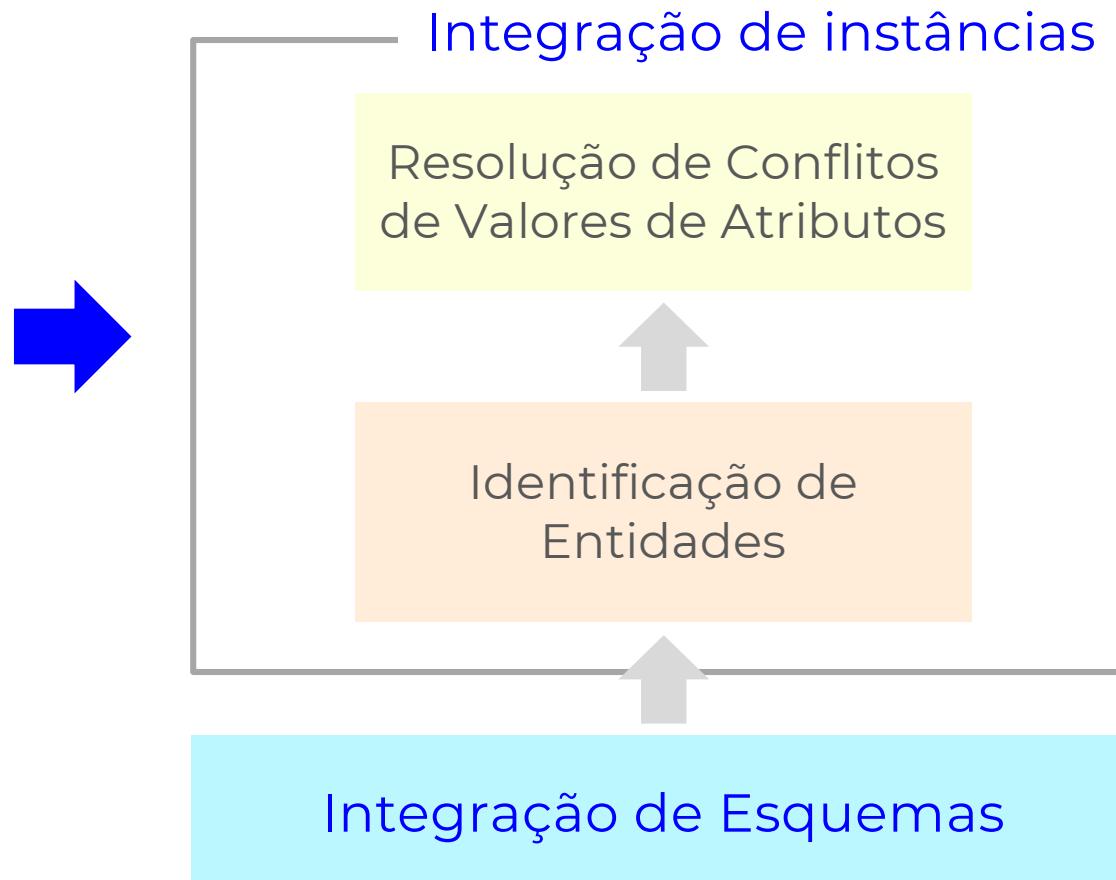
■ ■ funcionario ≡ empregadoPlanilha

- funcMatricula = Matrícula do Empregado
- funcNome = Nome do Empregado
- funcSexo = Sexo do Empregado
- funcDataNascimento = Data de Nascimento
- funcCidade = Cidade de Residência
- funcEstadoNome = Estado de Residência

■ conflito de nome
(sinônimos)

■ conflito semântico

Visão Geral



Identificação de Entidades

- Objetivos
 - Identificar quais entidades das fontes de dados heterogêneas referem-se à mesma entidade do mundo real
 - Agrupar essas entidades em agrupamentos de entidades similares
- Cenários
 - Identificação **única** das entidades por meio de atributos chave
 - **Não existe um atributo** que identifica univocamente cada entidade

Identificação Unívoca das Entidades



- Fontes de dados
 - `funcionarioRelacional`: atributo `funcMatricula`
 - `colaboradorJSON`: atributo `colab_matricula`
 - `empregadoPlanilha`: atributo `Matrícula do Empregado`
- Valores dos atributos referentes à matrícula dos funcionários
 - Analisados na operação de limpeza dos dados
 - Representam valores acurados

Resolução de Conflitos de Valores de Atributos

- Resolve **inconsistências** nos **valores** dos dados das entidades que referem-se à mesma entidade do mundo real, mas que **diferem** nos valores dos seus atributos
- Exemplos



Sexo do Funcionário

F/M

Feminino/Masculino

0/1

Nome do Funcionário

Adenildo Campos

Campos, Adenildo

A. Campos

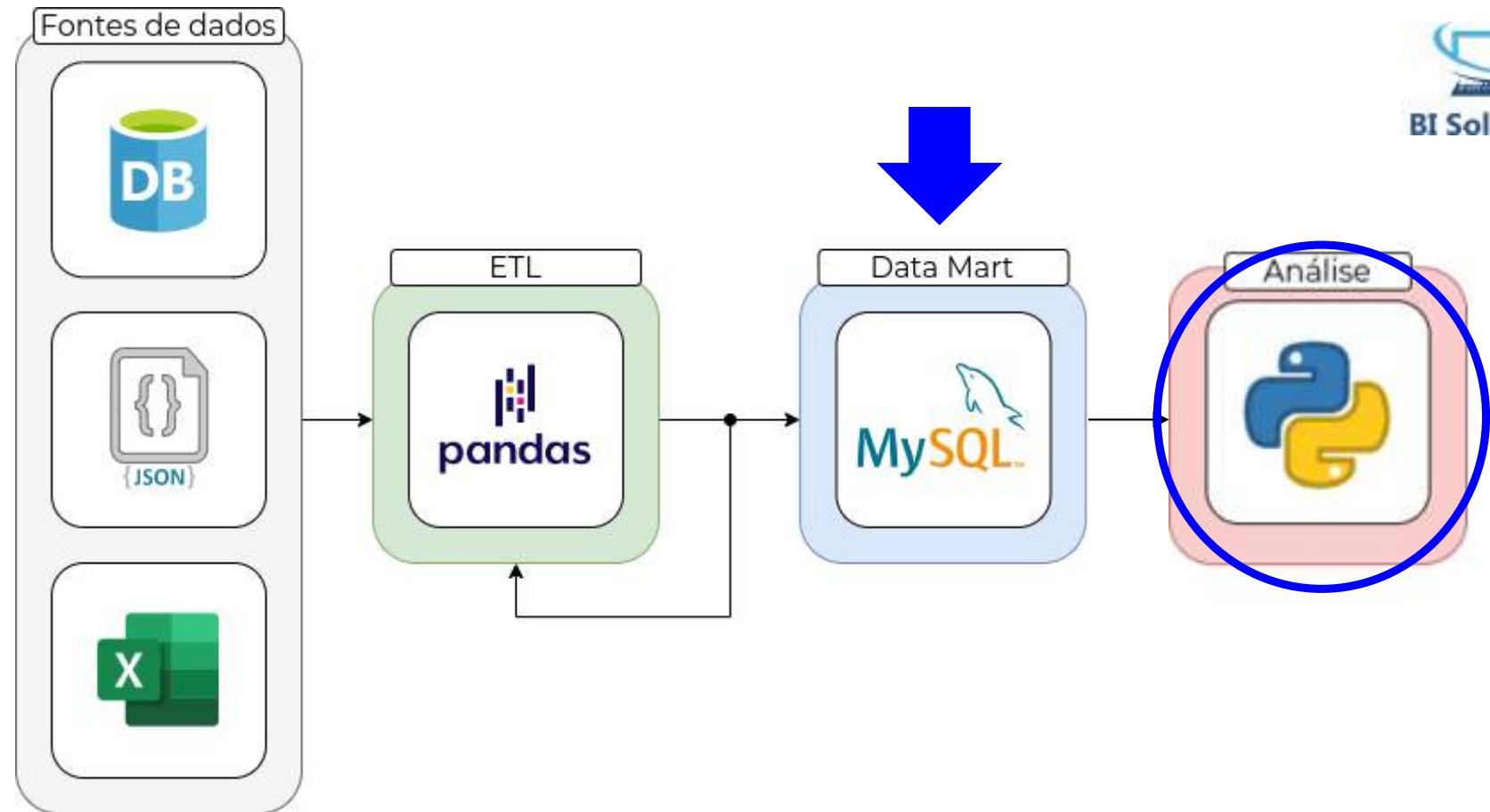
Operações

- Extração
- Tradução
- Limpeza
- Integração
- Carga

Funcionalidades

- Realizar processamentos adicionais
 - Geração de agregações (visões materializadas)
 - Necessidade de construção de índices
 - Verificação de restrições de integridade
 - Necessidade de ordenação dos dados
- Armazenar os dados no *data warehouse*

Armazenamento no Data Mart



Análise de Dados com Base em Processamento Massivo em Paralelo

Aula 4: Modelagem Conceitual de ETL/ELT

Cristina Dutra de Aguiar
ICMC/USP
cdac@icmc.usp.br



Agenda

- Características
- Modelo Intuitive
- Exemplo para a BI Solutions

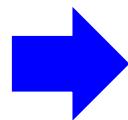
Projeto de ETL/ELT

- Características desejadas para o sucesso
 - Robustez
 - Boa documentação
 - Facilidade de Manutenção
- Representado como um *workflow*
 - Cadeia de operações ou tarefas aplicadas aos dados
 - Representado por meio de um *modelo*

Níveis de Abstração de um Modelo

nível de abstração

alto



Nível Conceitual

Nível Lógico

Nível Físico

baixo

complementa a análise de requisitos, facilitando o entendimento do processo

descreve os detalhes técnicos das tarefas envolvidas

incorpora aspectos de implementação e otimização

Desafios e Motivação

- Desafios
 - Criticidade e **complexidade** do processo de ETL/ELT
 - **Grande esforço** despendido para a construção do processo
 - Propensão a **falhas**
- Motivação
 - **Facilitar e padronizar** a construção do processo de ETL/ELT
 - **Melhorar a qualidade** do processo de ETL/ELT e dos dados armazenados no DW

Modelagem Conceitual

- Realizada na **fase inicial** do processo de ETL/ELT
 - Requisitos dos usuários de SSD
 - Entendimento do conteúdo e da estrutura das fontes de dados
 - Enfoque na estrutura proposta para o DW
- Produz um **esquema conceitual**
 - Representação gráfica e abstrata do processo de ETL/ELT

Requisitos da Modelagem Conceitual

- Características desejadas
 - Simplicidade e completude
 - Clareza, consistência, não ambiguidade
- Diagrama produzido
 - Deve ser facilmente entendido pelos usuários finais que são conhecedores do negócio
 - muitas vezes não possuem conhecimento profundo de tecnologias
 - Deve contribuir para diminuir o esforço dos projetistas e desenvolvedores

Funcionalidades Adicionais

- Documenta as decisões tomadas
- Possibilita a análise de impacto das alterações que ocorrem no ciclo de vida da aplicação de *data warehousing*
 - Alterações nas fontes de dados
 - Evolução dos requisitos ou das regras de negócio
 - Correção de erros cometidos durante a fase de projeto
- Facilita a exploração de cenários alternativos

Modelagem Conceitual *versus* Ferramentas

- Modelagem Conceitual
 - Fornece **alto nível de abstração**, sendo **independente** de ferramentas específicas
- Ferramentas de ETL/ELT disponíveis
 - Relacionadas aos **níveis lógico e físico**
 - Exemplos
 - Pentaho Data Integration (Kettle)
 - Talend Open Studio
 - CloverETL
 - Oracle Warehouse Builder
 - IBM Infosphere
 - MSSQLServer Integration Services

Agenda

- Características
- Modelo Intuitive
- Exemplo para a BI Solutions

Características do Modelo

- Operadores
 - Entidades de alto nível
 - Representam as operações típicas de ETL/ELT
 - Possuem notação gráfica
- Combinação de operadores
 - Realizada por setas unidirecionais que indicam a propagação dos dados
 - Representa sequências que compõem o workflow de ETL/ELT

Características do Workflow

- Início e final
 - **Início:** um ou mais repositórios de dados
 - **Final:** um ou mais repositórios, sendo o principal o *data warehouse* (ou *data mart*)
- Funcionalidades dos operadores
 - **Manipulação** de dados
 - **Organização** do fluxo de dados no *workflow*

Entrada
Parâmetro
Saída

Especificação dos Operadores: Entrada

- Um ou mais conjuntos de dados
- Classificação
 - **Unária**: apenas um conjunto de dados →
 - **Binária**: dois conjuntos de dados →
 - **N-ária**: vários conjuntos de dados →
...

Especificação dos Operadores: Tipos de Parâmetro

- Lista de atributos
 - Nome de um atributo do conjunto de dados
 - Exemplo: funcNome, funcMatricula
- Condição
 - Expressão relacional
 - Exemplo: funcCidade = São Carlos
 - Expressão lógica
 - funcEstadoSigla = SP AND funcMatricula > 32879

Operações relacionais

= > < <> <= >=

Operadores lógicos

NOT, AND, OR

Especificação dos Operadores: Tipos de Parâmetro

- Ordenação
 - **Ordem** crescente ou decrescente dos dados
 - Exemplo: asc e desc
- Precedência
 - Qual **conjunto de dados** deve ser **analisado primeiro**
 - Exemplo: dados do conjunto A — dados do conjunto B
- Lista de atribuições
 - **Atributo** <--- **valor**
 - Exemplo: funcMatricula <--- 234334, funcEstadoSigla <--- PE

Especificação dos Operadores: Saída

- Um ou mais conjuntos de dados
- Classificação
 - **Unária**: apenas um conjunto de dados →
 - **Binária**: dois conjuntos de dados →
 - **N-ária**: vários conjuntos de dados →
... →

Categorias de Operadores

- Classificação baseada em
 - Características dos operadores
 - Efeitos que causam sobre os dados ou sobre a organização do processo

Armazenamento

Manipulação de
Dados

Inicialização

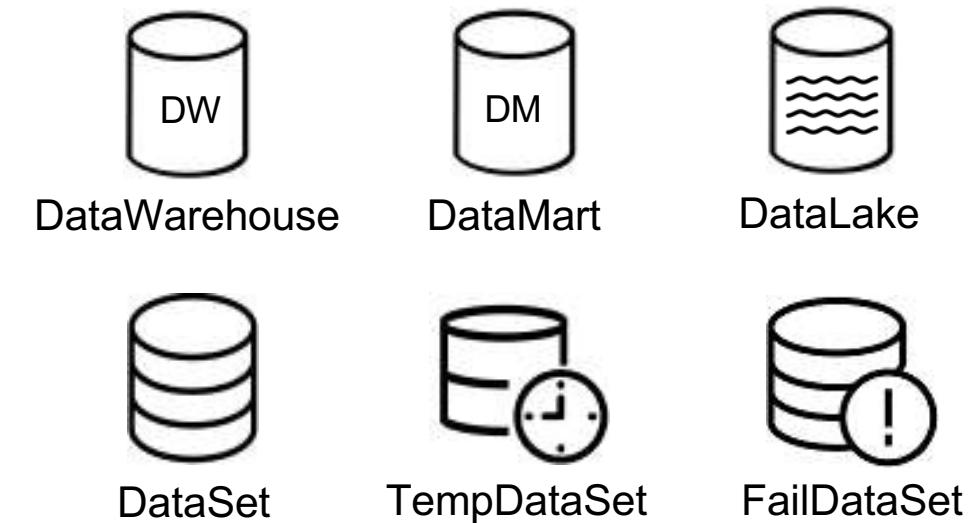
Agregação

Fluxo

Especiais

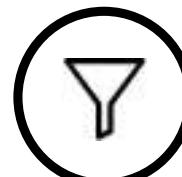
Operadores de Armazenamento

Representam locais de armazenamento de dados, tais como repositórios, arquivos ou bancos de dados

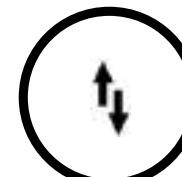


Operadores de Manipulação de Dados

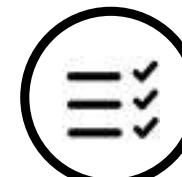
Representam **operações** de transformação e de **limpeza** dos dados



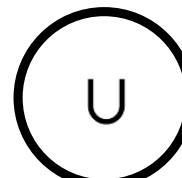
Filter



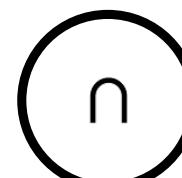
Sort



Update



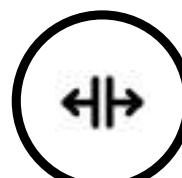
Union



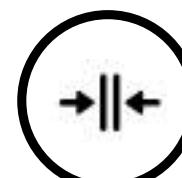
Intersect



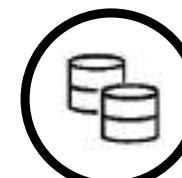
Diff



Split

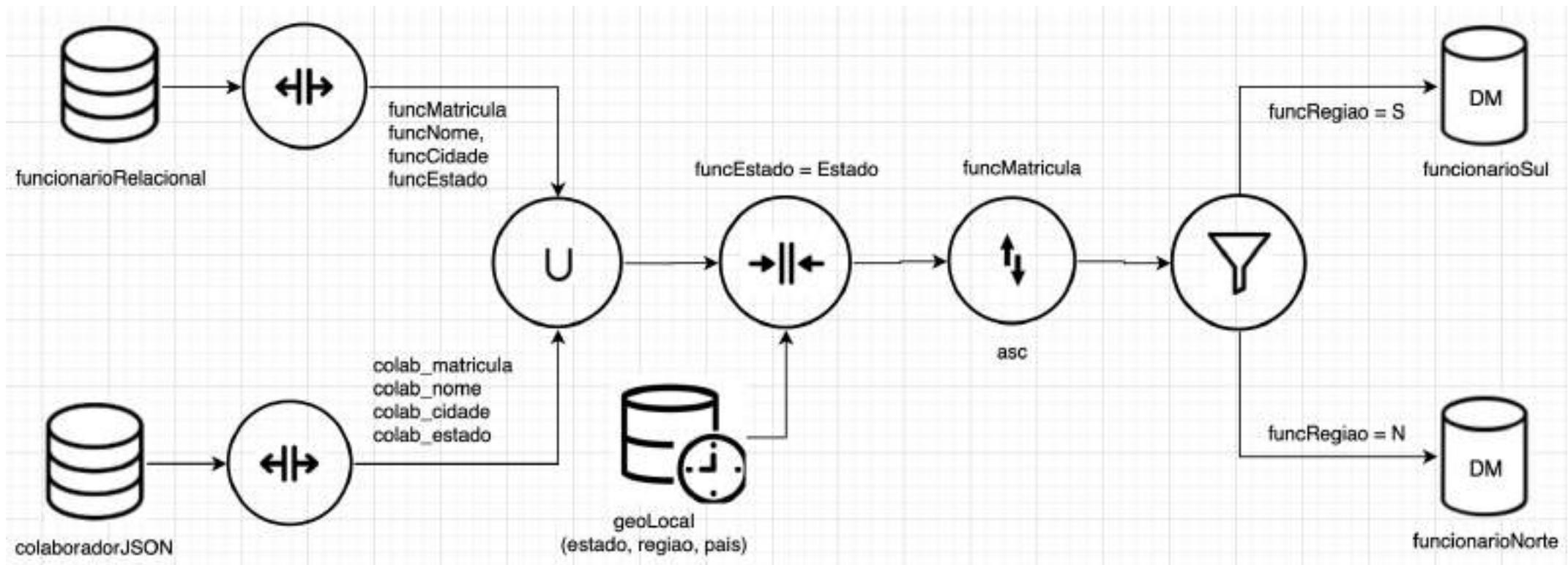


Join



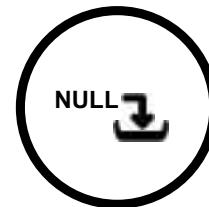
Copy

Geração de Data Marts Regionais de Funcionários

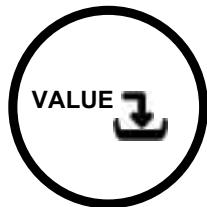


Operadores de Inicialização

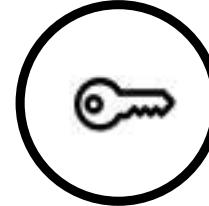
Representam a inicialização
de um **atributo** com um
valor específico



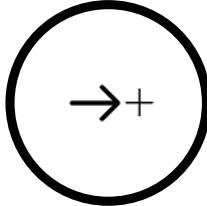
SetNullAsDefault



SetDefaultValue



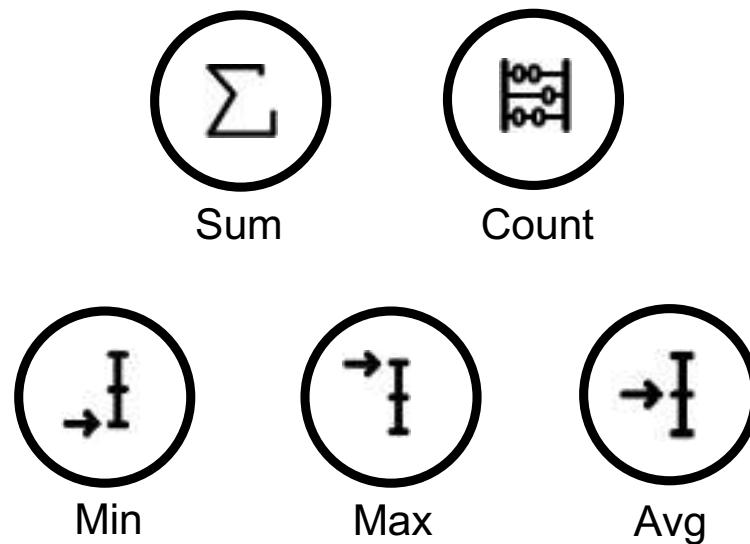
SurrogateKey



Sequence

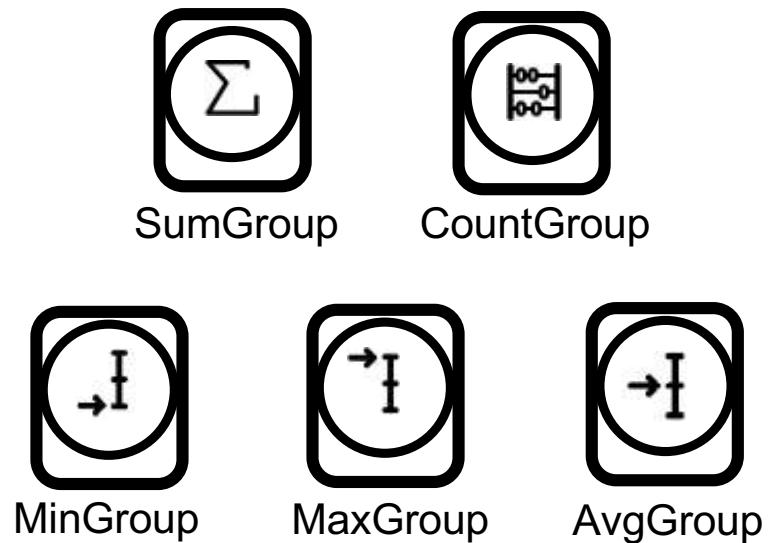
Operadores de Agregação (1/2)

Representam **funções** que processam os valores de um atributo e **retornam um único valor**

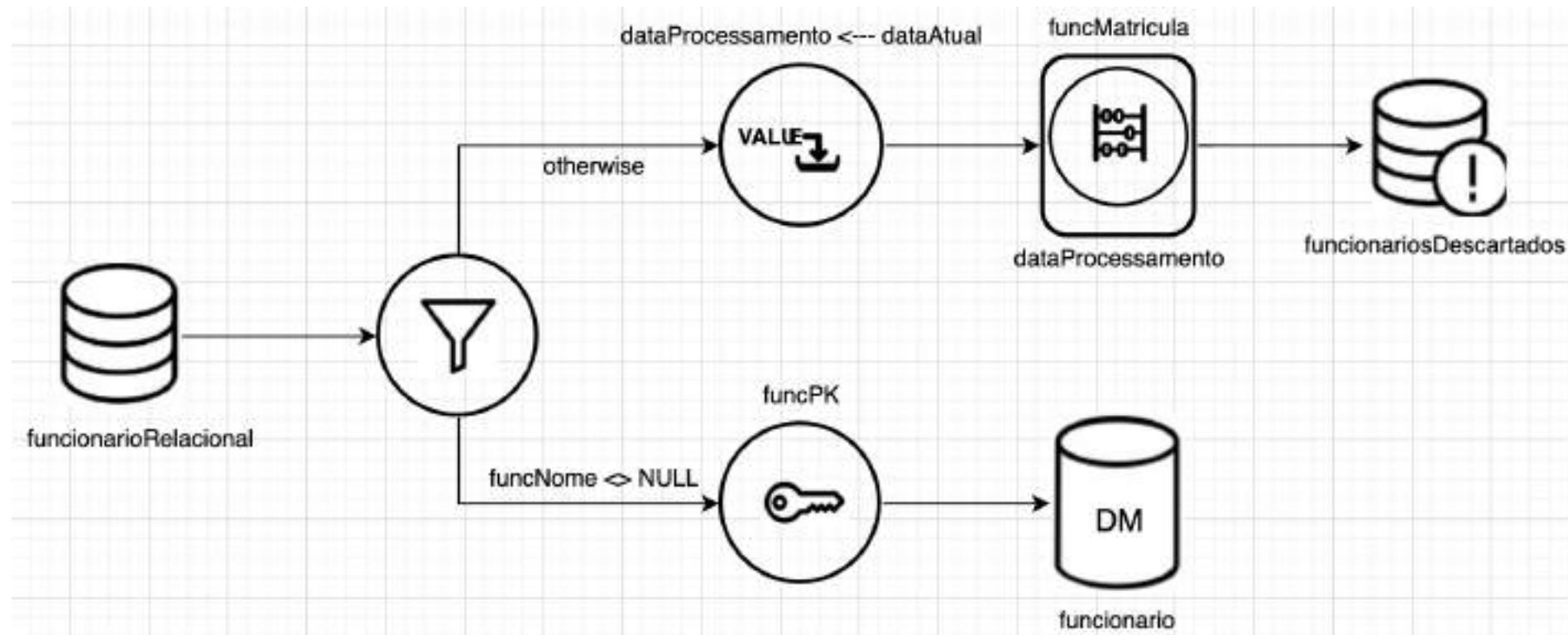


Operadores de Agregação (2/2)

Representam funções que processam os valores de um atributo e retornam um único valor para cada atributo do agrupamento

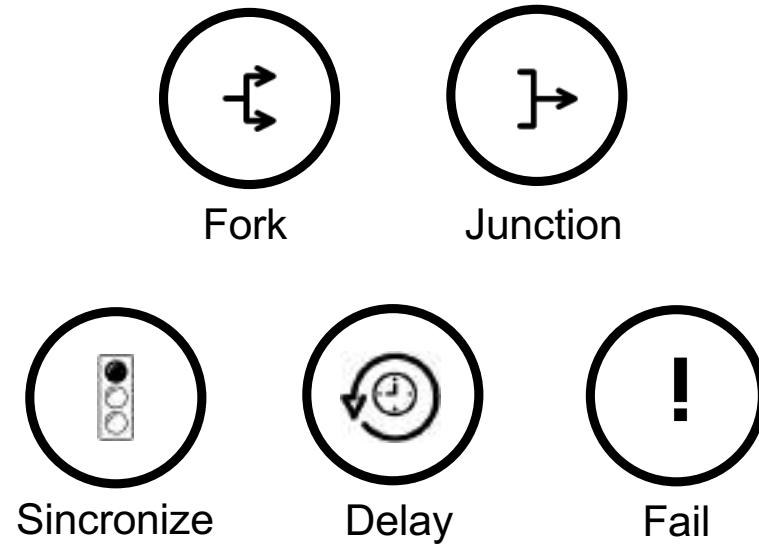


Análise do Processamento de Funcionários

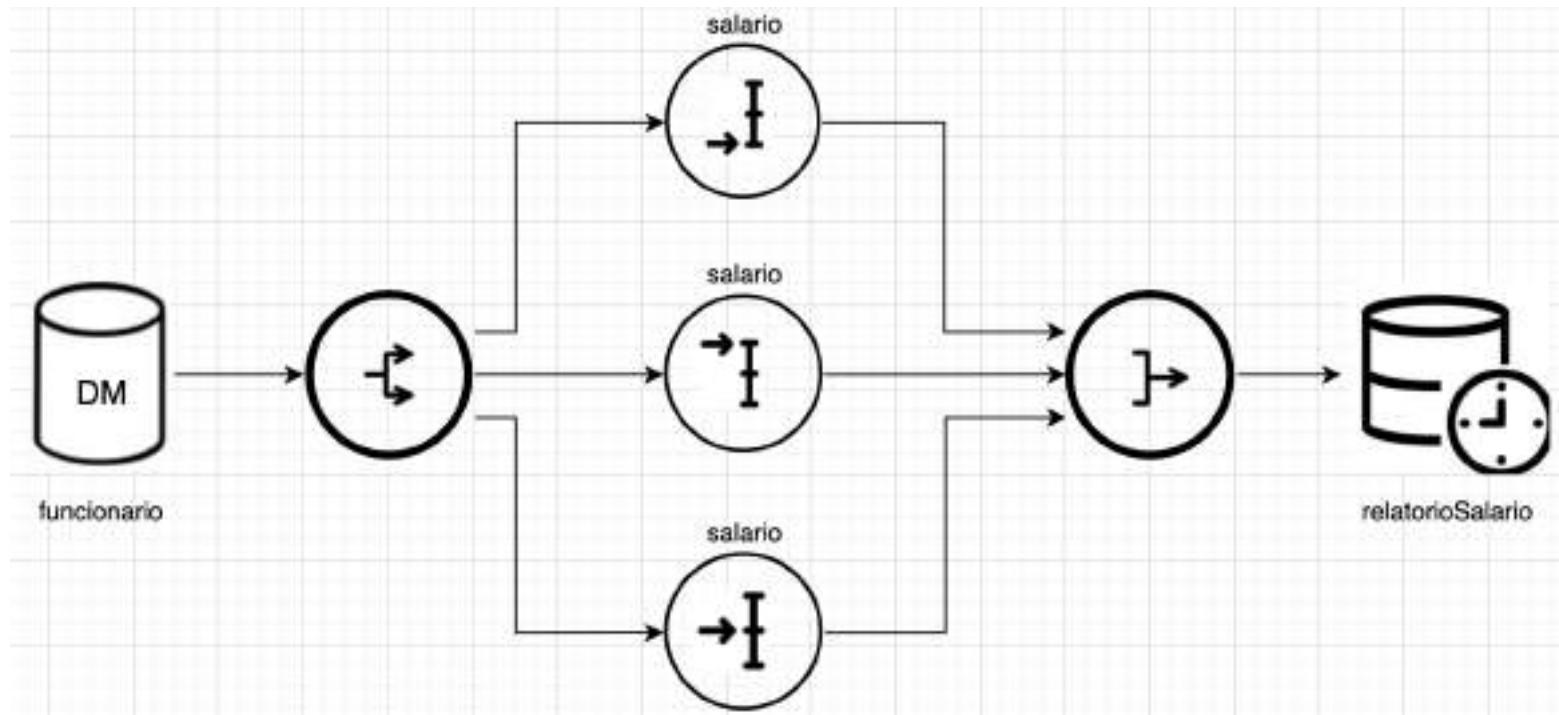


Operadores de Fluxo

Representam uma alteração no fluxo dos dados, sem impactar esses dados

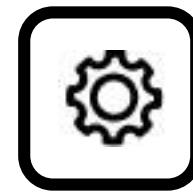


Geração de Relatório de Salários



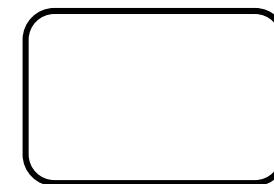
Operadores Especiais

Representam operações que envolvem **especificidades**, complementando as funcionalidades dos demais operadores



Function

Function title
Short description
Details



SubFlow

Material Suplementar

- Tabela descritiva dos operadores
 - Operador e sua funcionalidade
 - Representação gráfica, incluindo entrada, parâmetro e saída

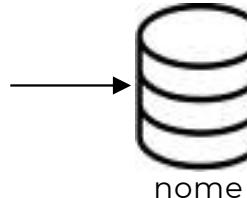
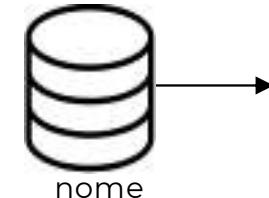
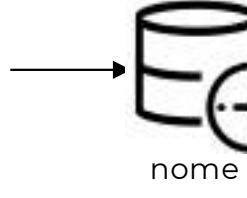
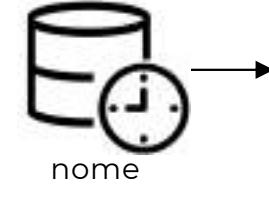
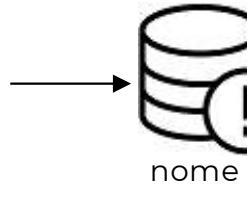
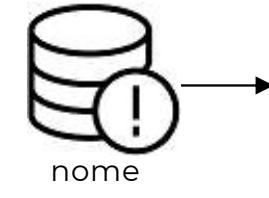
Operador	Funcionalidade	Representação Gráfica
Fork	direcionar um conjunto de dados para duas ou mais tarefas paralelas	... → ⚡ → → ⚡ → ...

- Acompanha os slides no formato .pdf

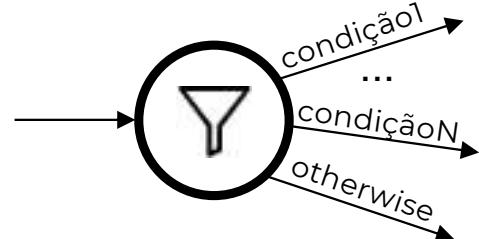
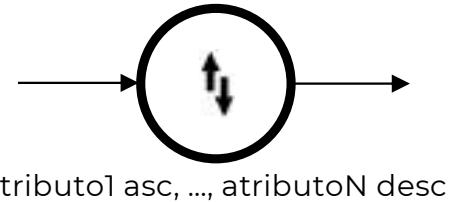
Operadores de Armazenamento

Operador	Funcionalidade	Representação Gráfica
DataWarehouse	armazena dados multidimensionais	
DataMart	data warehouse com escopo limitado	
DataLake	armazena dados brutos que ainda não foram transformados	

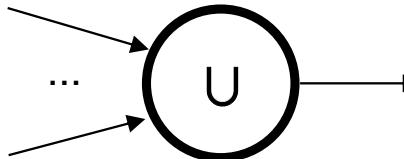
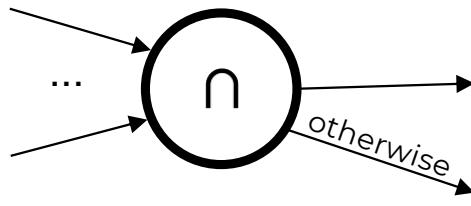
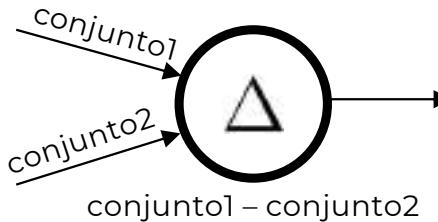
Operadores de Armazenamento

Operador	Funcionalidade	Representação Gráfica
DataSet	armazena dados	 ou 
TempDataSet	área temporária de armazenamento de dados	 ou 
FailDataSet	armazena dados rejeitados por uma operação ou para efeitos de log	 ou 

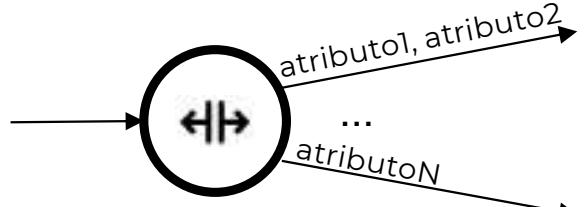
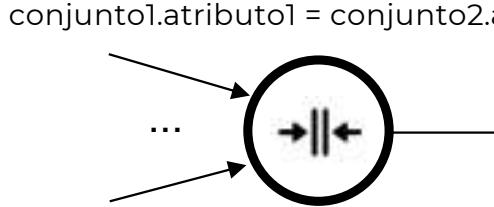
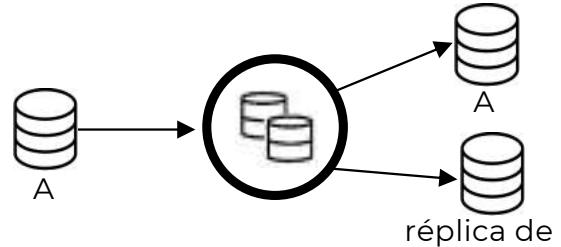
Operadores de Manipulação de Dados

Operador	Funcionalidade	Representação Gráfica
Filter	seleciona subconjuntos de dados de acordo com condições definidas	
Sort	ordena dados em ordem crescente ou decrescente, de acordo com atributos definidos	
Update	altera os valores dos dados, de acordo com condições definidas sobre atributos	

Operadores de Manipulação de Dados

Operador	Funcionalidade	Representação Gráfica
Union	une conjuntos de dados, gerando um conjunto que contém todos os dados de entrada, sem repetição	
Intersect	une conjuntos de dados, gerando um conjunto que contém apenas os dados em comum, sem repetição	
Diff	gera os dados que estão presentes no primeiro conjunto de dados, mas não estão no segundo conjunto	

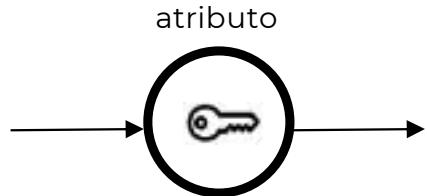
Operadores de Manipulação de Dados

Operador	Funcionalidade	Representação Gráfica
Split	separa atributos de um conjunto de dados, direcionando-os para fluxos diferentes	
Join	combina dois conjuntos de dados usando como base atributos em comum	
Copy	a partir de um conjunto de dados de entrada, gera o próprio conjunto e uma réplica deste	

Operadores de Inicialização

Operador	Funcionalidade	Representação Gráfica
SetNullAsDefault	inicializa um atributo específico com o valor nulo, para todos os itens do conjunto de dados	<p>lista de atributos</p> <p>condição</p>
SetDefaultValue	atribui um determinado valor para um atributo específico, para todos os itens do conjunto de dados	<p>lista de atribuições</p>

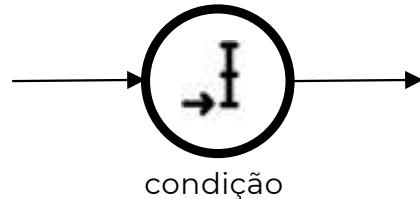
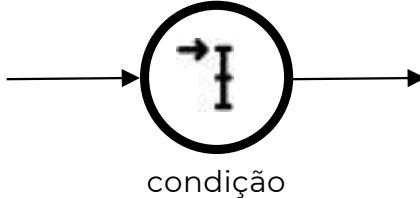
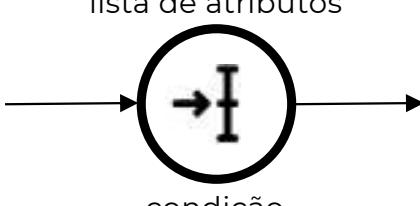
Operadores de Inicialização

Operador	Funcionalidade	Representação Gráfica
SurrogateKey	cria um atributo chave e atribui a ele um valor único para cada item do conjunto de dados	
Sequence	cria um atributo não-chave e atribui a ele um valor único para cada item do conjunto de dados, o qual é gerado a partir de um valor inicial	

Operadores de Agregação

Operador	Funcionalidade	Representação Gráfica
Sum	para cada atributo, soma seus valores e produz um único valor	<p>lista de atributos</p> <p>condição</p>
Count	para cada atributo, conta seus valores e produz um único valor	<p>lista de atributos</p> <p>condição</p>

Operadores de Agregação

Operador	Funcionalidade	Representação Gráfica
Min	para cada atributo, produz o menor valor	<p>lista de atributos</p>  <p>condição</p>
Max	para cada atributo, produz o maior valor	<p>lista de atributos</p>  <p>condição</p>
Avg	para cada atributo, produz o valor médio	<p>lista de atributos</p>  <p>condição</p>

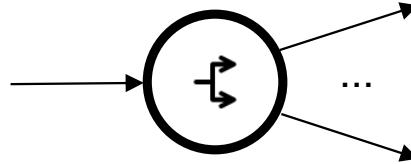
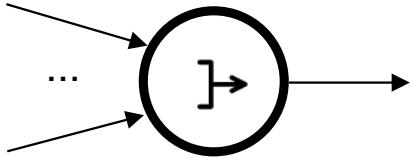
Operadores de Agregação

Operador	Funcionalidade	Representação Gráfica
SumGroup	para cada grupo, soma os valores dos dados de cada atributo	 <p>lista de atributos lista de atributos de agrupamento condição</p>
Count	para cada agrupamento, conta o número de dados de cada atributo	 <p>lista de atributos lista de atributos de agrupamento condição</p>

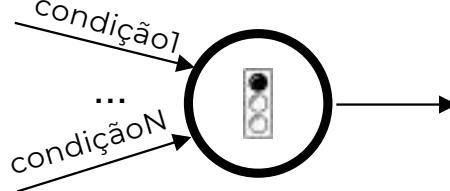
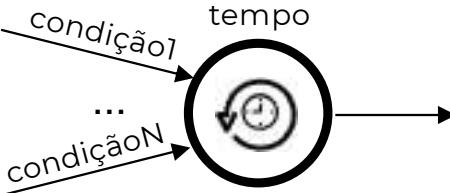
Operadores de Agregação

Operador	Funcionalidade	Representação Gráfica
MinGroup	para cada grupo, produz o menor valor de cada atributo	<p>lista de atributos</p> <p>listas de atributos de agrupamento condição</p>
MaxGroup	para cada grupo, produz o maior valor de cada atributo	<p>lista de atributos</p> <p>listas de atributos de agrupamento condição</p>
AvgGroup	para cada grupo, produz o valor médio de cada atributo	<p>lista de atributos</p> <p>listas de atributos de agrupamento condição</p>

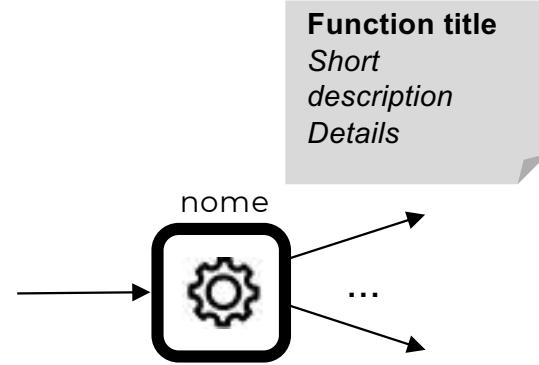
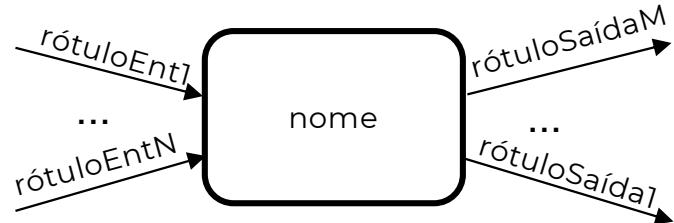
Operadores de Fluxo

Operador	Funcionalidade	Representação Gráfica
Fork	direciona um conjunto de dados para dois ou mais fluxos executados em paralelo ou para um repositório e fluxos	
Junction	junta dois ou mais fluxos executados em paralelo	

Operadores de Fluxo

Operador	Funcionalidade	Representação Gráfica
Sincronize	sincroniza dois ou mais fluxos paralelos com base em uma condição de finalização	
Delay	temporiza o tempo no qual será feita a análise de conjuntos de dados de fluxos paralelos	
Fail	representa um fluxo alternativo para indicar falha	

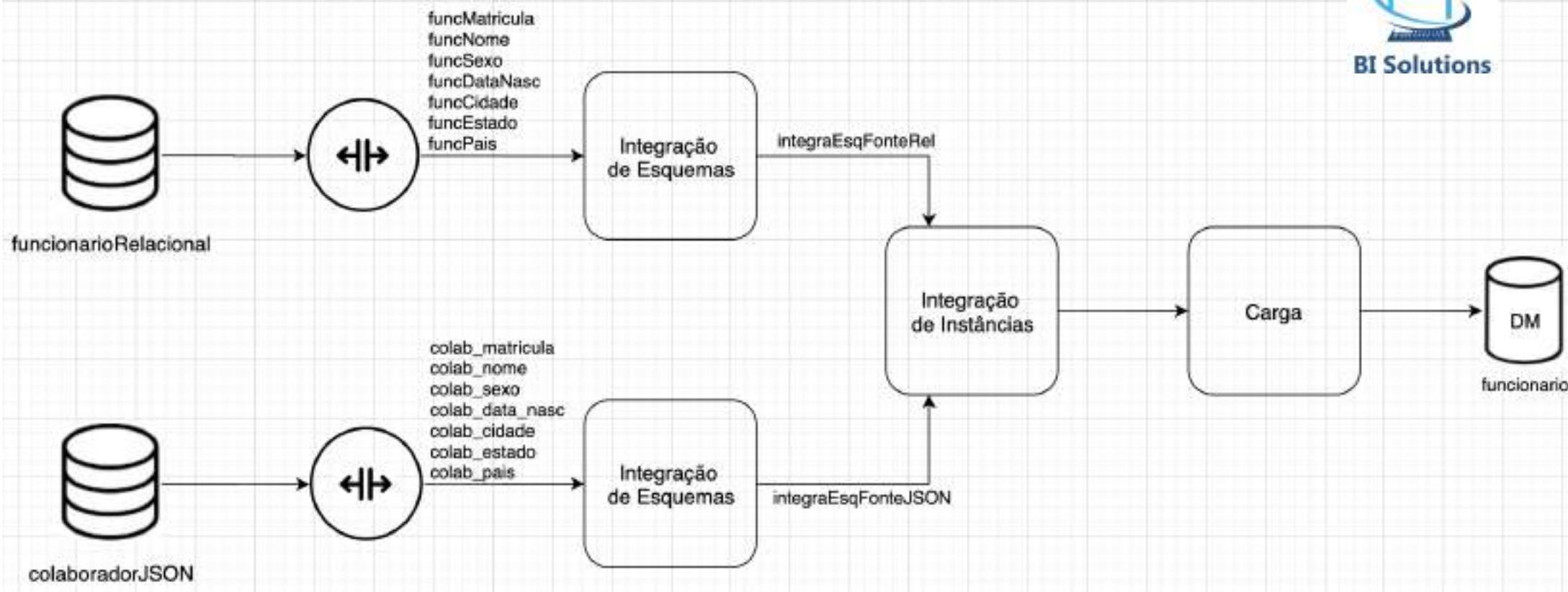
Operadores Especiais

Operador	Funcionalidade	Representação Gráfica
Function	representa operações ou atividades muito específicas que não podem ser representadas pelos outros operadores	
SubFlow	encapsula subfluxos que envolvem conjuntos de tarefas específicas	

Agenda

- Características
- Modelo Intuitive
- Exemplo para a BI Solutions

Processo de ETL da BI Solutions



Diagrams

Copyright © 2020. Todos os direitos reservados
ao CeMEAI-USP. Proibida a cópia e reprodução
sem autorização

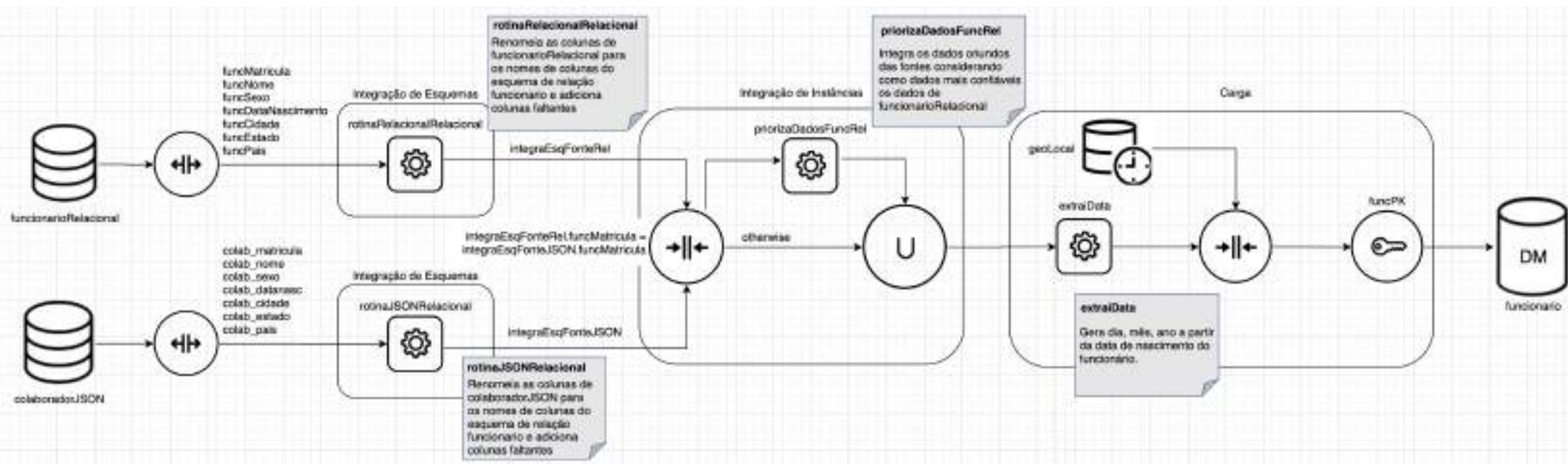


- Exemplo do Processo de ETL
- Implementação em Pandas

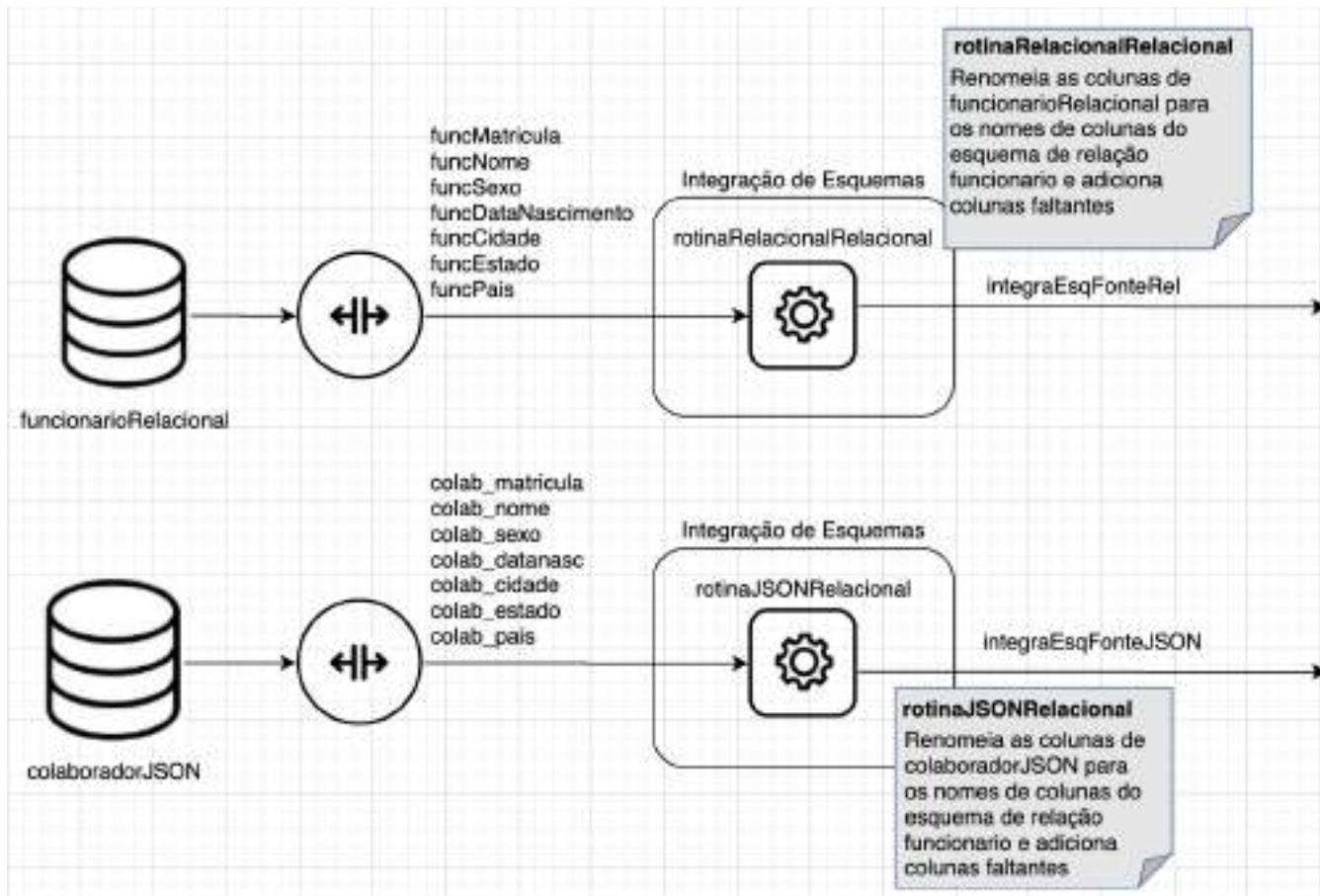
Diagrama Conceitual Completo



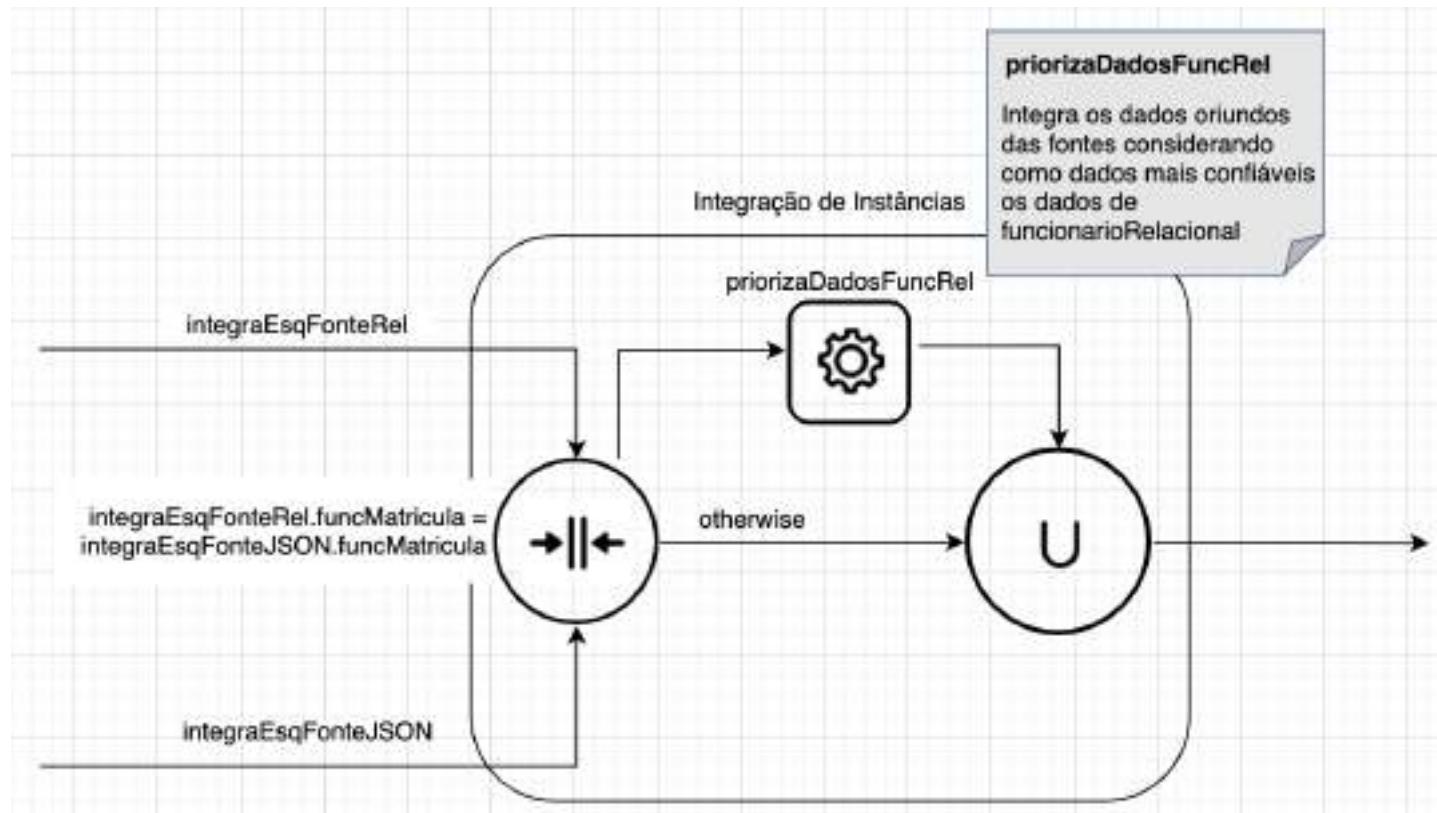
BI Solutions



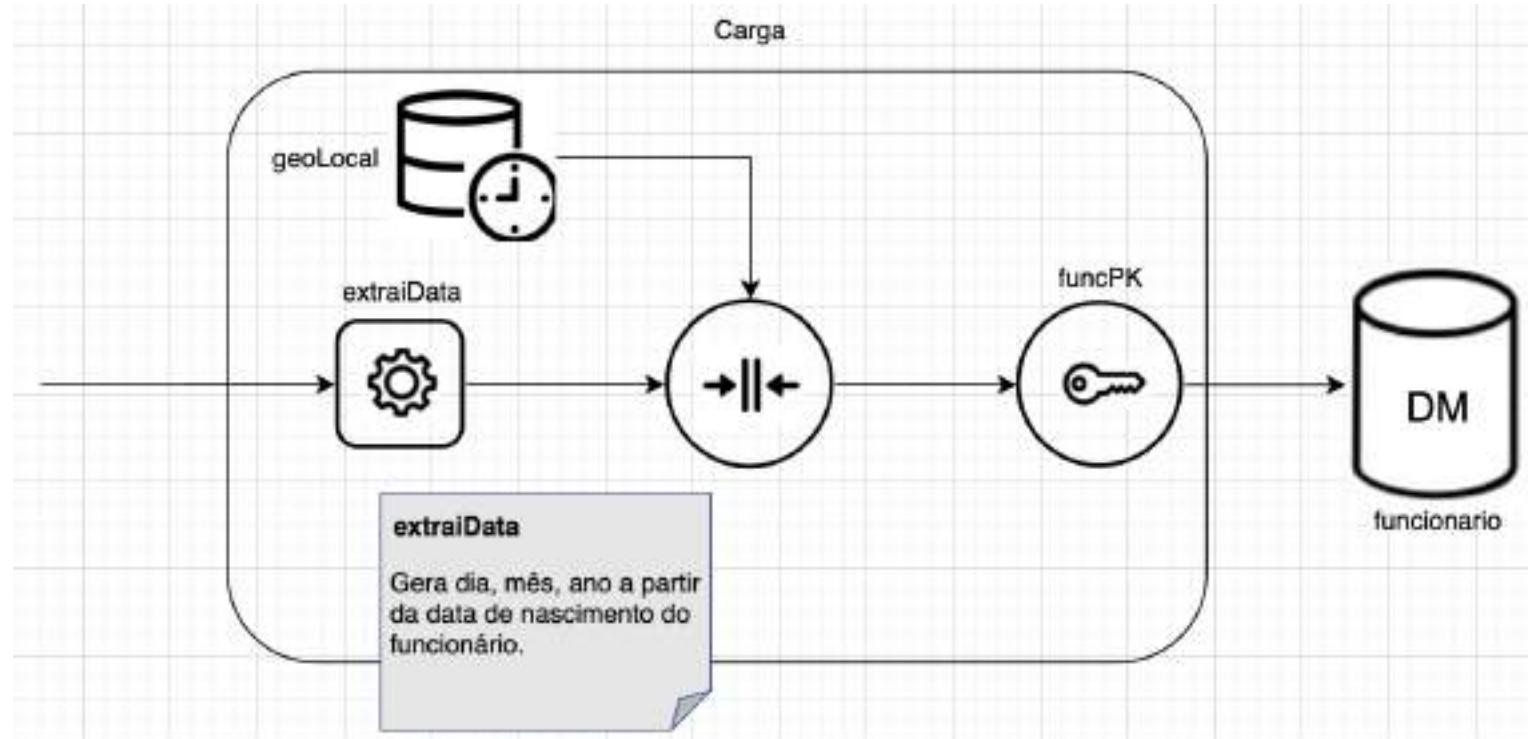
Extração e Integração de Esquemas



Integração de Instâncias



Carga



Diagrams

Copyright © 2020. Todos os direitos reservados
ao CeMEAI-USP. Proibida a cópia e reprodução
sem autorização

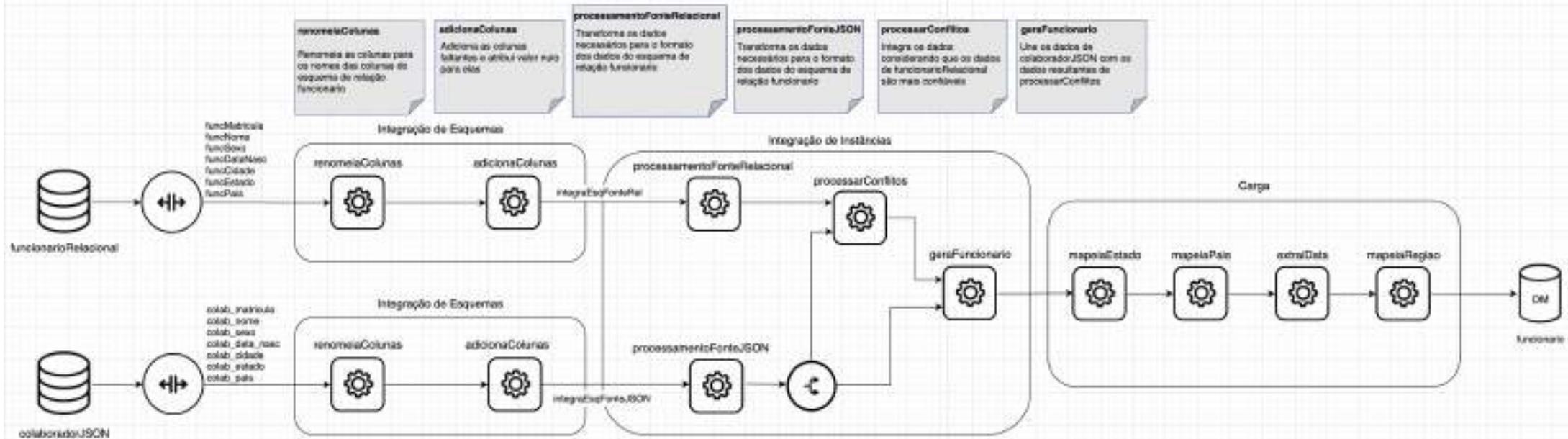


- Exemplo do Processo de ETL
- Implementação em Pandas

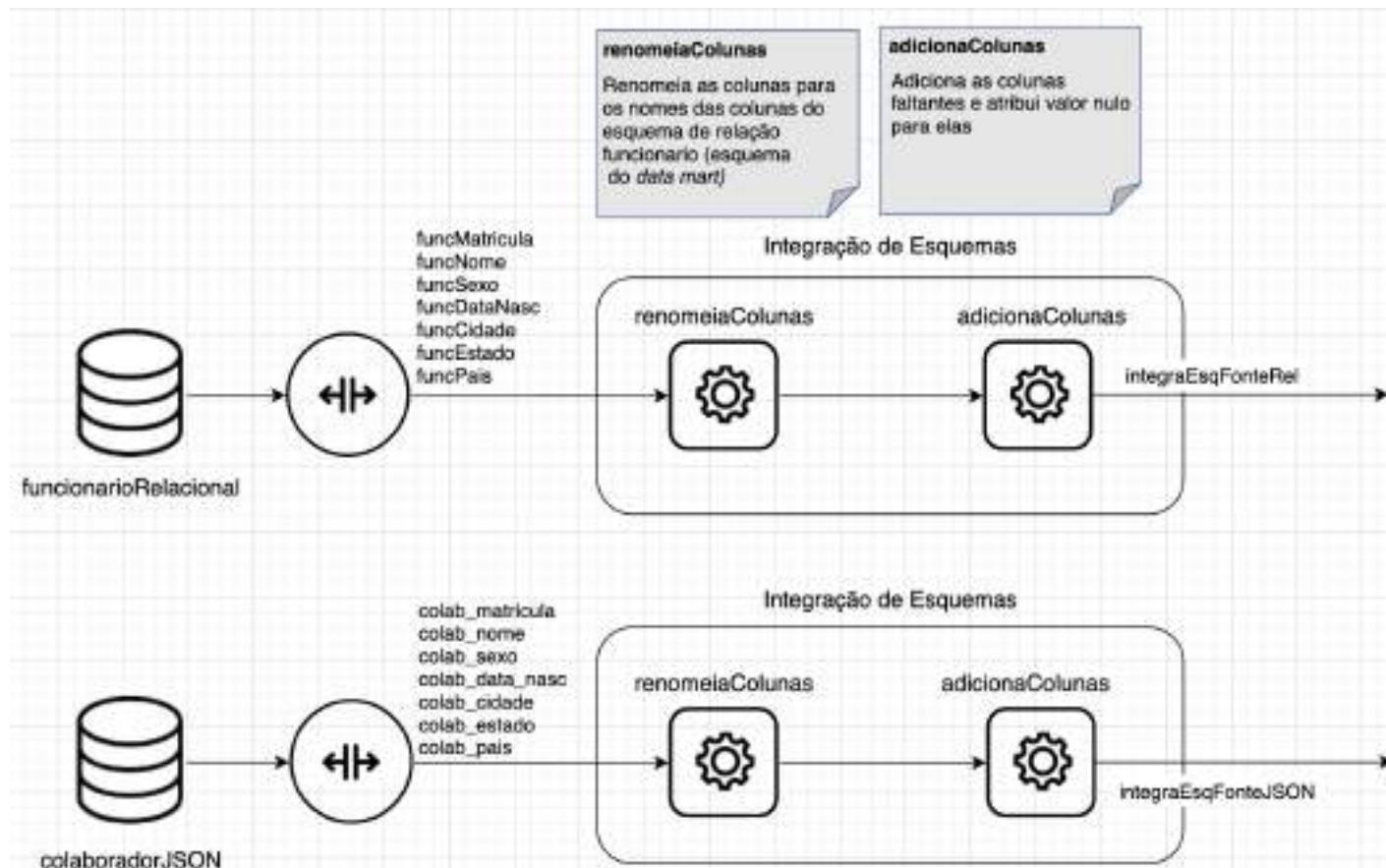
Visão Geral da Implementação em Pandas



BI Solutions



Extração e Integração de Esquemas



Integração de Instâncias



processamentoFonteRelacional

Transforma os dados necessários para o formato dos dados do esquema de relação funcionário

processamentoFonteJSON

Transforma os dados necessários para o formato dos dados do esquema de relação funcionário

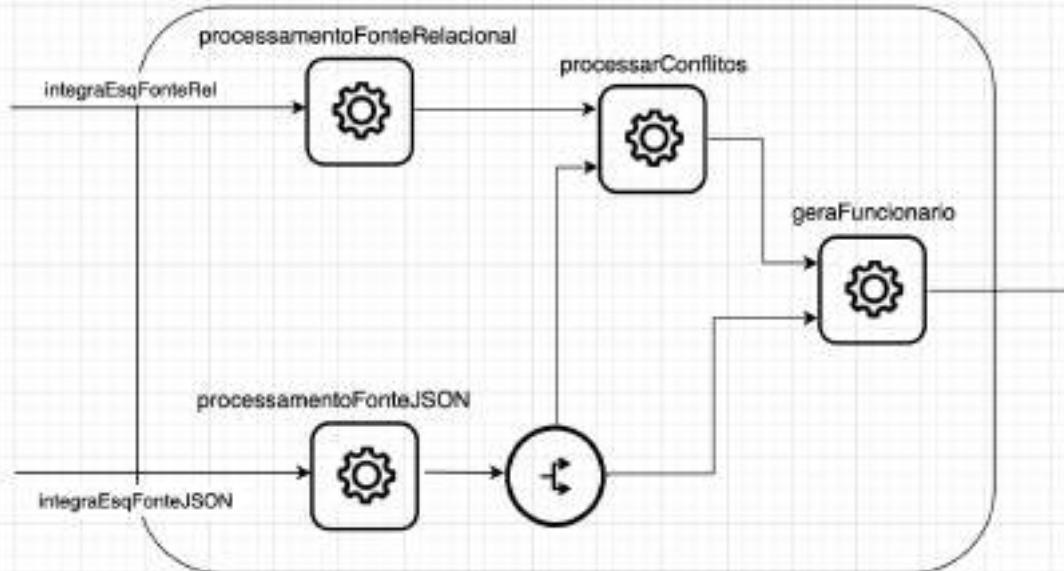
processarConflitos

Integra os dados considerando que os dados de funcionarioRelacional são mais confiáveis

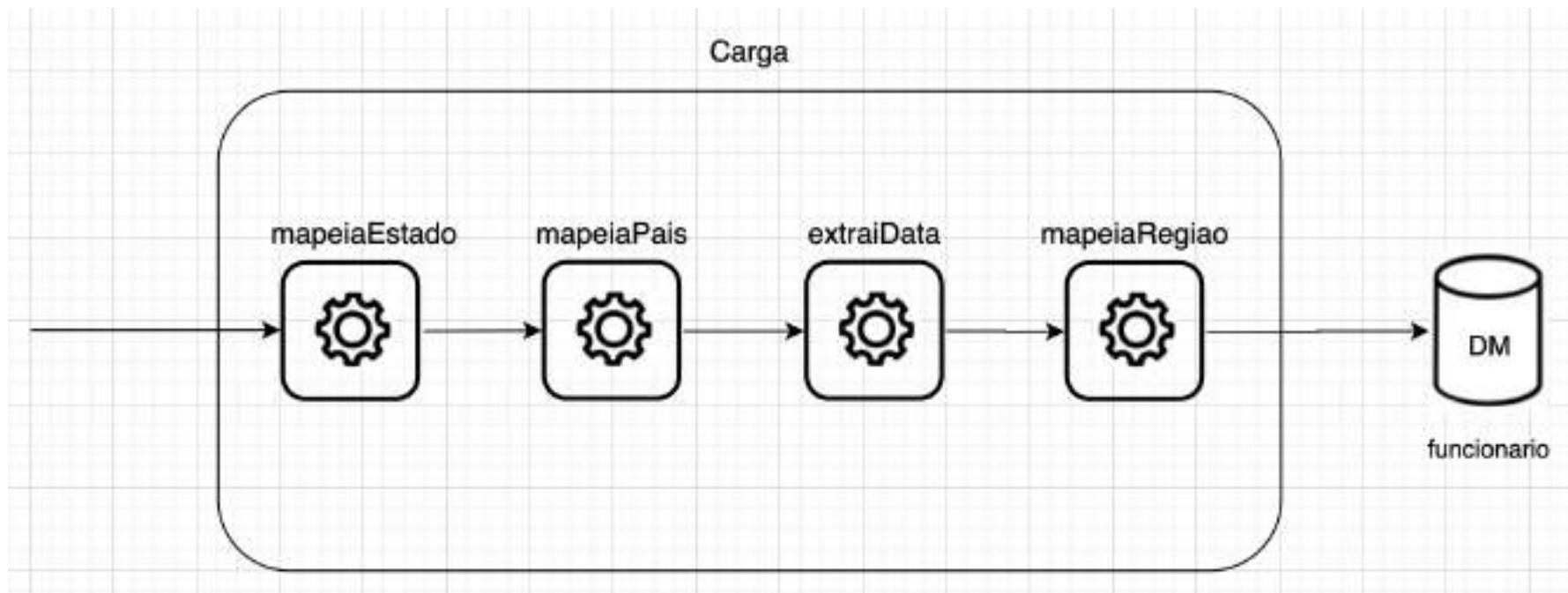
geraFuncionario

Une os dados de colaboradorJSON com os dados resultantes de processarConflitos

Integração de Instâncias



Carga



Análise de Dados com Base em Processamento Massivo em Paralelo

Lista de Exercícios: Modelagem Conceitual de ETL/ELT

Profa. Dra. Cristina Dutra de Aguiar

Observação:

Esta lista contém exercícios referentes à Aula 04. A resposta de cada exercício encontra-se destacada na cor azul. Recomenda-se fortemente que a lista de exercícios seja respondida antes de se consultar as respostas dos exercícios.

1. Descreva qual a importância de se modelar conceitualmente um *workflow* de ETL/ELT antes de implementá-lo.

A anuência de que o processo de ETL/ELT é a etapa mais custosa de todo o projeto de *data warehousing* já é algo consolidado tanto na literatura quanto no mercado. Sendo assim, é importante modelá-lo conceitualmente a fim de contribuir para diminuir o esforço dos projetistas e desenvolvedores durante a implementação do *workflow*. Além disso, o esquema conceitual é um recurso precioso para a documentação das decisões tomadas na construção do processo de ETL/ELT, para a análise de impacto das alterações necessárias para atendimento de demandas que ocorrem no ciclo de vida do *data warehouse* (tais como alterações nas fontes de dados, evolução dos requisitos ou das regras de negócio, necessidade de melhoria no desempenho das consultas, correção de erros cometidos durante a fase de projeto, entre outros) e para facilitar a exploração de cenários alternativos para a solução desejada.



2. Por que é interessante o uso de um modelo específico para projetar o processo de ETL/ELT (como o Modelo Intuitive) e não o uso de um modelo de processos genérico (como o Modelo BPMN - *Business Process Model and Notation*)?

O uso de um modelo específico para projetar o processo de ETL/ELT justifica-se pelo fato desse modelo prover clareza na representação do processo. O modelo é baseado em operadores que representam graficamente as operações usualmente presentes em processos de ETL/ELT. Como resultado, o projeto final provê melhor compreensão por parte do usuário final e não evidencia detalhes de implementação. Modelos genéricos também podem ser utilizados. Entretanto, esses modelos definem operadores mais genéricos, os quais não descrevem de forma gráfica e visual as operações usualmente presentes em processos de ETL/ELT.



3. Considere as seguintes categorias de operadores:

- (a) Operadores de armazenamento;
- (b) Operadores de manipulação de dados;
- (c) Operadores de inicialização;
- (d) Operadores de agregação;
- (e) Operadores de fluxo;
- (f) Operadores especiais.

Descreva, de forma sucinta, o objetivo de cada uma das categorias supracitadas.

Operadores de armazenamento: Os operadores de armazenamento podem ser usados para representar áreas de armazenamento de dados, tais como repositórios, arquivos ou bases de dados.

Operadores de manipulação de dados: Os operadores de manipulação de dados são usados para representar as tarefas de transformação e de limpeza que são aplicadas aos dados extraídos das diversas fontes para torná-los compatíveis com a estrutura proposta para o *data warehouse*.

Operadores de inicialização: Os operadores de inicialização de dados servem para representar a atribuição de valores específicos para atributos de um conjunto de dados.

Operadores de agregação: Os operadores de agregação podem ser usados para representar funções que, quando aplicadas a um conjunto de dados, processam os valores de um atributo específico e retornam um único valor como resultado. Se forem definidos atributos de agrupamento, as funções processam os valores de um atributo e retornam um único valor para cada atributo de agrupamento.

Operadores de fluxo: Os operadores de fluxo de dados permitem representar uma alteração no fluxo dos dados no *workflow* de ETL, sem impactar esses dados.

Operadores especiais: Os operadores especiais representam operações que envolvem especificidades, complementando as funcionalidades dos demais operadores.



4. Considere um *data mart* implementado segundo o modelo relacional que armazena três conjuntos de dados, conforme descrito nas relações seguir:

- (i) funcionario (funcPK, funcNome, funcEndereco, funcDataNasc)
- (ii) cargo (cargoPK, nomeCargo, descricaoCargo)
- (iii) funcCargo (funcPK, cargoPK, salario)

Faça um diagrama conceitual que considere desde a extração dos dados dessas relações até a geração de um relatório departamental que exibe, para cada cargo, o nome do cargo, o maior salário, o menor salário e o salário médio. Note que deve ser feito um diagrama conceitual, ou seja, não é para se fazer um *workflow* que represente a implementação.

Resposta:

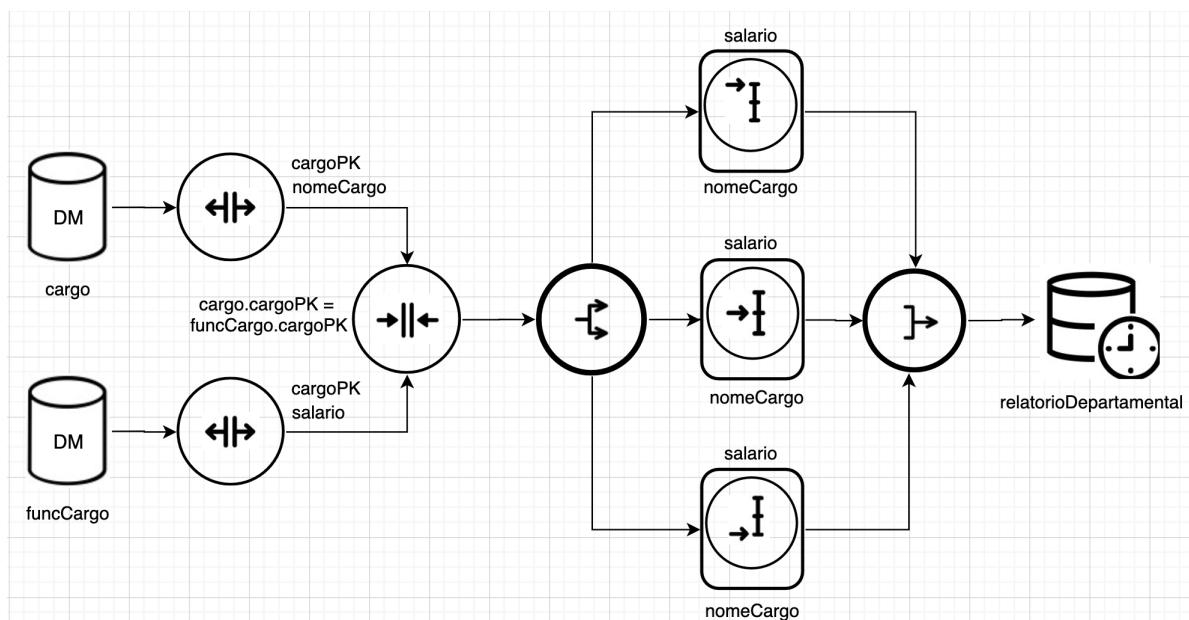


Figura 1: Resposta da questão 4.

5. Considere o *workflow* de ETL modelado na Figura 2, o qual ilustra a extração de funcionários de duas bases de origem: (i) funcionarioRelacional, a qual representa um sistema gerenciador de banco de dados relacional; e (ii) colaboradorJSON, a qual representa uma coleção de documentos JSON.

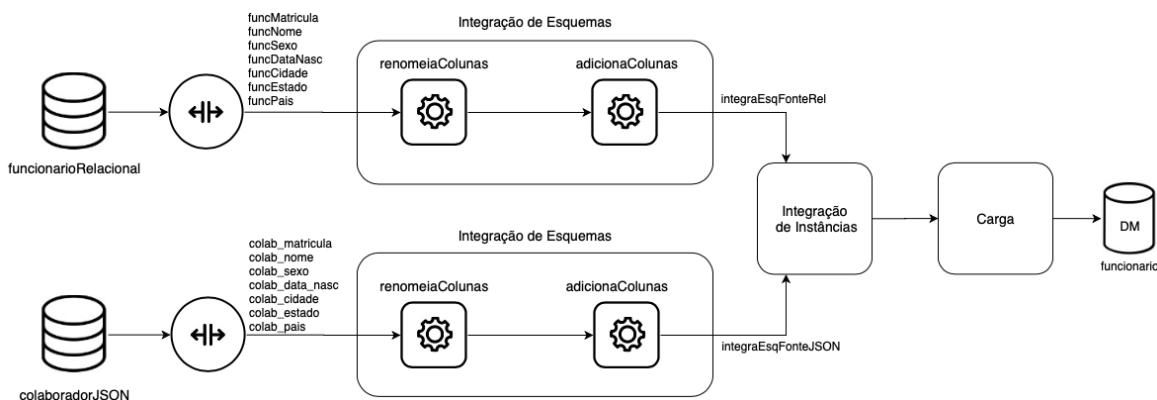


Figura 2: Visão geral do processo de ETL da **BI Solutions**.

Considere que a empresa **BI Solutions**, responsável pela manutenção do *workflow* ilustrado na Figura 2, necessita incluir mais uma fonte de dados no processo de ETL. Essa nova fonte de dados, denominada *empregadoPlanilha*, representa uma planilha Excel que contém os seguintes dados de funcionários: “Matrícula do Empregado”, “Nome do Empregado”, “Sexo do Empregado”, “Data de Nascimento”, “Cidade de Residência”, “Estado de Residência”.

Estenda o *workflow* de ETL para incluir essa nova fonte de dados. Modele apenas as etapas anteriores ao subfluxo de integração de instâncias.

Resposta:

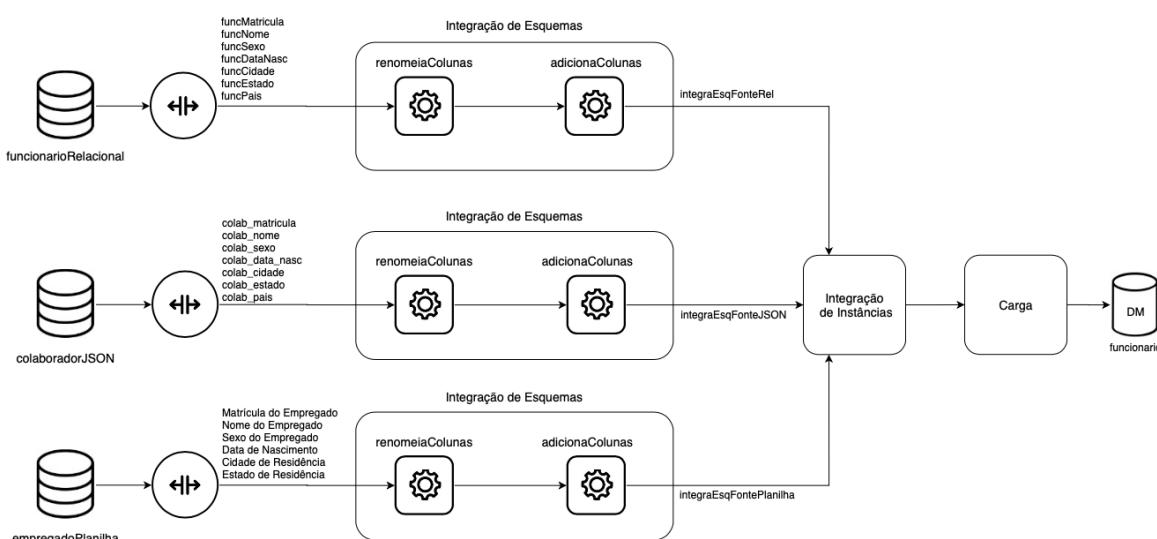


Figura 3: Resposta da questão 4.



6. Considere o subfluxo relacionado à “Integração de Instâncias” da **BI Solutions**, representado tanto no diagrama conceitual da Figura 4 quanto no workflow da Figura 5. Esse subfluxo considera como entradas dados oriundos das fontes de dados funcionarioRelacional e colaboradorJSON. Estenda o diagrama conceitual e o workflow de forma que o subfluxo relacionado à Integração de Instâncias também considere como entrada os dados oriundos da fonte de dados empregadoPlanilha.

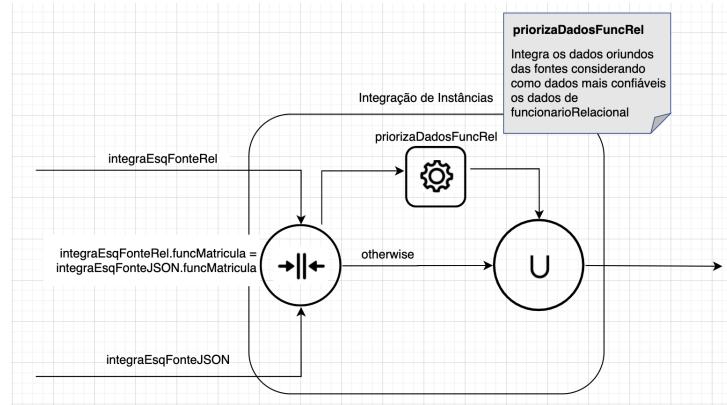


Figura 4: Diagrama conceitual para o subfluxo de “Integração de Instâncias” da **BI Solutions**.

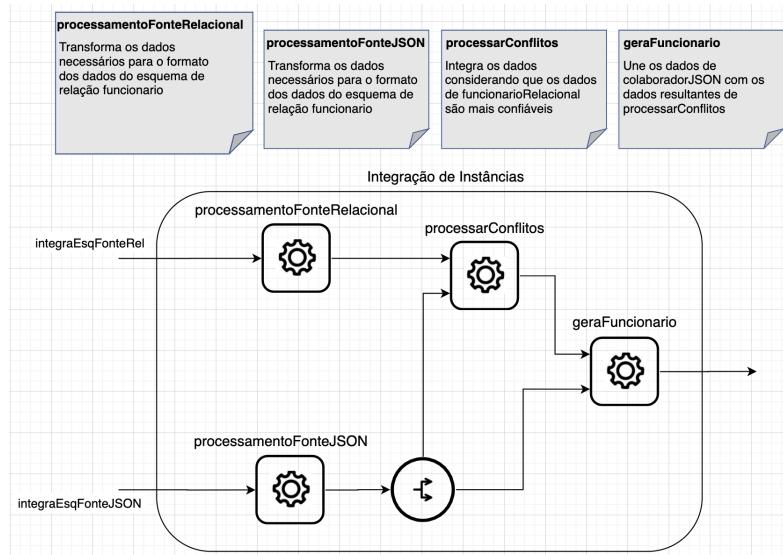


Figura 5: Workflow para o subfluxo de “Integração de Instâncias” da **BI Solutions**.

Resposta:

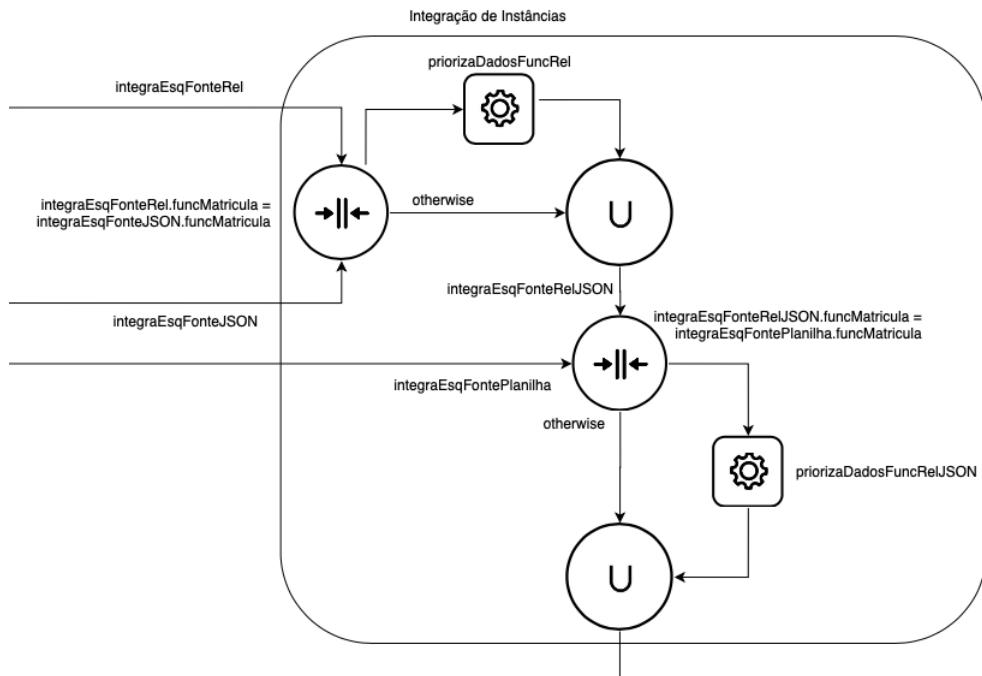


Figura 6: Resposta da primeira parte da questão 5.

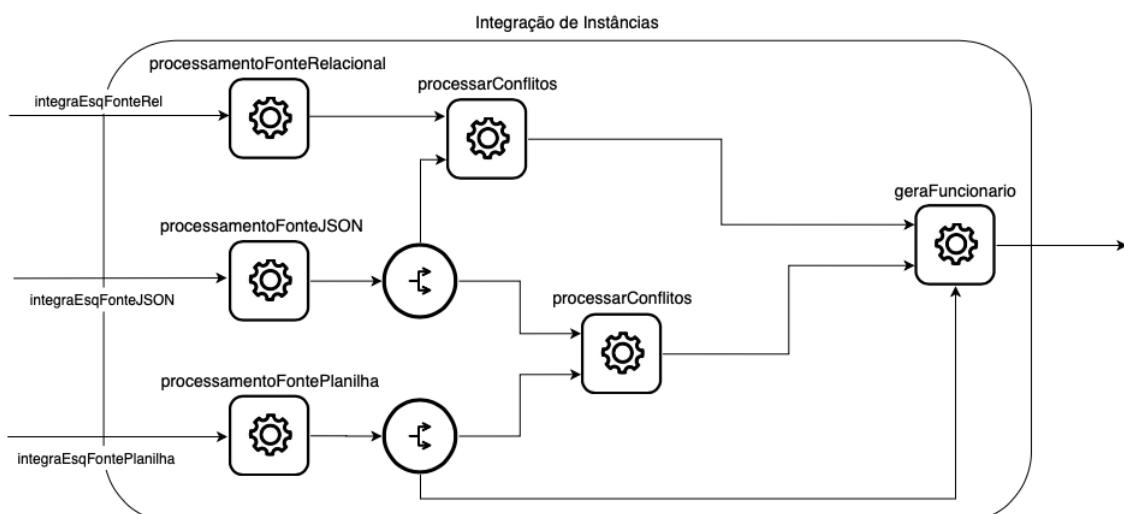


Figura 7: Resposta da segunda parte da questão 5.

7. Considere o exemplo do *data mart* apresentado nas aulas da disciplina, referente à folha de pagamento da empresa **BI Solutions**. Nele, são considerados dados de funcionários, datas, cargos e departamentos.
- (a) Escolha uma dessas perspectivas (exceto a perspectiva de funcionários, visto que esta já foi modelada nas aulas) e crie diferentes fontes de dados.
 - (b) Desenvolva um modelo conceitual para seu processo de ETL. Considere as fontes de dados do item (a) e englobe as etapas de integração de esquemas, integração de instâncias e carga em seu modelo.

Questão elaborada para gerar discussões nas tutorias. Não há uma única resposta correta.



Análise de Dados com Base em Processamento Massivo em Paralelo

Aula 5: Consultas OLAP

Cristina Dutra de Aguiar
ICMC/USP
cdac@icmc.usp.br



Agenda

- Características dos Dados
- Operações OLAP
- Sistemas ROLAP
- Exemplo usando Pandas

Data Warehouse

- Banco de dados
 - Voltado para o suporte aos processos de **gerência e tomada de decisão**
 - Análises dos usuários de SSD
 - Representam requisições **multidimensionais** aos dados do *data warehouse*
 - Visualização dos dados segundo **diferentes perspectivas**
- Dados organizados multidimensionalmente

Características dos Dados

- Integrados
 - Obtidos de fontes de dados autônomas, heterogêneas e distribuídas
 - Resultantes do processo de ETL
- Orientados a assunto
 - Relativos aos temas de negócio de maior interesse da corporação
 - Foco medidas (métricas) relacionadas à tomada de decisão

Características dos Dados

- Históricos
 - Relativos a um grande período de tempo (usualmente 5 a 10 anos)
 - Armazenados para cada mudança relevante nos dados do ambiente operacional
 - Possibilitam a realização de análises históricas
- Não voláteis
 - Permanecem estáveis por longos períodos de tempo
 - Dados não sofrem alterações, caracterizando um ambiente *load-and-access*
 - Operações de carga (*append-only*) e consultas analíticas (*read-only*)

Granularidade

- Grau de detalhamento em que os dados são armazenados
 - Dados **mais detalhados** (ou **menos agregados**)
 - Dados **menos detalhados** (ou **mais agregados**)
- Aspecto de projeto muito importante
 - **Volume** de dados do *data warehouse*
 - **Consultas** que podem ser respondidas

Tamanho do Grão

- Grão muito pequeno
 - Dados mais detalhados
 - *Data warehouse* é muito **mais volumoso**
 - Quantidade de consultas que podem ser respondidas é **maior**
- Grão muito grande
 - Dados menos detalhados
 - *Data warehouse* é **menos volumoso**
 - Quantidade de consultas que podem ser respondidas é **menor**

Aplicação 1: Área Médica

Integrados

dados integrados de pacientes, tipos de exame, hospitais nos quais os exames foram feitos e datas de coleta dos exames

Não voláteis

aplicação permite operações de carga e consultas analíticas

Demand: investigar número de pacientes

Foco: número de pacientes

Perspectivas: paciente

hospital

exame

data

Orientados a assunto
número de pacientes

Históricos

datas nas quais cada paciente realizou cada exame e hospitais correspondentes

Aplicação 1: Área Médica

- Assunto de interesse

- número de pacientes

- Granularidade

- pacientes: representados por faixa etária ←
 - exames: cada tipo de exame
 - hospitais: cada hospital da rede hospitalar
 - datas: representando dias

Análises que podem ser realizadas

Qualquer análise relacionada à faixa etária

Análises que não podem ser realizadas

Análises que requeiram dados relativos a cada idade ou dados individualizados de pacientes

Aplicação 1: Área Médica

- Assunto de interesse
 - número de pacientes
- Granularidade
 - pacientes: representados por faixa etária
 - exames: cada tipo de exame
 - hospitais: cada hospital da rede hospitalar
 - datas: representando dias

Análises que podem ser realizadas

Qualquer análise relacionada a hospital

Análises que não podem ser realizadas

Análises que requerem dados de setores do hospital



Aplicação 1: Área Médica

- Assunto de interesse
 - número de pacientes
- Granularidade
 - pacientes: representados por faixa etária
 - exames: cada tipo de exame
 - hospitais: cada hospital da rede hospitalar
 - datas: representando dias ←

Análises que podem ser realizadas

Qualquer análise relacionada ao número diário de pacientes

Análises que não podem ser realizadas

Análises relacionadas aos períodos do dia (manhã, tarde, noite)

Aplicação 2: Cadeia de Supermercados

Integrados
dados integrados de
produtos vendidos,
promoções realizadas,
filiais nas quais os
produtos foram
vendidos e **datas** das
vendas

Não voláteis
aplicação permite
operações de carga e
consultas analíticas

Demanda: investigar vendas e lucros

Foco: **unidades vendidas**
lucro obtido

Perspectivas: produto
promoção
filial
data

Orientados a assunto
unidades vendidas
lucro obtido

Históricos
datas nas quais cada
produto foi vendido
sob qual promoção
em qual filial

Aplicação 2: Cadeia de Supermercados

- Assuntos de interesse
 - unidades vendidas e lucro
- Granularidade
 - produtos: cada produto disponível
 - promoções: cada promoção realizada
 - filiais: cada filial da cadeia de supermercados
 - datas: representando dias

Análises que podem ser realizadas

Qualquer análise relacionada às filiais

Análises que não podem ser realizadas

Análises que requeiram avaliação de setores específicos

Aplicação 2: Cadeia de Supermercados

- Assuntos de interesse
 - unidades vendidas e lucro
- Granularidade
 - produtos: cada produto disponível
 - promoções: cada promoção realizada
 - filiais: cada filial da cadeia de supermercados
 - datas: representando dias ←

Análises que podem ser realizadas

Qualquer análise relacionada às vendas diárias

Análises que não podem ser realizadas

Análises no nível de cada transação de venda

Aplicação 3: Folha de Pagamento da BI Solutions



Integrados

dados integrados de **funcionários, cargos** ocupados por estes, **equipes** nas quais os funcionários trabalham e **datas** de pagamento

Não voláteis

aplicação permite operações de carga e consultas analíticas

Demanda: investigar gastos em salários

Foco: **salário**
quantidade de lançamentos

Perspectivas: funcionário
equipe
cargo
data

Orientados a assunto

salário
quantidade de lançamentos

Históricos

datas de pagamento
dos salários dos
funcionários

Aplicação 3: Folha de Pagamento da BI Solutions

- Assuntos de interesse
 - salários e quantidade de lançamentos
- Granularidade
 - funcionários: cada funcionário da empresa
 - equipe: cada equipe do funcionário
 - cargo: cada cargo da empresa
 - datas: representando dias



Análises que podem ser realizadas

Qualquer análise relacionada à área na qual a equipe trabalha

Análises que não podem ser realizadas

Qual o produto específico dentro daquela área

Aplicação 3: Folha de Pagamento da BI Solutions

- Assuntos de interesse
 - salários e quantidade de lançamentos
- Granularidade
 - funcionários: cada funcionário da empresa
 - equipe: cada equipe do funcionário
 - cargo: cada cargo da empresa
 - datas: representando dias



Análises que podem ser realizadas

Qualquer análise relacionada aos pagamentos diários

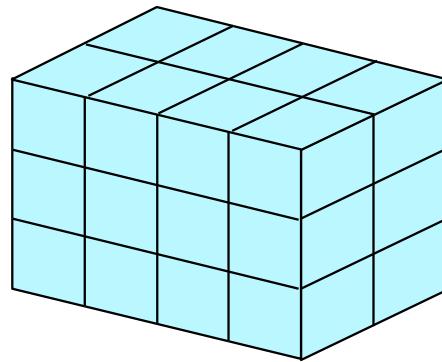
Análises que não podem ser realizadas

Identificar o valor de cada pagamento caso um funcionário receba dois pagamentos no mesmo dia

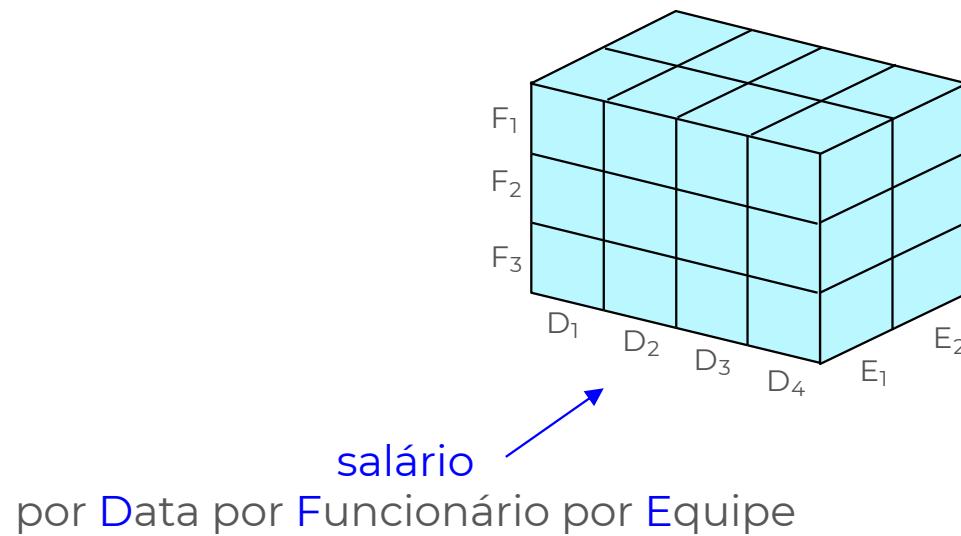
Organização em Níveis de Agregação

- Nível inferior
 - Dados **menos agregados**, os quais foram coletados do ambiente operacional
- Níveis intermediários (1 ... N)
 - Dados com graus de **agregação crescentes**
 - Nível 1: representa uma agregação do nível inferior
 - Nível N: representa uma agregação do nível N-1
- Nível superior
 - Dados **altamente agregados**, representando uma agregação do nível N

Cubo de Dados Multidimensional

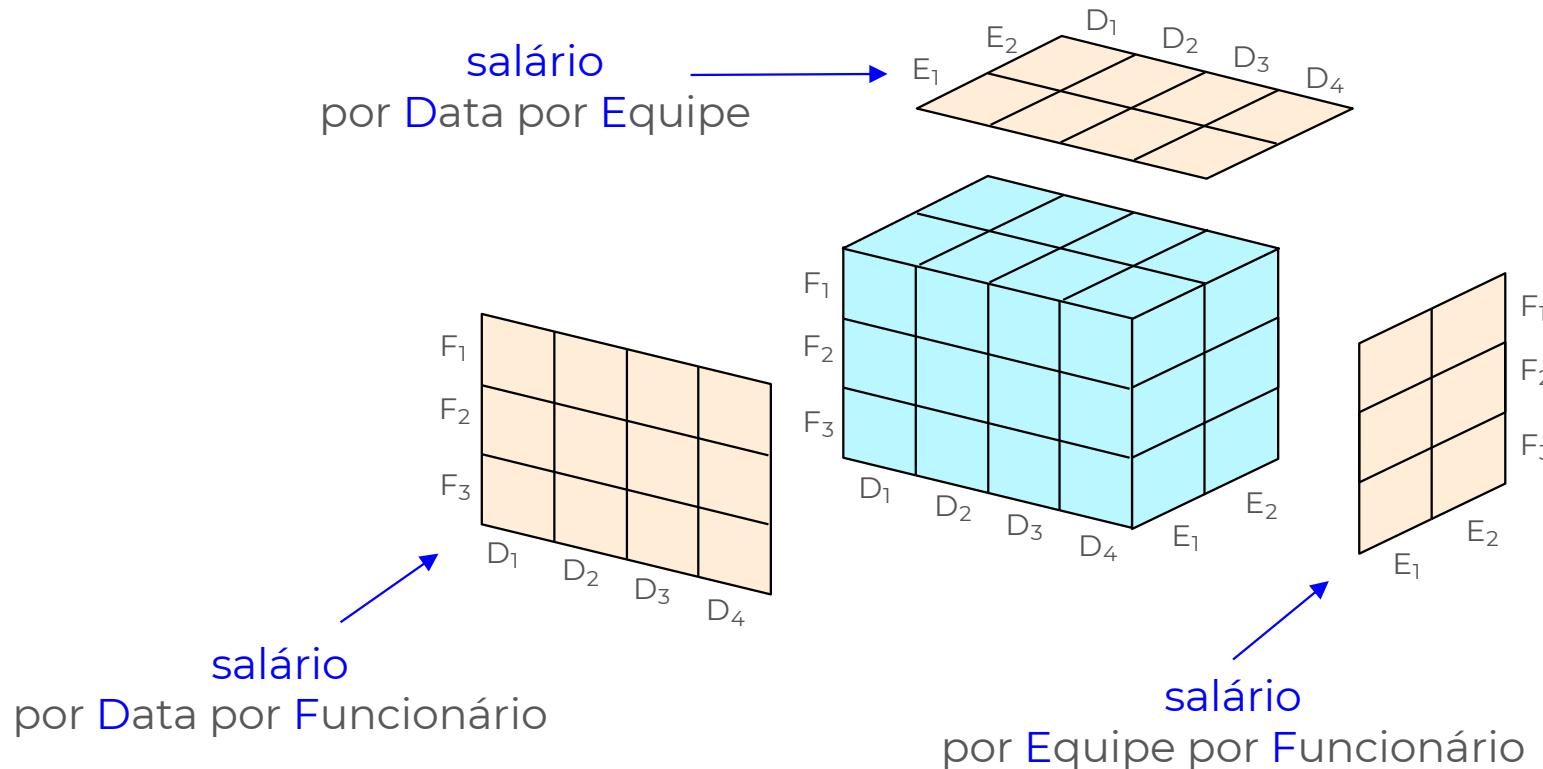


Cubo de Dados Multidimensional



- Visões multidimensionais
- nível superior
 - nível intermediário 2
 - nível intermediário 1
 - nível inferior

Cubo de Dados Multidimensional

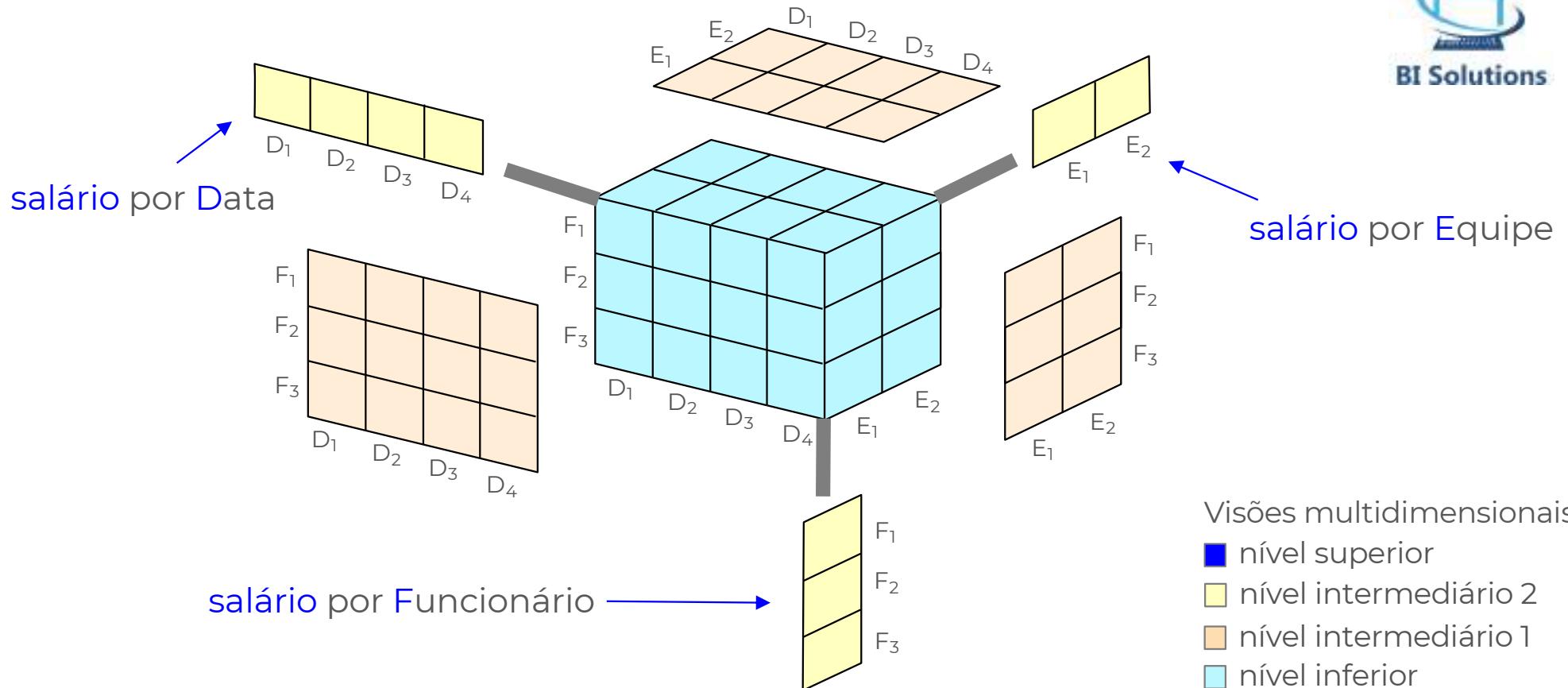


- Visões multidimensionais
- nível superior
 - nível intermediário 2
 - nível intermediário 1
 - nível inferior

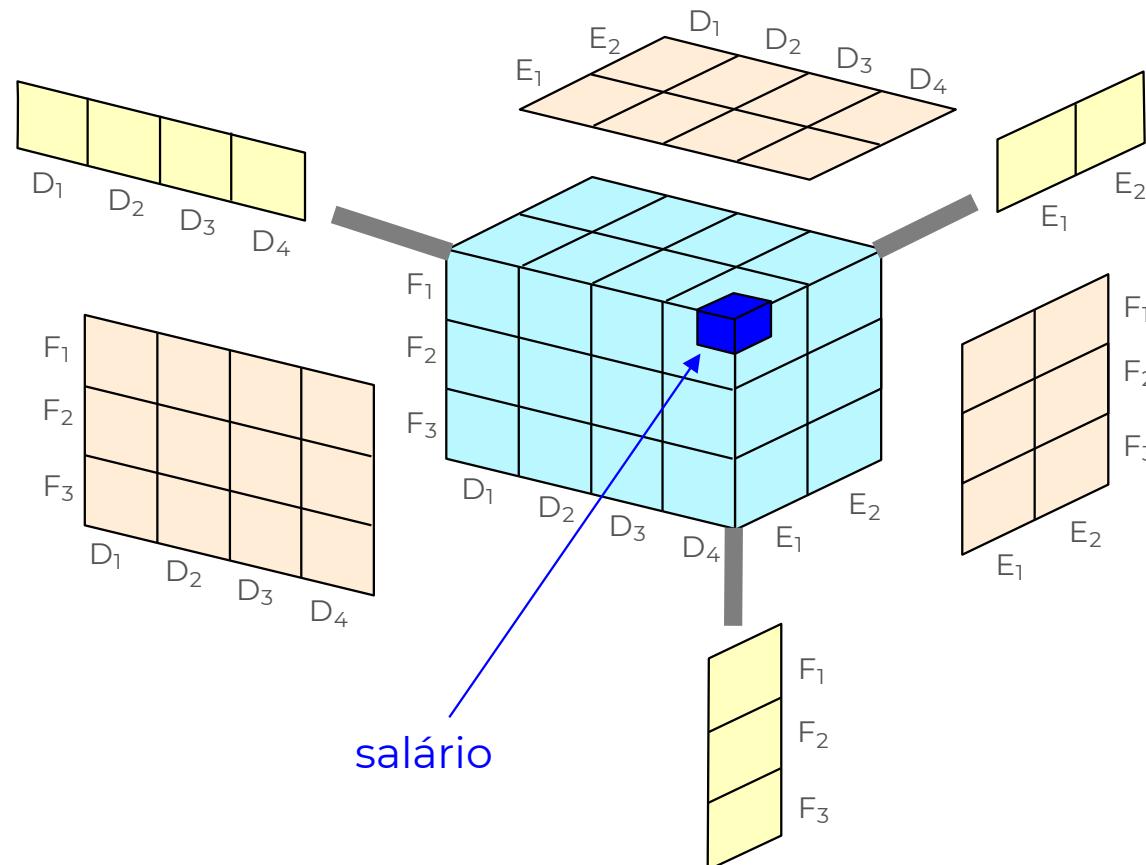
Cubo de Dados Multidimensional



BI Solutions



Cubo de Dados Multidimensional



Visões multidimensionais

- nível superior
- nível intermediário 2
- nível intermediário 1
- nível inferior

Aspectos Estáticos do Modelo Multidimensional

- Objetivam a **modelagem** dos dados em termos da estrutura desses dados
- Incluem a definição de
 - **Dimensões**
 - Determinam o contexto para as medidas numéricas
 - **Medidas numéricas**
 - Funções de suas dimensões correspondentes representando valores no espaço multidimensional

Dimensão

- Representa uma perspectiva de análise
- Composta por atributos
 - **Funcionário:** funcPK, funcMatricula, funcNome, funcSexo, funcDataNascimento, funcDiaNascimento, funcMesNascimento, funcAnoNascimento, funcCidade, funcEstadoNome, funcEstadoSigla, funcRegiaoNome, funcRegiaoSigla, funcPaisNome, funcPaisSigla
 - **Equipe:** equipePK, equipeNome, filialNome, filialCidade, filialEstadoNome, filialEstadoSigla, filialRegiaoNome, filialRegiaoSigla, filialPaisNome, filialPaisSigla
 - **Data:** dataPK, dataCompleta, dataDia, dataMes, dataBimestre, dataTrimestre, dataSemestre, dataAno



Hierarquia de Atributos

- Permite que atributos de uma dimensão relacionem-se com outros atributos da mesma dimensão
- Especifica **granularidade** dos itens de dados
 - Base para a geração dos **níveis de agregação**
 - Define as **dependências** existentes entre esses níveis
- Semântica
 - um atributo de **maior nível de granularidade** de uma hierarquia pode ser determinado usando um atributo de **menor nível de granularidade**

Hierarquia de Dados de Localização

- Atributos relacionados em dimensões

- **Funcionário:** funcCidade, funcEstadoNome, funcRegiaoNome, funcPaisNome
 - **Equipe:** filialCidade, filialEstadoNome, filialRegiaoNome, filialPaisNome

- Hierarquia de atributos

- **Cidades** podem ser agregadas em **estados**
 - **Estados** podem ser agregados em **regiões**
 - **Regiões** podem ser agregadas em **países**

Visões multidimensionais
■ nível superior

funcPaisNome: ■ nível intermediário 3
funcRegiaoNome: □ nível intermediário 2
funcEstadoNome: △ nível intermediário 1
funcCidade: ▲ nível inferior

Hierarquia de Dados de Data

- Atributos relacionados em dimensões

- **Funcionário:** dataDiaNascimento, dataMesNascimento, dataAnoNascimento
- **Data:** dataDia, dataMes, **dataBimestre**, dataTrimestre, dataSemestre, dataAno

- Hierarquia de atributos 1

- **Dias** podem ser agregados em **meses**
- **Meses** podem ser agregados em **bimestres**
- **Bimestres** podem ser agregados em **semestres**
- **Semestres** podem ser agregados em **anos**

Visões multidimensionais

■ nível superior

dataAno: ■ nível intermediário 4

dataSemestre: □ nível intermediário 3

dataBimestre: △ nível intermediário 2

dataMes: ▲ nível intermediário 1

dataDia: ▨ nível inferior

Hierarquia de Dados de Data



- Atributos relacionados em dimensões
 - **Funcionário:** dataDiaNascimento, dataMesNascimento, dataAnoNascimento
 - **Data:** dataDia, dataMes, dataBimestre, **dataTrimestre**, dataSemestre, dataAno
- Hierarquia de atributos 2
 - **Dias** podem ser agregados em **meses**
 - **Meses** podem ser agregados em **trimestres**
 - **Trimestres** podem ser agregados em **semestres**
 - **Semestres** podem ser agregados em **anos**

Visões multidimensionais

■ nível superior

dataAno: ■ nível intermediário 4

dataSemestre: ■ nível intermediário 3

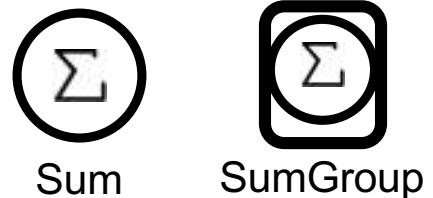
dataTrimestre: ■ nível intermediário 2

dataMes: ■ nível intermediário 1

dataDia: ■ nível inferior

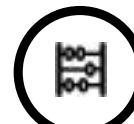
Medidas Numéricas Aditivas

- Podem ser **somadas** considerando **todas as dimensões**
- Exemplos
 - **Salário**, com a semântica de total de gastos
 - **Quantidade de lançamentos** na folha de pagamento
- São agregadas usando
 - Função de agregação **SOMA**



Medidas Numéricas Não Aditivas

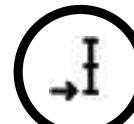
- Não podem ser **somadas**
- Exemplos
 - **Salário**, com a semântica de média salarial
 - **Frequência**, com a semântica de porcentagem de presença
- São agregadas usando
 - Funções de agregação **AGV, MAX, MIN, COUNT**
 - Outra função complexa



Count



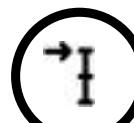
CountGroup



Min



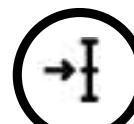
MinGroup



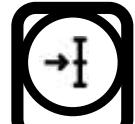
Max



MaxGroup



Avg



AvgGroup

Medidas Numéricas Semiaditivas

- Podem ser **somadas** considerando somente **algumas dimensões**
- Exemplo
 - **Número de clientes**, com a semântica de vendas de produtos
 - **Não aditiva**: para dois produtos vendidos pela mesma equipe no mesmo dia, não é possível somar o número de clientes, desde que o mesmo cliente pode estar sendo contabilizado duas vezes
 - **Aditiva**: número de clientes de um produto por dia pode ser agregado para se obter o número de clientes do mesmo produto por mês

Agenda

- Características dos Dados
- Operações OLAP
- Sistemas ROLAP
- Exemplo usando Pandas

Aspectos Dinâmicos do Modelo Multidimensional

- Representam as operações analíticas
 - Operações OLAP (*on-line analytical processing*)
- Operações típicas
 - *Drill-down* e *roll-up*
 - *Slice and dice*
 - *Pivot*
 - *Drill-across*

Visão Multidimensional Base

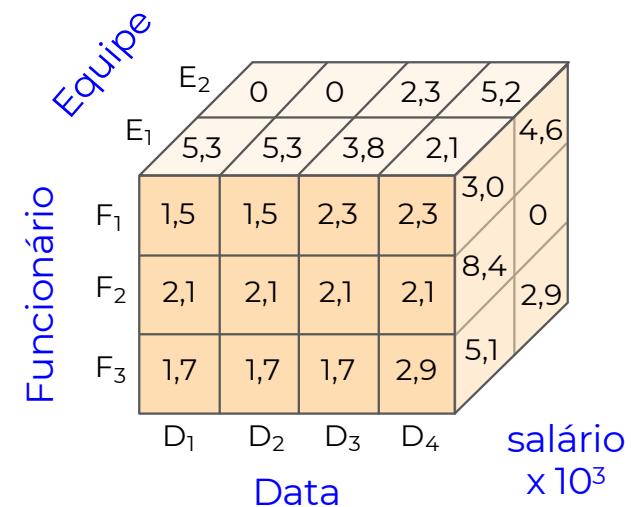
Data	Funcionário	Equipe	Salário
D1	F1	E1	1,5
D1	F2	E1	2,1
D1	F3	E1	1,7
D2	F1	E1	1,5
D2	F2	E1	2,1
D2	F3	E1	1,7
D3	F1	E2	2,3
D3	F2	E1	2,1
D3	F3	E1	1,7
D4	F1	E2	2,3
D4	F2	E1	2,1
D4	F3	E2	2,9

salário
 $\times 10^3$

■ nível inferior

função de agregação: SOMA

salário por data
por funcionário por equipe



■ nível intermediário 1

Operações Drill-Down e Roll-up

- Analisam os dados considerando níveis progressivos de agregação
- Roll-up
 - Níveis de agregação progressivamente **menos detalhados**, ou de maior granularidade
- Drill-down
 - Níveis de agregação progressivamente **mais detalhados**, ou de menor granularidade

Exemplo de Operação Roll-up

salário por **dia**
por funcionário por equipe

		Equipe					
		E ₂	0	0	2,3	5,2	
Funcionário		E ₁	5,3	5,3	3,8	2,1	4,6
F ₁	D ₁	1,5	1,5	2,3	2,3	3,0	0
	D ₂	2,1	2,1	2,1	2,1	8,4	2,9
F ₂	D ₃	1,7	1,7	1,7	2,9	5,1	
	D ₄						
salário x 10 ³							
Dia							



salário por **ano**
por funcionário por equipe

		Equipe			
		E ₂	0	7,5	
Funcionário		E ₁	10,6	5,9	4,6
F ₁	A ₁	3,0	4,6	3,0	0
	A ₂	4,2	4,2	8,4	2,9
F ₂	A ₁	3,4	4,6	5,1	
	A ₂				
Ano					
salário x 10 ³					

função de agregação: **SOMA**
D1 e D2 são agregados em A1
D3 e D4 são agregados em A2

Exemplo de Operação Drill-down

salário por **ano**
por funcionário por equipe

		Equipe	
		E ₁	E ₂
Funcionário		10,6	0
F ₁		5,9	7,5
F ₂		3,0	4,6
F ₃		8,4	2,9
	salário x 10 ³	A ₁	A ₂
		Ano	

salário por **mês**
por funcionário por equipe

		Equipe	
		E ₁	E ₂
Funcionário		5,3	0
F ₁		9,1	2,3
F ₂		2,1	4,2
F ₃		8,4	2,9
	salário x 10 ³	M ₁	M ₂
		Mês	M ₃

função de agregação: **SOMA**
D1 é agregado em M1
D2 e D3 são agregados em M2
D4 é agregado em M3

Operação Slice and Dice

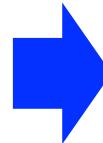
- Restringe os dados sendo analisados a um subconjunto desses dados
- Slice
 - Corte para um **valor fixo**, diminuindo a dimensionalidade do cubo
- Dice
 - Seleção de **faixas de valores**

Exemplo de Operação Slice

salário por data
por funcionário por equipe

salário por data por funcionário
para equipe = E_1

		Equipe					
		E_2	0	0	2,3	5,2	
		E_1	5,3	5,3	3,8	2,1	4,6
Funcionário		F_1	1,5	1,5	2,3	2,3	3,0
		F_2	2,1	2,1	2,1	2,1	8,4
		F_3	1,7	1,7	1,7	2,9	5,1
salário x 10 ³		Data	D ₁	D ₂	D ₃	D ₄	



		Funcionário					
		F_1	1,5	1,5	2,3	2,3	
		F_2	2,1	2,1	2,1	2,1	
Funcionário		F_3	1,7	1,7	1,7	2,9	
salário x 10 ³		Data	D ₁	D ₂	D ₃	D ₄	

Exemplo de Operação Dice

salário por data
por funcionário por equipe

		Equipe						
		E ₂	0	0	2,3	5,2		
		E ₁	5,3	5,3	3,8	2,1	4,6	
Funcionário		F ₁	1,5	1,5	2,3	2,3	3,0	0
		F ₂	2,1	2,1	2,1	2,1	8,4	2,9
		F ₃	1,7	1,7	1,7	2,9	5,1	
salário x 10 ³		Data	D ₁	D ₂	D ₃	D ₄		



salário por data por funcionário por
equipe, para datas entre D1 e D3

		Equipe					
		E ₂	0	0	2,3		
		E ₁	5,3	5,3	3,8	4,6	
Funcionário		F ₁	1,5	1,5	2,3	3,0	0
		F ₂	2,1	2,1	2,1	8,4	2,9
		F ₃	1,7	1,7	1,7	5,1	
salário x 10 ³		Data	D ₁	D ₂	D ₃		

Operação Pivot

- Oferece diferentes perspectivas dos mesmos dados
- Reorienta a visão multidimensional dos dados
 - Altera a ordem das dimensões
- Possibilita a geração de **qualquer combinação** das dimensões

Exemplo de Operação Pivot

salário por data
por funcionário por equipe

		Equipe					
		E ₁	E ₂	D ₁	D ₂	D ₃	D ₄
Funcionário	E ₁	5,3	5,3	0	0	2,3	5,2
		1,5	1,5	2,3	2,3	2,1	3,0
F ₁	2,1	2,1	2,1	2,1	8,4	0	0
	1,7	1,7	1,7	2,9	5,1	2,9	2,9



salário por funcionário
por data por equipe

		Equipe					
		E ₂	E ₁	D ₁	D ₂	D ₃	D ₄
Data	D ₁	4,6	0	2,9	0		
		3,0	8,4	5,1	5,3	0	0
D ₂	1,5	2,1	1,7	1,7	5,3	2,3	
	1,5	2,1	1,7	1,7	3,8	5,2	
D ₃	2,3	2,1	1,7	1,7	2,1	2,1	
	2,3	2,1	2,9	2,9	2,1	2,1	

Operação Drill-Across

- Compara medidas numéricas de cubos de dados diferentes
- Cubos de dados
 - Devem ser relacionados entre si por meio de **pelo menos uma dimensão em comum**
 - Devem estar no **mesmo nível de agregação**

Exemplo de Operação Drill-across

salário por data
por funcionário
por equipe

		Equipe						
		E ₂	0	0	2,3	5,2		
		E ₁	5,3	5,3	3,8	2,1	4,6	
Funcionário		F ₁	1,5	1,5	2,3	2,3	3,0	0
		F ₂	2,1	2,1	2,1	2,1	8,4	2,9
		F ₃	1,7	1,7	1,7	2,9	5,1	
salário x 10 ³		D ₁	D ₂	D ₃	D ₄			
		Data						

receita por data
por cliente por
equipe

		Equipe						
		E ₂	7,2	2,3	6,1	3,9		
		E ₁	6,6	5,7	0	8,0	5,4	
Cliente		C ₁	6,9	1,5	1,8	3,9	8,7	6,6
		C ₂	3,3	2,3	4,3	4,1	7,4	7,5
		C ₃	3,6	4,2	0	3,9	4,2	
receita x 10 ³		D ₁	D ₂	D ₃	D ₄			
		Data						

		Equipe				
		E ₁	5,3	5,3	3,8	2,1
		E ₂	0	0	2,3	5,2
salário x 10 ³		D ₁	D ₂	D ₃	D ₄	
		Data				

salário por data
por equipe

		Equipe				
		E ₁	6,6	5,7	0	8,0
		E ₂	7,2	2,3	6,1	3,9
receita x 10 ³		D ₁	D ₂	D ₃	D ₄	
		Data				

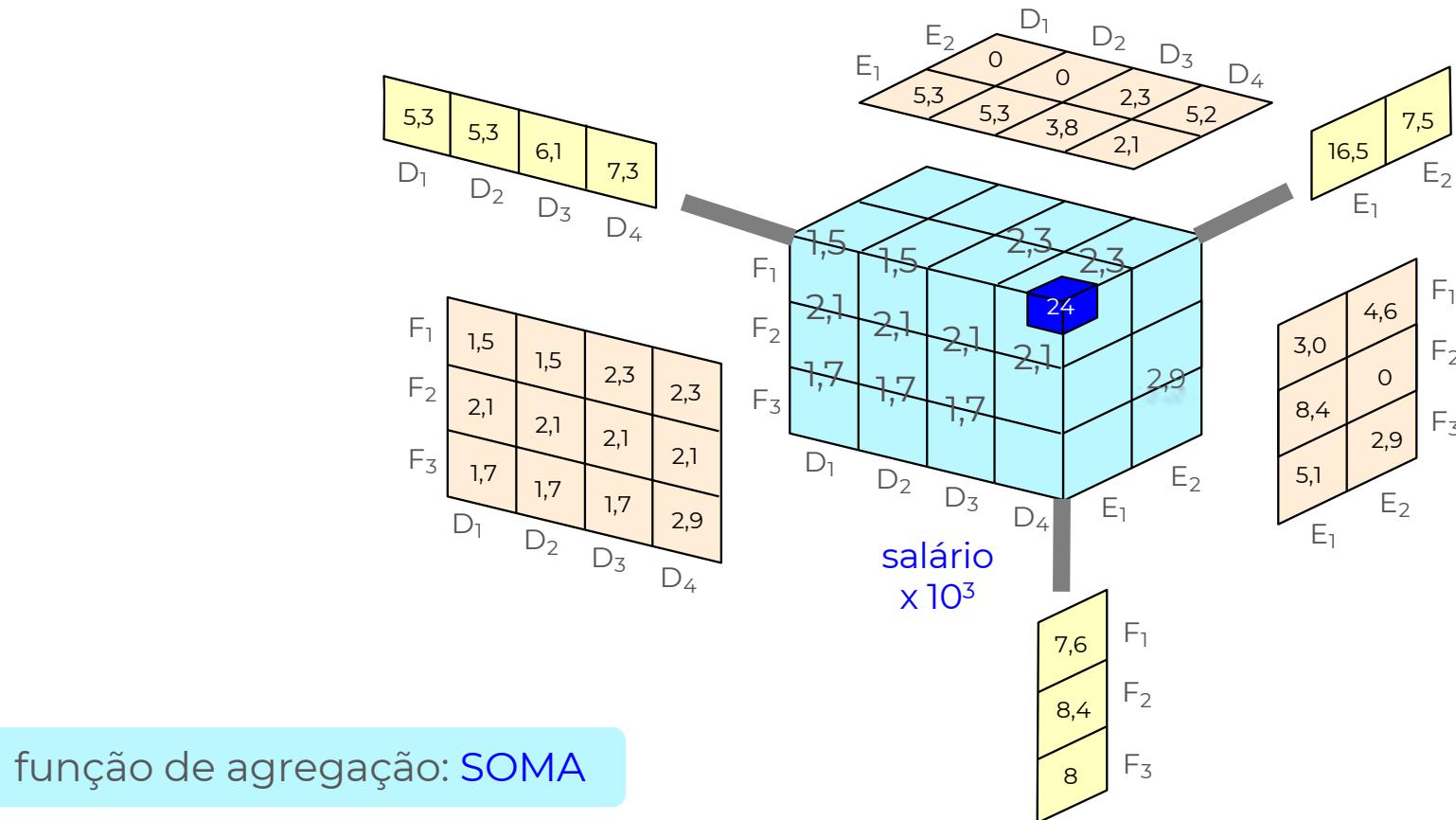
receita por data
por equipe

salário, receita
por data por equipe

		Equipe				
		E ₁	5,3	5,3	3,8	2,1
		E ₂	6,6	5,7	0	8,0
salário, receita x 10 ³		D ₁	D ₂	D ₃	D ₄	
		Data				

função de agregação:
SOMA

Cubo de Dados Multidimensional



função de agregação: SOMA

Agenda

- Características dos Dados
- Operações OLAP
- Sistemas ROLAP
- Exemplo usando Pandas

Sistemas ROLAP (OLAP Relacional)

- Baseado no uso do [modelo relacionais](#)
- *Data Warehouse*
 - Representado como uma [coleção de esquemas de relação](#)
- Sistema gerenciador de banco de dados (SGBD)
 - Estendido para oferecer suporte às operações analíticas

Modelo Relacional

- Esquema de relação
 - Possui um **nome único** e um conjunto de atributos
 - Consiste de uma **tabela bidimensional**
- Características da tabela bidimensional
 - Cada **coluna** tem um nome distinto e representa um **atributo**
 - Cada atributo possui um **domínio**
 - Todos os valores de uma coluna são valores do **mesmo atributo**
 - Cada **linha** é distinta e representa uma **instância** (ou tupla)
 - A **ordem** das colunas e das linhas é **irrelevante**

Restrições sobre um Esquema de Relação

- Domínio de cada atributo
 - Deve ser **atômico**
 - Pode possuir valores **nulos**
- Chave primária
 - Identifica de forma **única** cada tupla de cada esquema de relação
- Integridade de entidade
 - Nenhum valor de chave primária pode ser nulo

Exemplo para a Dimensão Funcionário

funcionario (funcPK, funcMatricula, funcNome, funcSexo, funcDataNascimento,
funcDiaNascimento, funcMesNascimento, funcAnoNascimento,
funcCidade, funcEstadoNome, funcEstadoSigla, funcRegiaoNome,
funcRegiaoSigla, funcPaisNome, funcPaisSigla)



funcPK	funcMatricula	funcNome	funcSexo	funcDataNascimento	funcDiaNascimento	funcMesNascimento	...
1	M-1	ALINE ALMEIDA	F	1/1/1990	1	1	...
2	M-2	ARAO ALVES	M	2/2/1990	2	2	...
3	M-3	ARON ANDRADE	M	3/3/1990	3	3	...
4	M-4	ADA BARBOSA	F	4/4/1990	4	4	...
5	M-5	ABADE BATISTA	M	5/5/1990	5	5	...
6	M-6	ABADE BARROS	M	6/6/1990	6	6	...
7	M-7	ABADIA BORGES	F	7/7/1990	7	7	...
...

Restrições sobre dois Esquemas de Relação

- Integridade referencial
 - Mantém a **consistência** entre as tuplas presentes em dois esquemas de relação
 - Declara que uma tupla em uma primeira tabela, a qual faz referência a uma outra tabela, deve se referir a uma tupla existente nessa segunda tabela
- Relacionamento
 - Primeiro esquema de relação: **chave primária (PK)**
 - Segundo esquema de relação: **chave estrangeira (FK)**

Exemplo para Funcionário e Pagamento

funcionário (funcPK, funcMatricula, funcNome, funcSexo, funcDataNascimento, funcDiaNascimento, funcMesNascimento, funcAnoNascimento, funcCidade, funcEstadoNome, funcEstadoSigla, funcRegiaoNome, funcRegiaoSigla, funcPaisNome, funcPaisSigla)

pagamento (funcData, funcPK, funcEquipe, funcCargo, salario, quantidadeLancamento)

funcionario

funcPK	funcMatricula	funcNome	...
1	M-1	ALINE ALMEIDA	...
2	M-2	ARAO ALVES	...
3	M-3	ARON ANDRADE	...
4	M-4	ADA BARBOSA	...
5	M-5	ABADE BATISTA	...
6	M-6	ABADE BARROS	...
...

pagamento

dataPK	funcPK	funcEquipe	funcCargo	salario	quantidadeLancamento
1	1	7	112	2.226,66	1
1	3	2	74	9.169,90	1
2	6	7	43	5.784,28	1
5	5	2	112	2.226,66	1
5	2	1	74	9.169,90	1
7	1	3	112	3.828,90	1
...

Esquema do Data Warehouse

- Definido em termos de esquemas de relação
 - Organizado especialmente para refletir a visão multidimensional dos dados
- Tipos de esquema
 - Estrela (*star*)
 - Foco de neve (*snowflake*)
 - Estrela-foco (*starflake*)

Tipos de Tabela

Tabela de Fatos

Tabela de Dimensão

Tabela de Fatos

- Localizada visualmente no **centro** da estrela
- Armazena
 - As medidas numéricas relevantes ao negócio (**fatos**)
 - Uma **chave estrangeira (FK)** para cada tabela de dimensão
 - Uma **chave primária (PK)** composta pela combinação das chaves estrangeiras
- Características
 - Usualmente fina e longa
 - Sem redundância
 - Sem dados esparsos

Tabela de Dimensão

- Localizada visualmente na **extremidade** da estrela
- Armazena
 - Uma **chave primária** (**chave artificial**)
 - Atributos da dimensão
- Características
 - Usualmente larga e curta
 - Com redundância

Folha de Pagamento da BI Solutions



BI Solutions

Demanda: investigar gastos em salários

Foco: **salário**

quantidade de lançamentos

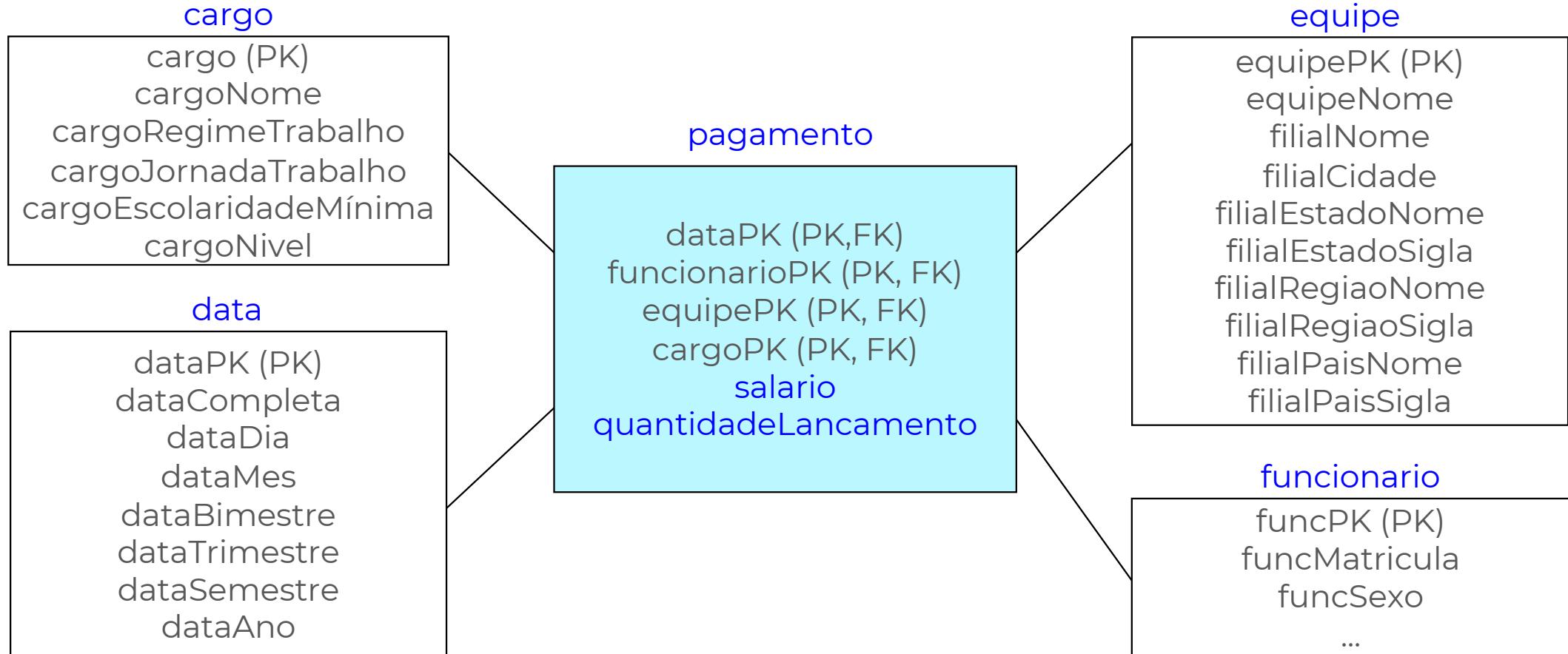
Perspectivas: funcionário

equipe

cargo

data

Esquema Estrela Pagamento



Esquema Estrela



Esquema Estrela Pagamento

cargo
cargo (PK)
cargoNome
cargoRegimeTrabalho
cargoJornadaTrabalho
cargoEscolaridadeMínima
cargoNivel

data
dataPK (PK)
dataCompleta
dataDia
dataMes
dataBimestre
dataTrimestre
dataSemestre
dataAno

TABELA DE FATOS

pagamento

dataPK (PK,FK)
funcionarioPK (PK, FK)
equipePK (PK, FK)
cargoPK (PK, FK)
salario
quantidadeLancamento

equipe
equipePK (PK)
equipeNome
filialNome
filialCidade
filialEstadoNome
filialEstadoSigla
filialRegiaoNome
filialRegiaoSigla
filialPaisNome
filialPaisSigla

funcionario
funcPK (PK)
funcMatricula
funcSexo
...

Esquema Estrela Pagamento



Esquema Relacional Pagamento

data (dataPK, dataCompleta, dataDia, dataMes, dataBimestre, dataTrimestre, dataSemestre, dataAno)



funcionario (funcPK, funcMatricula, funcNome, funcSexo, funcDataNascimento, funcDiaNascimento, funcMesNascimento, funcAnoNascimento, funcCidade, funcEstadoNome, funcEstadoSigla, funcRegiaoNome, funcRegiaoSigla, funcPaisNome, funcPaisSigla)

equipe (equipePK, equipeNome, filialNome, filialCidade, filialEstadoNome, filialEstadoSigla, filialRegiaoNome, filialRegiaoSigla, filialPaisNome, filialPaisSigla)

cargo (cargoPK, cargoNome, cargoRegimeTrabalho, cargoEscolaridadeMinima, cargoNivel)

pagamento (dataPK, funcPK, equipePK, cargoPK, salario, quantidadeLancamento)

Negociação da BI Solutions

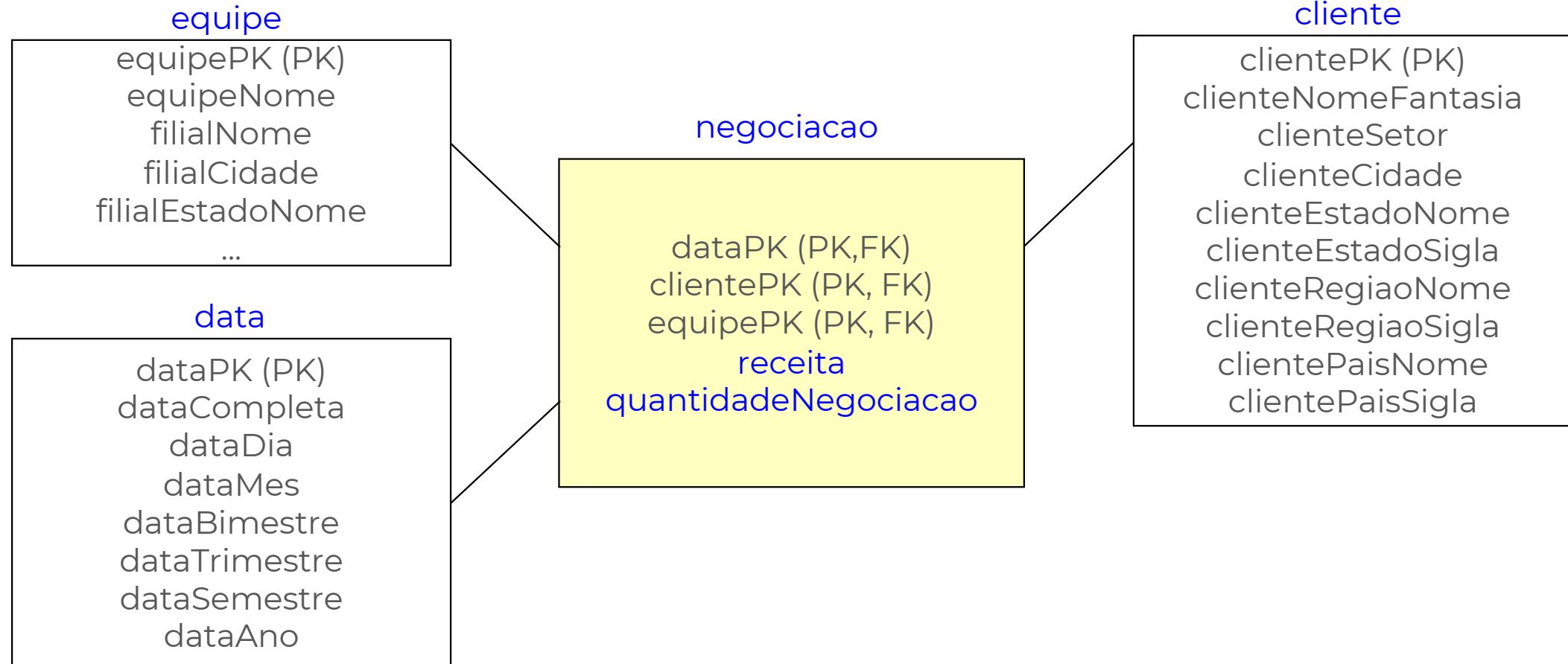


Demanda: investigar receitas
recebidas pelas equipes

Foco: **receita**
quantidade de negociações

Perspectivas: equipe
cliente
data

Esquema Estrela Negociação



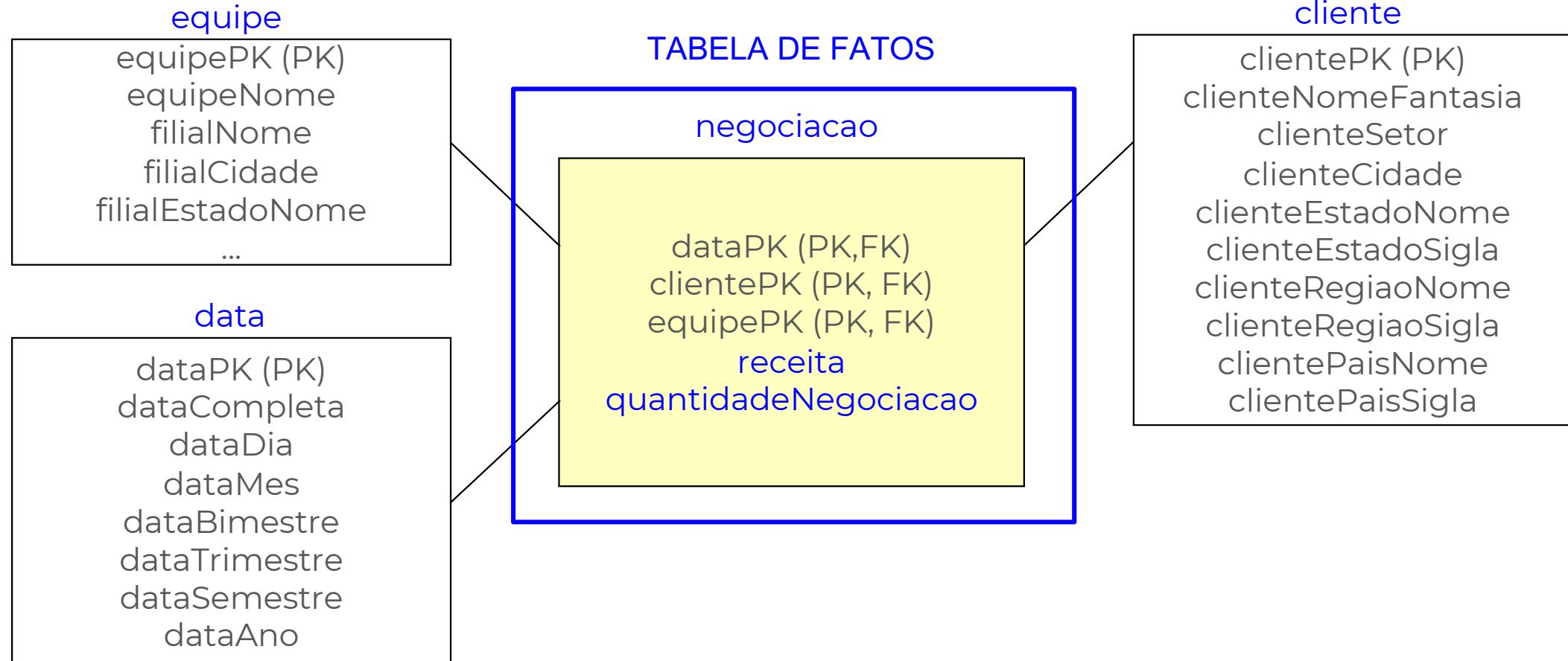
Esquema Estrela Negociação



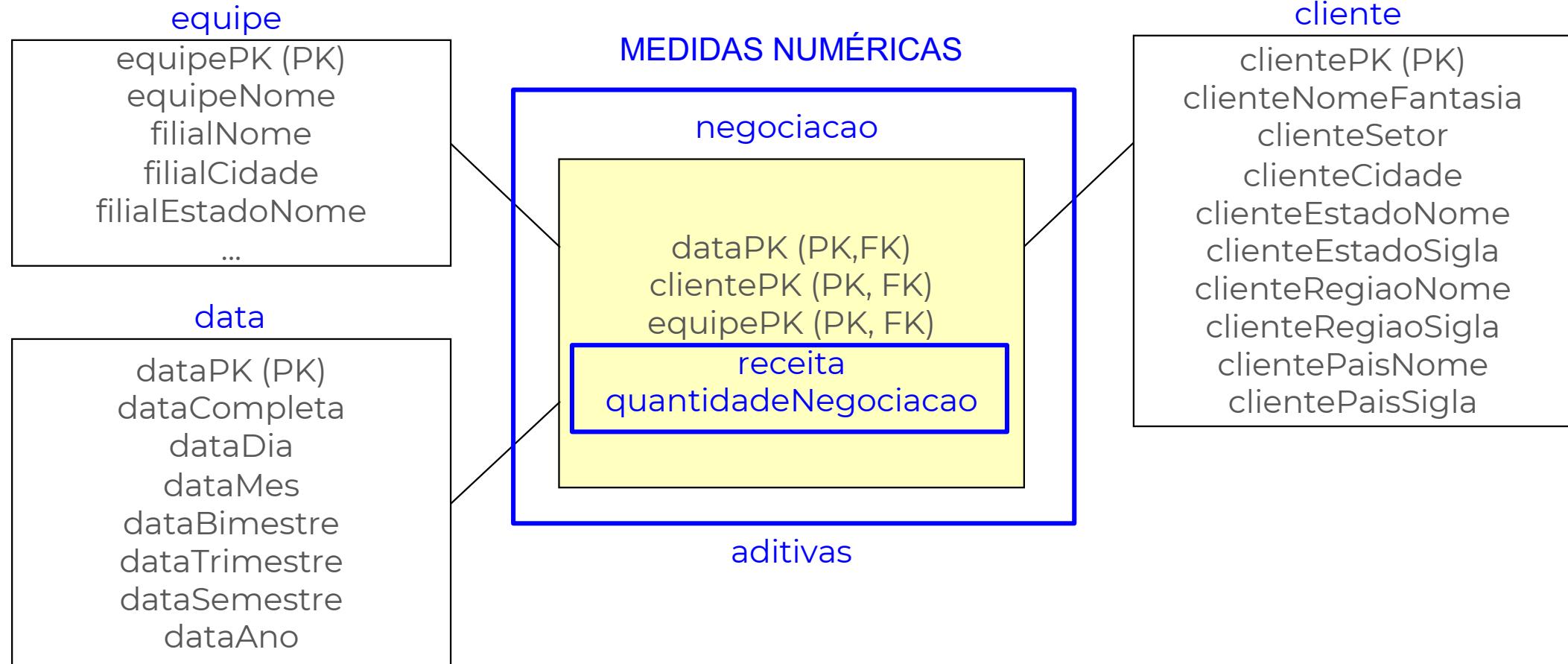
Esquema Estrela Negociação



Esquema Estrela Negociação



Esquema Estrela Negociação



Esquema Relacional Negociação

data (dataPK, dataCompleta, dataDia, dataMes, dataBimestre, dataTrimestre, dataSemestre, dataAno)



equipe (equipePK, equipeNome, filialNome, filialCidade, filialEstadoNome, filialEstadoSigla, filialRegiaoNome, filialRegiaoSigla, filialPaisNome, filialPaisSigla)

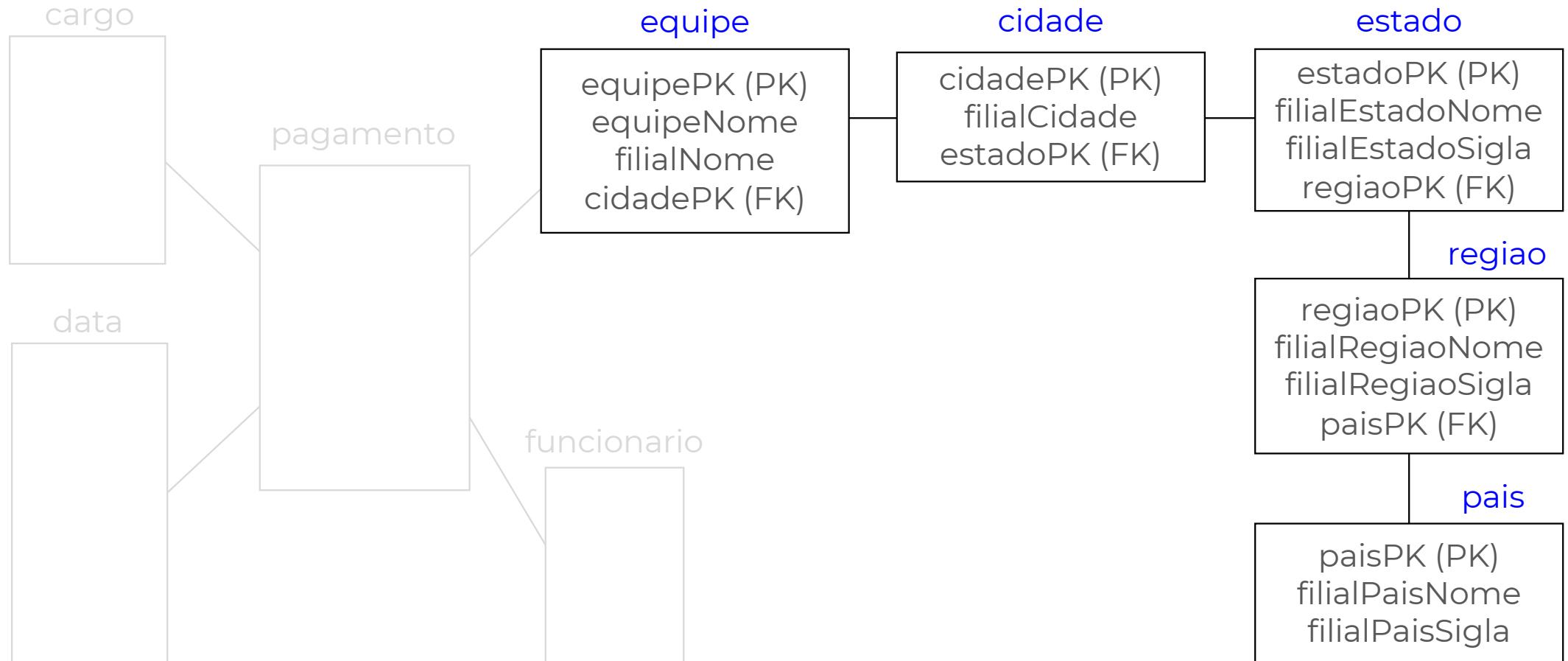
cliente (clientePK, clienteNomeFantasia, clienteSetor, clienteCidade, clienteEstadoNome, clienteEstadoSigla, clienteRegiaoNome, clienteRegiaoSigla, clientePaisNome, clientePaisSigla)

negociacao (dataPK, equipePK, clientePK, receita, quantidadeNegociacao)

Esquema Floco de Neve

- Extensão do esquema estrela
- Tabelas de dimensão
 - Normalizadas com base nas hierarquias de atributos
 - Projetadas para evitar redundância dos dados
- Redundância
 - Melhora o desempenho no processamento de consultas OLAP
 - Requer maior espaço de armazenamento

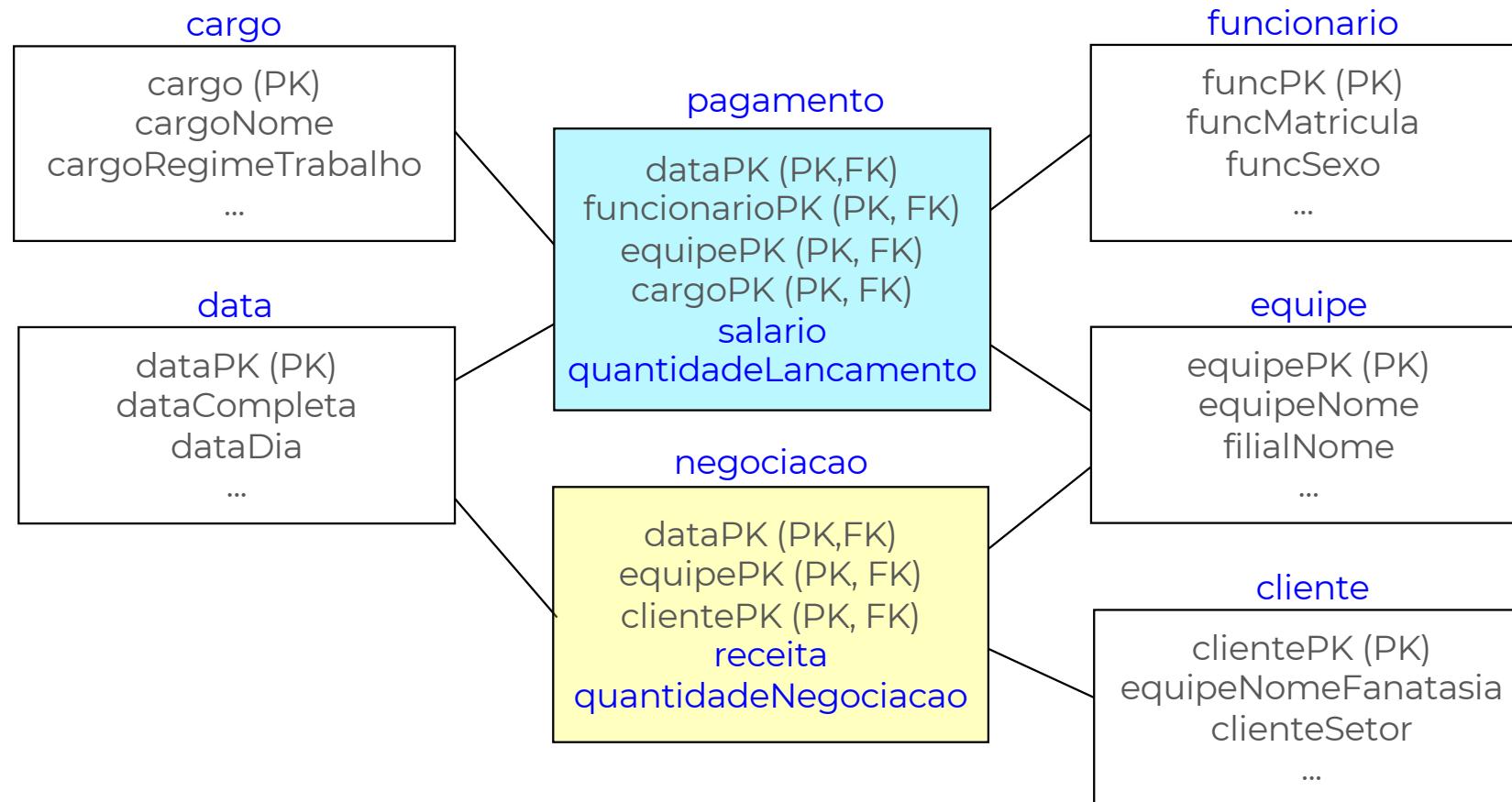
Normalização da Tabela de Dimensão Equipe



Esquema Estrela-Floco

- Extensão dos esquemas
 - Estrela
 - Floco de Neve
- Tabelas de dimensão
 - Algumas tabelas são **desnormalizadas** (contêm dados redundantes)
 - Algumas tabelas são **normalizadas** (não contêm dados redundantes)

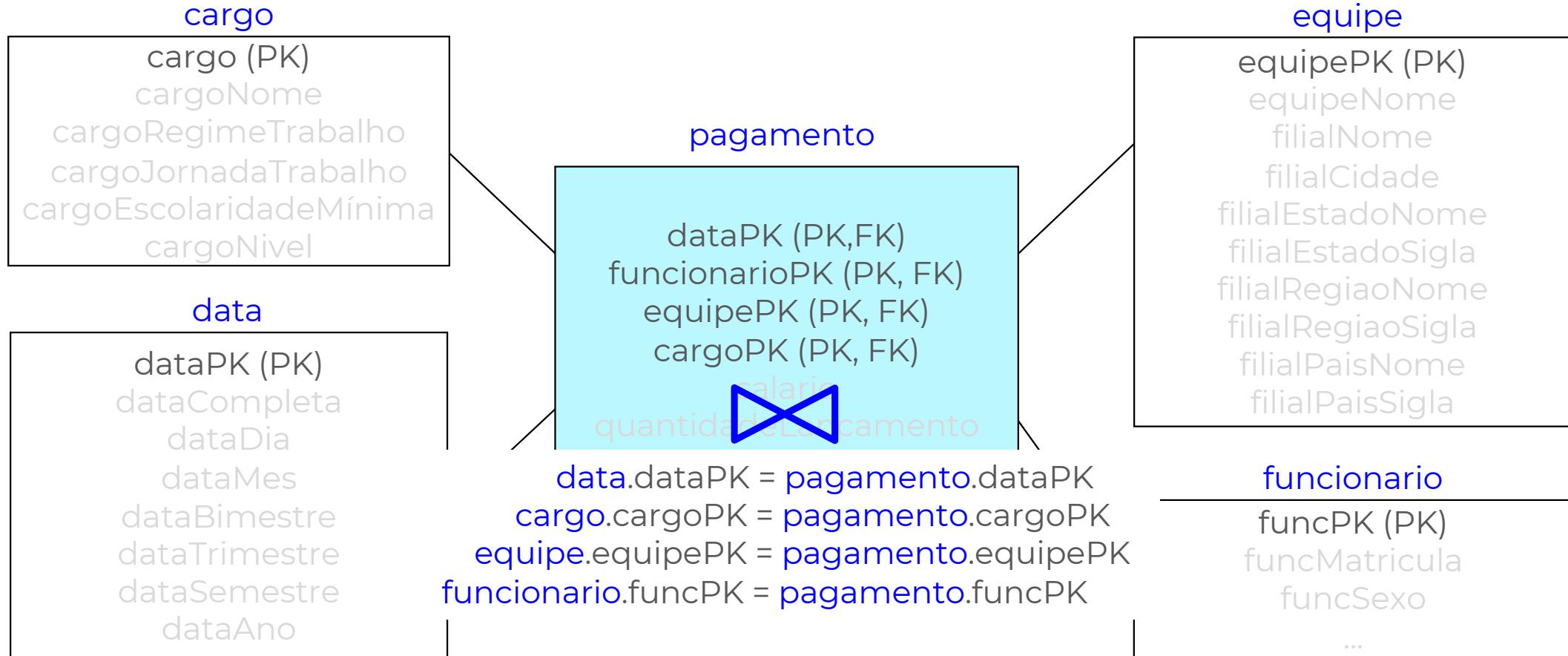
Constelação de Fatos



Junção Estrela

- Operação necessária devido à organização dos dados
 - Segundo os tipos de esquema estrela, flocos de neve ou estrela-foco
- Dada uma consulta OLAP, consiste em
 - Acessar a **tabela de fatos** e todas as **tabelas de dimensão** envolvidas
 - Realizar as **junções** necessárias
 - Base na **integridade referencial**, isto é, **pares (chave estrangeira, chave primária)**
 - Representação gráfica da junção: 

Esquema Estrela Pagamento



Exemplo para Funcionário e Pagamento

funcionario

funcPK	funcMatricula	funcNome	...
1	M-1	ALINE ALMEIDA	...
2	M-2	ARAO ALVES	...
3	M-3	ARON ANDRADE	...
4	M-4	ADA BARBOSA	...
5	M-5	ABADE BATISTA	...
6	M-6	ABADE BARROS	...
...

pagamento

dataPK	funcPK	funcEquipe	funcCargo	salario	quantidadeLancamento
1	1	7	112	2.226,66	1
1	2	2	74	9.169,90	1
2	6	7	43	5.784,28	1
5	5	2	112	2.226,66	1
5	2	1	74	9.169,90	1
7	1	3	112	3.828,90	1
...

funcionario  pagamento (funcionario.funcPK = pagamento.funcPK)

funcionario.funcPK	funcMatricula	funcNome	...	dataPK	funcEquipe	funcCargo	salario	quantidadeLancamento
1	M-1	ALINE ALMEIDA	...	1	7	112	2.226,66	1
1	M-1	ALINE ALMEIDA	...	7	3	112	3.828,90	1
2	M-2	ARAO ALVES	...	1	2	74	9.169,90	1
2	M-2	ARAO ALVES	...	5	1	74	9.169,90	1
5	M-5	ABADE BATISTA	...	5	2	112	2.226,66	1
6	M-6	ABADE BARROS	...	2	7	43	5.784,28	1
...

Análise de Dados com Base em Processamento Massivo em Paralelo

Aula 6: Processamento Paralelo e Distribuído

Cristina Dutra de Aguiar
ICMC/USP
cdac@icmc.usp.br



Agenda

- Ambientes Computacionais
- Modelo MapReduce
- Apache Spark RDD

Ambientes Computacionais

- Características
 - Grande capacidade de armazenamento e processamento
 - Suporte para a manipulação de *big data*
- Tipos
 - *Cluster* de computadores
 - Ambiente de computação em nuvem

Cluster de Computadores

- Popularizado em meados da década de 90
- Modelo composto por uma coleção de computadores
 - Chamados de **nós**
 - Dispostos de forma **paralela** e **distribuída**
 - **Interconectados** por redes de alta velocidade

Características

- Processamento
 - **Recebimento** das tarefas
 - **Divisão** das tarefas entre os nós
 - **Execução simultânea** das tarefas nos nós
- Transparência
 - *Cluster* deve ser visto como um **único computador**
- Computadores
 - **Hardware** não precisa ser exatamente igual em cada nó
 - Podem ser **dedicados** ou não

Software

- Sistema operacional
 - Todos os nós devem utilizar o mesmo sistema operacional
 - Objetivos
 - Diminuir a complexidade de configuração e manutenção
 - Facilitar monitoramento, distribuição de tarefas e controle de recursos
- Middleware ←
 - Sistema que permite o gerenciamento do cluster
 - Diretamente relacionado ao sistema operacional
 - Oferece uma interface para configuração

instalado em uma
máquina chamada
nó mestre
(nó controlador)

Tipos de Cluster

- Voltados a atender aos requisitos específicos das aplicações
 - *Cluster de alto desempenho*
 - *Cluster de alta disponibilidade*
 - *Cluster para balanceamento de carga*
- Combinação de tipos de cluster
 - *Suprir às demandas das aplicações*
 - Exemplo
 - *cluster de alta disponibilidade* para funcionamento 24 x 7
 - *cluster de balanceamento de carga* para garantir eventual aumento de requisições em períodos de pico

Cluster de Alto Desempenho

- Voltado à computação intensiva
 - Realiza o **processamento coletivo** de uma única **tarefa computacional complexa**
- Os nós devem, idealmente
 - Ser majoritariamente **homogêneos** em termos de arquitetura de processadores e de sistema operacional
 - **Compartilhar** uma rede dedicada
 - Estar **acoplados** de forma próxima

Cluster de Alta Disponibilidade

- O sistema deve **funcionar** continuamente
 - Mesmo em caso de eventuais **falhas**
- Manutenção de nós redundantes
 - Quando um nó falha, ele é **substituído** por outro nó sem prejuízo
- Recursos que podem ser utilizados
 - Ferramentas de monitoramento dos nós para investigar falhas
 - Replicação (redundância) de sistemas
 - Computadores para substituição imediata de máquinas com problemas
 - Uso de geradores para garantir o funcionamento do sistema

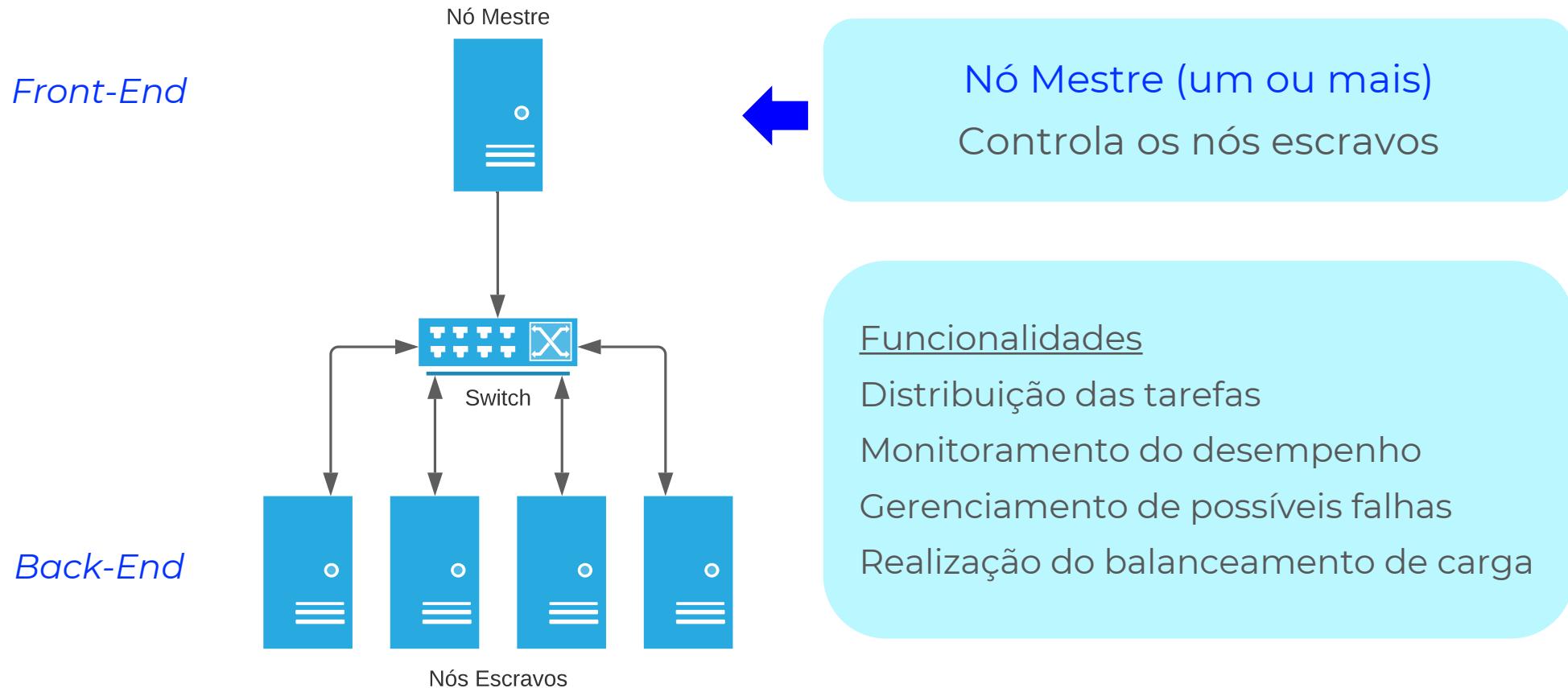
Cluster para Balanceamento de Carga

- Divisão o mais **uniforme** possível das tarefas entre os nós
 - Cada nó deve receber e atender a uma tarefa
 - Nós não devem, necessariamente, dividir uma tarefa com outros nós
- Recursos que podem ser utilizados
 - Ferramentas de monitoramento dos nós para analisar a carga
 - Direcionamento de tarefas para nós que possuam menor quantidade de tarefas

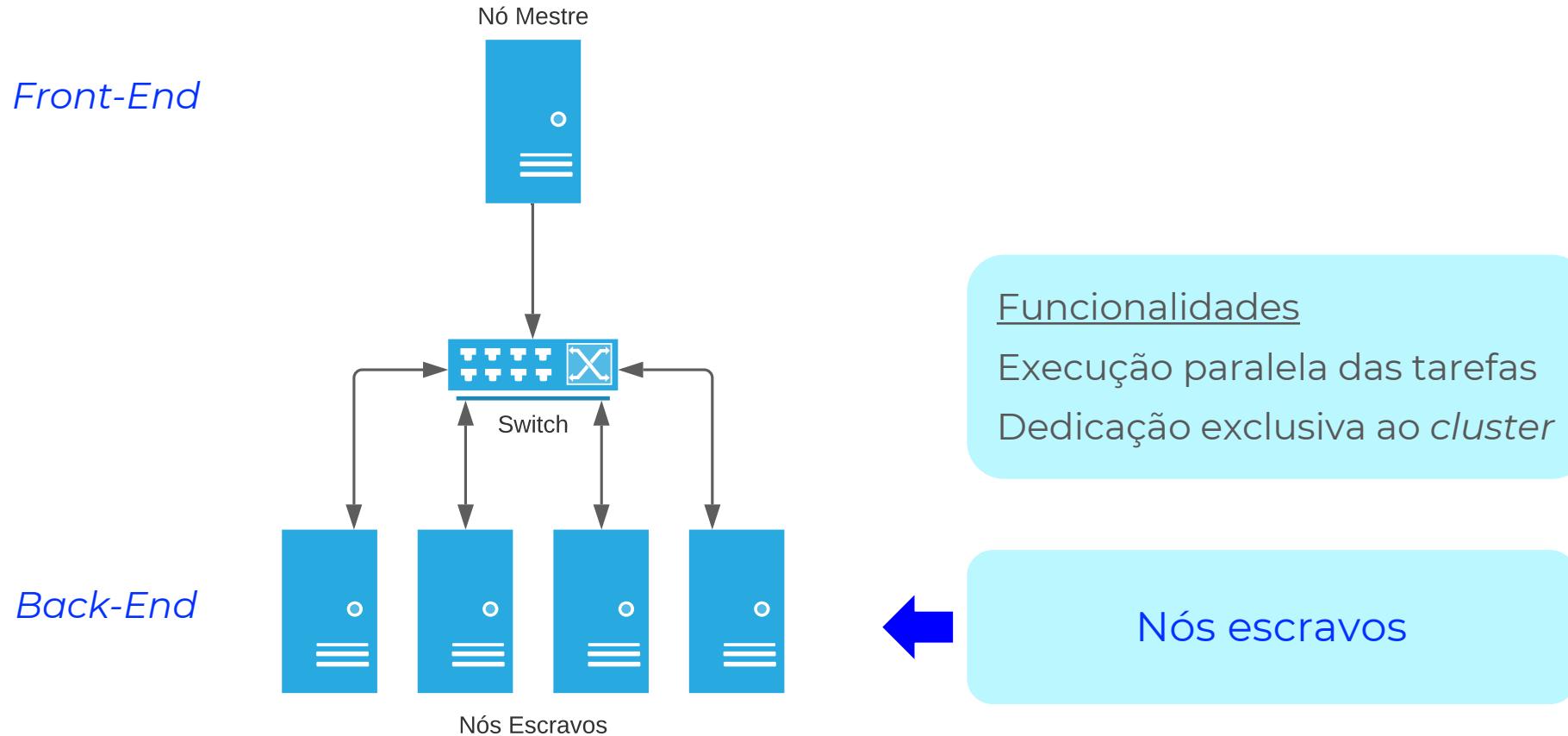
Cluster Beowulf

- Padrão de *cluster* disponibilizado pela NASA em 1994
- Características
 - A **conexão** entre os nós pode ser feita por **redes Ethernet**
 - Não é necessário o uso de **hardware** específico ou potente
 - **Sistema operacional**
 - Deve ser baseado em código aberto
 - Deve conter as ferramentas necessárias para a configuração do *cluster*

Cluster Beowulf: Nós Mestre e Nós Escravos



Cluster Beowulf: Nós Mestre e Nós Escravos



Dificuldades de Cluster de Computadores

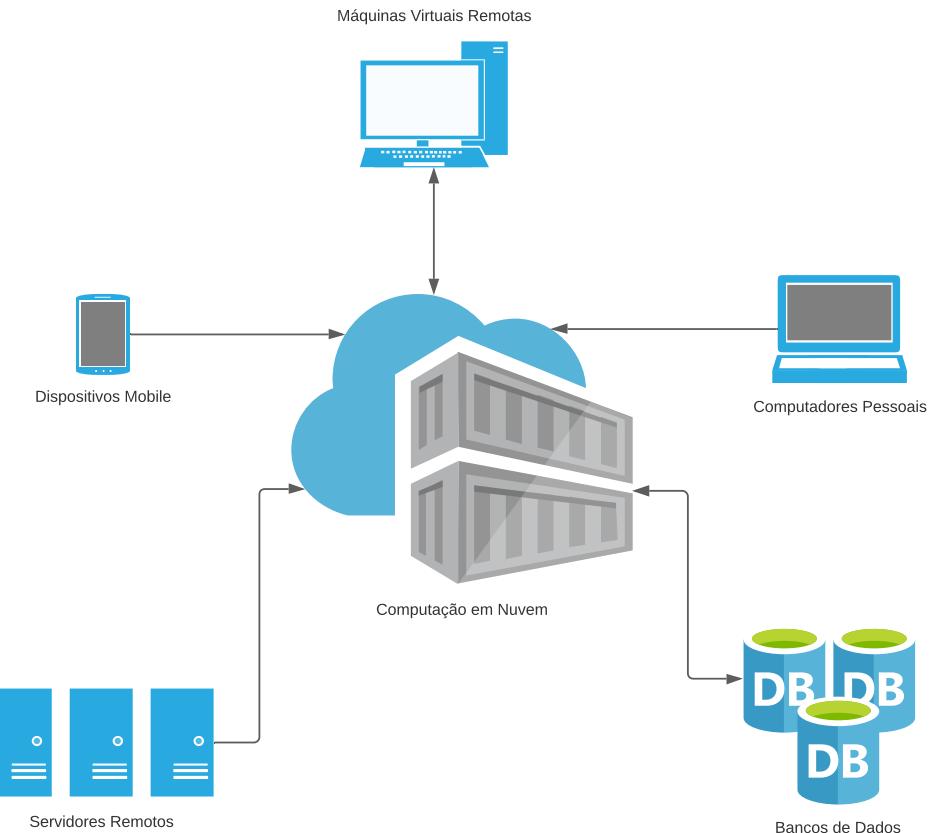
- Usuários
 - Enxergam o *cluster* como uma coleção de computadores independentes
- Manipulação do *cluster* pode ser uma tarefa complexa
 - Manipulação dos componentes
 - Divisão de tarefas entre os nós
 - Gerenciamento da comunicação entre os nós

Computação em Nuvem

- Modelo que possibilita acesso a
 - Recursos computacionais compartilhados e interligados via rede
- Exemplos de recursos
 - Redes, servidores, equipamentos de armazenamento
 - Aplicações, serviços
- Nuvem
 - Abstração que oculta a complexidade de infraestrutura

Abstração Nuvem

Metáfora para
Internet ou infraestrutura
de comunicação entre
ambientes computacionais



Definição Segundo NIST

Computação em nuvem é um modelo que permite acesso ubíquo, conveniente e sob demanda via rede a um conjunto compartilhado e configurável de recursos computacionais que pode ser rapidamente fornecido e liberado com esforços mínimos de gerenciamento ou interação com o provedor de serviços

Composição Segundo NIST

- Tecnologias chave
 - Redes de longa distância rápidas
 - Computadores servidores poderosos e/ou baratos
 - Virtualização de alto desempenho
- Ambiente de nuvem
 - Cinco características essenciais
 - Três modelos de serviços
 - Quatro modelos de implantação

Composição Segundo NIST

- Tecnologias chave
 - Redes de longa distância rápidas
 - Computadores servidores poderosos e/ou baratos
 - Virtualização de alto desempenho
- Ambiente de nuvem
 - Cinco características essenciais
 - Três modelos de serviços
 - Quatro modelos de implantação

Características Essenciais

- Serviço sob demanda
 - Recursos são acessados de forma direta e **sob demanda**
 - Alocação e liberação de recursos ocorre **sem interação** entre o usuário e o provedor
 - Usuários têm interação mínima com o provedor
- Serviço sob demanda
 - Recursos são acessados de forma direta e **sob demanda**
 - Alocação e liberação de recursos ocorre **sem interação** entre o usuário e o provedor
 - Usuários têm interação mínima com o provedor

Características Essenciais

- Compartilhamento de recursos
 - Recursos são agrupados e **compartilhados** entre diversos usuários
 - Usuários não precisam ter conhecimento acerca da localização dos recursos
- Rápida elasticidade
 - Recursos são **rapidamente alocados e liberados** a qualquer momento
 - Aplicação pode demandar **qualquer quantidade** de recursos
 - Usuários têm a sensação de capacidade de armazenamento e processamento **infinita**

Características Essenciais

- Serviço mensurável
 - Baseado no modelo *pay-as-you-go*
 - Identificação de quais recursos foram utilizados
 - Cobrança apenas desses recursos utilizados
- Usuários pagam apenas os serviços que usam e não pagam pelos recursos ociosos

Infraestrutura da Nuvem

- Coleção de *hardware* e *software*
 - Oferece suporte para as cinco características essenciais

software

Camada de Abstração

exemplos

sistema operacional, sistema gerenciador de banco de dados

hardware

Camada Física

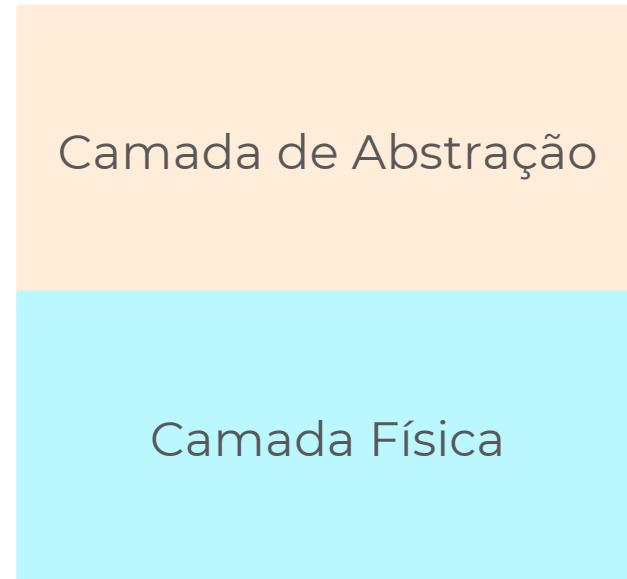
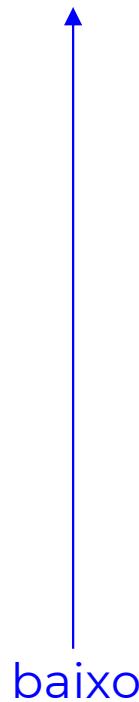
exemplos

rede, servidores, armazenamento

Modelos de Serviços

nível de abstração

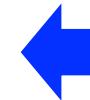
alto



Software as a Service



Platform as a Service



Infrastructure as a Service

Software as a Service (SaaS)

- Recursos fornecidos são os **programas aplicativos**
- Acesso
 - Diferentes dispositivos cliente
 - Uso de uma interface de navegador ou uma interface de programa
- Usuários
 - Podem apenas configurar seus aplicativos específicos

Platform as a Service (PaaS)

- Recurso fornecido é a **plataforma para a execução dos aplicativos**
 - Implantação de aplicativos criados ou adquiridos
- Aplicativos
 - Desenvolvidos usando linguagens de programação, bibliotecas, serviços e ferramentas presentes no provedor
 - Devem ser executados nos recursos da nuvem
- Usuários
 - Têm controle sobre os aplicativos implantados
 - Podem definir configurações para o ambiente de hospedagem de aplicativos

Infrastructure as a Service (IaaS)

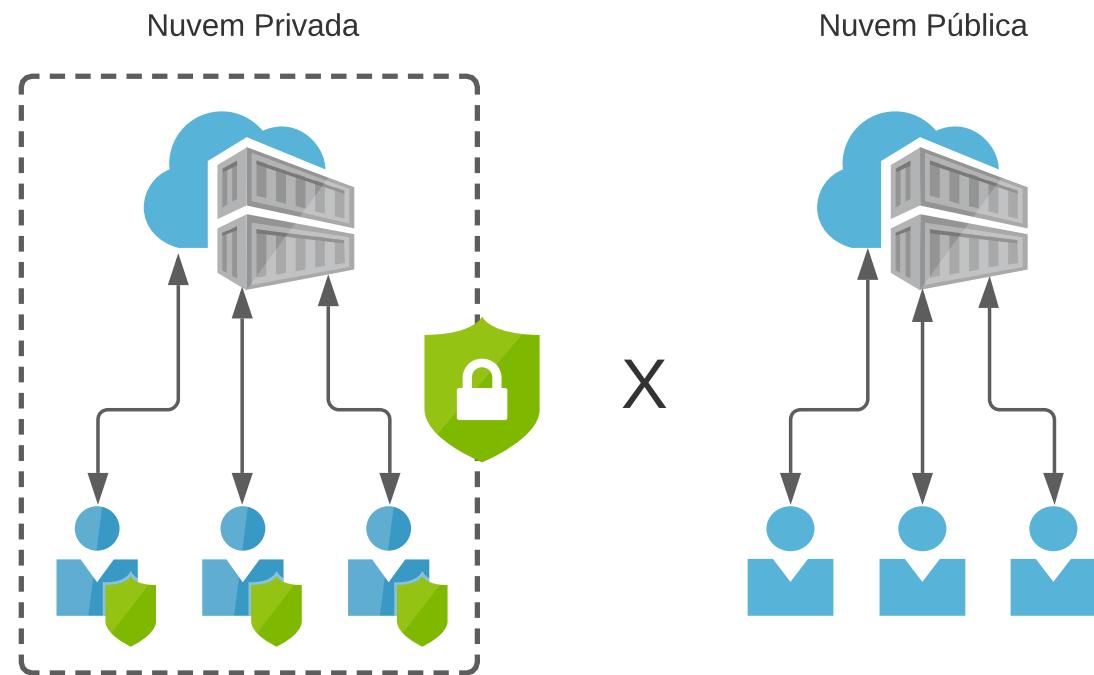
- Recursos fornecidos são relacionados à **infraestrutura**
 - Processamento, armazenamento, redes
 - Outros recursos necessários para implantar e executar qualquer *software*
 - Uso de máquinas virtuais acessadas de forma transparente
- Usuários
 - Têm controle sobre os sistemas operacionais, armazenamento e aplicativos implantados
 - Podem ter controle limitado sobre componentes de rede selecionados

Modelos de Implantação

- Diferenciam-se pelas **restrições de acesso**

- Modelos

- Nuvem privada
- Nuvem comunitária
- Nuvem pública
- Nuvem híbrida



Cluster de Computadores e Computação em Nuvem

	Cluster de Computadores	Computação em Nuvem
Funcionamento dos Componentes	responsabilidade do usuário	responsabilidade do provedor de serviços
Custos de Funcionamento	todos os custos envolvidos, inclusive os referentes aos recursos ociosos	somente os recursos usados (serviço mensurável)

Cluster de Computadores e Computação em Nuvem

	Cluster de Computadores	Computação em Nuvem
Adição ou Remoção de Componentes	responsabilidade do usuário	responsabilidade do provedor de serviços (serviço sob demanda)
Visão do Usuário	transparente (vários computadores conectados para suprir uma necessidade)	ubíqua (serviço criado para atender uma necessidade, sem conhecer detalhes de funcionamento)

Agenda

- Ambientes Computacionais
- Modelo MapReduce
- Apache Spark RDD

MapReduce

- Introduzido pela Google em 2004
- Modelo de programação funcional
- Voltado ao processamento massivo de dados
 - Projetado para *clusters* formados por computadores comuns
 - Foco em processamento paralelo e distribuído
 - Garante alta disponibilidade e tolerância a falhas

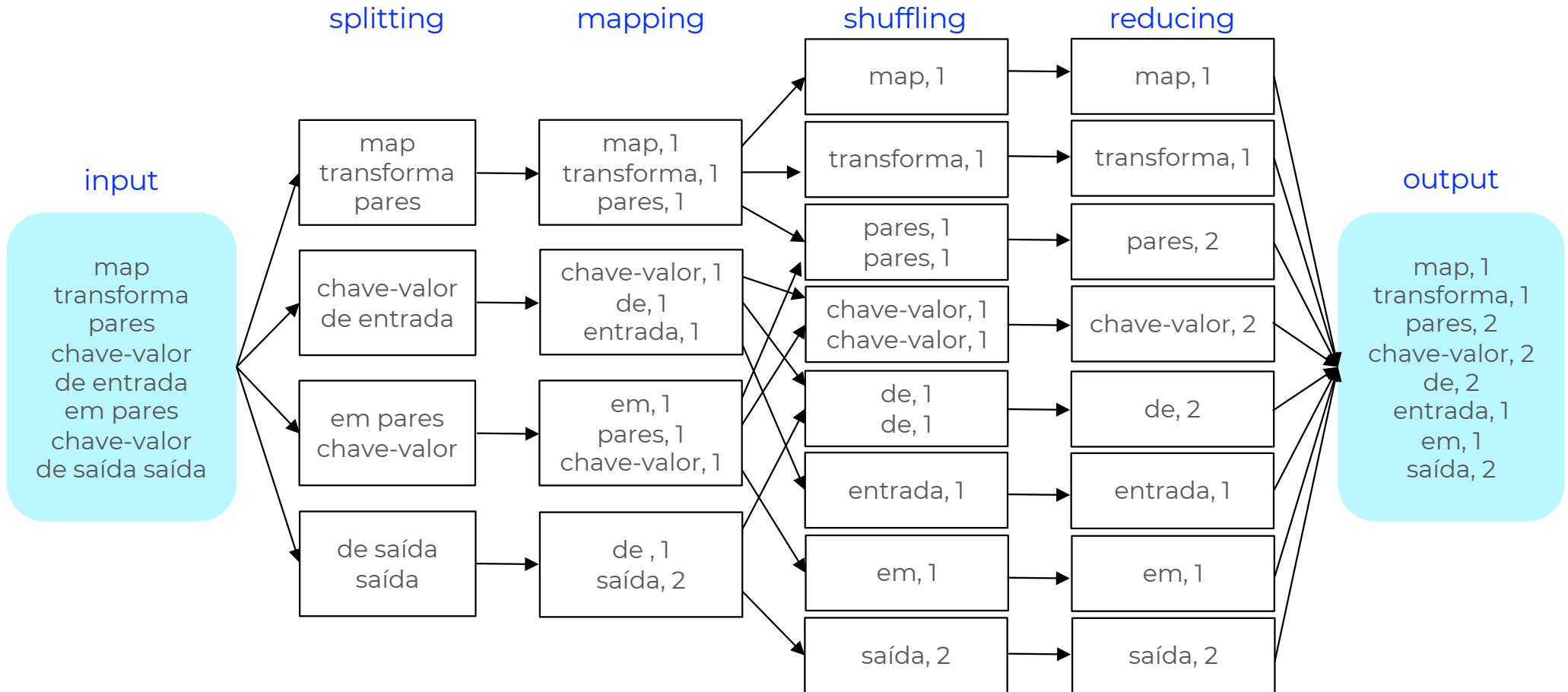
Abstração

- Esconde a complexidade inerente ao paralelismo e à distribuição
 - Armazenamento dos dados
 - Particionamento (distribuição e replicação) dos dados de entrada
 - Balanceamento de carga
 - Escalonamento da execução das tarefas nos nós do *cluster*
 - Manipulação de falhas
 - Gerenciamento da comunicação entre os nós do *cluster*
- Possibilita
 - Manipulação de gigantescos volumes de dados

Funções Base

- Map
 - Processa dados de entrada na forma de **pares chave-valor**
 - Transforma esses dados em saídas intermediárias na forma de **pares chave-valor**
 - Reduce
 - Processa as saídas intermediárias na forma de **pares chave-valor**
 - Agrupa os valores associados a uma mesma chave em um **resultado único**
 - Produz **pares chave-valor**
- Cada job MapReduce executa as funções map e reduce em sequência

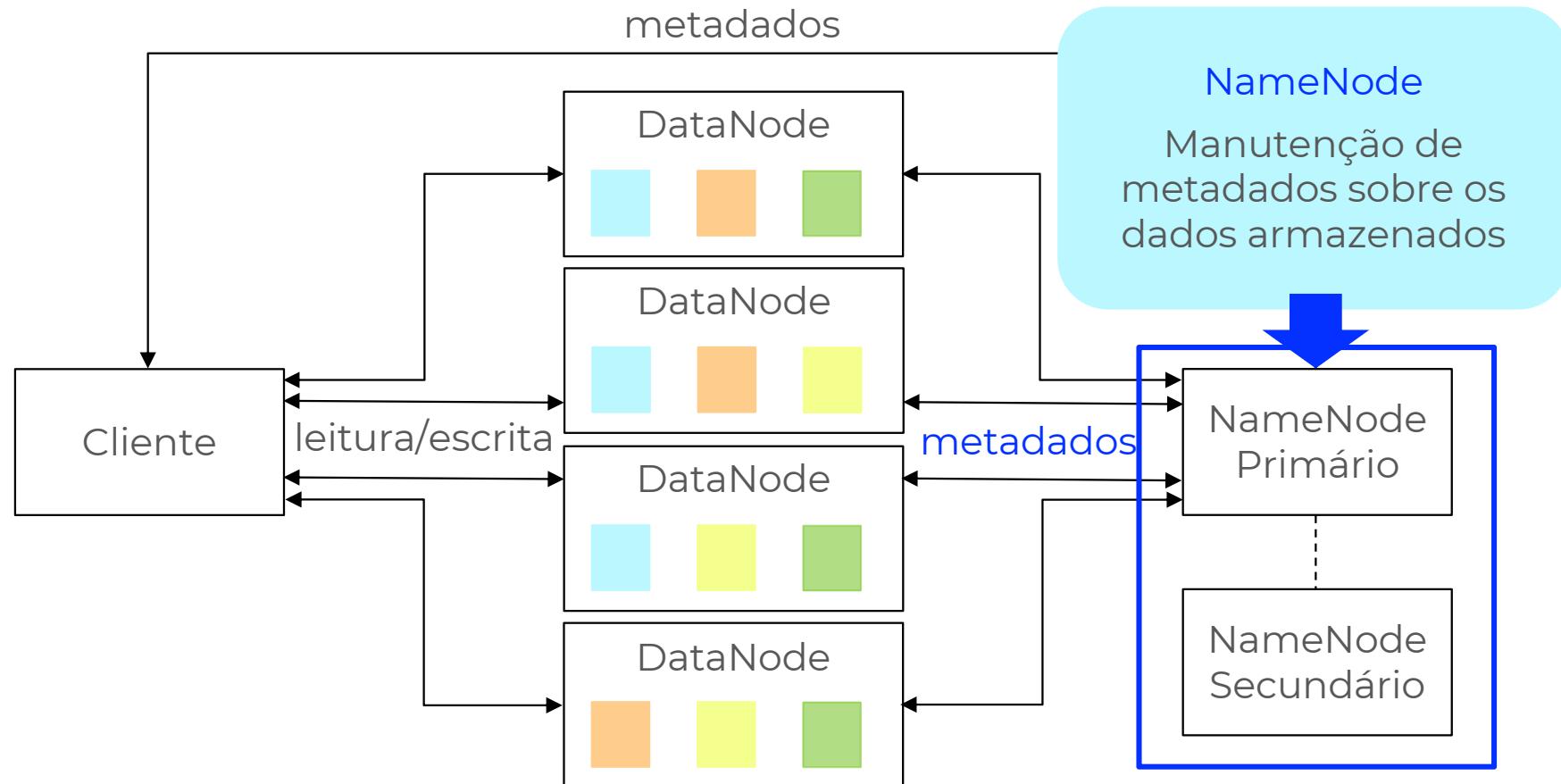
Contador de Palavras



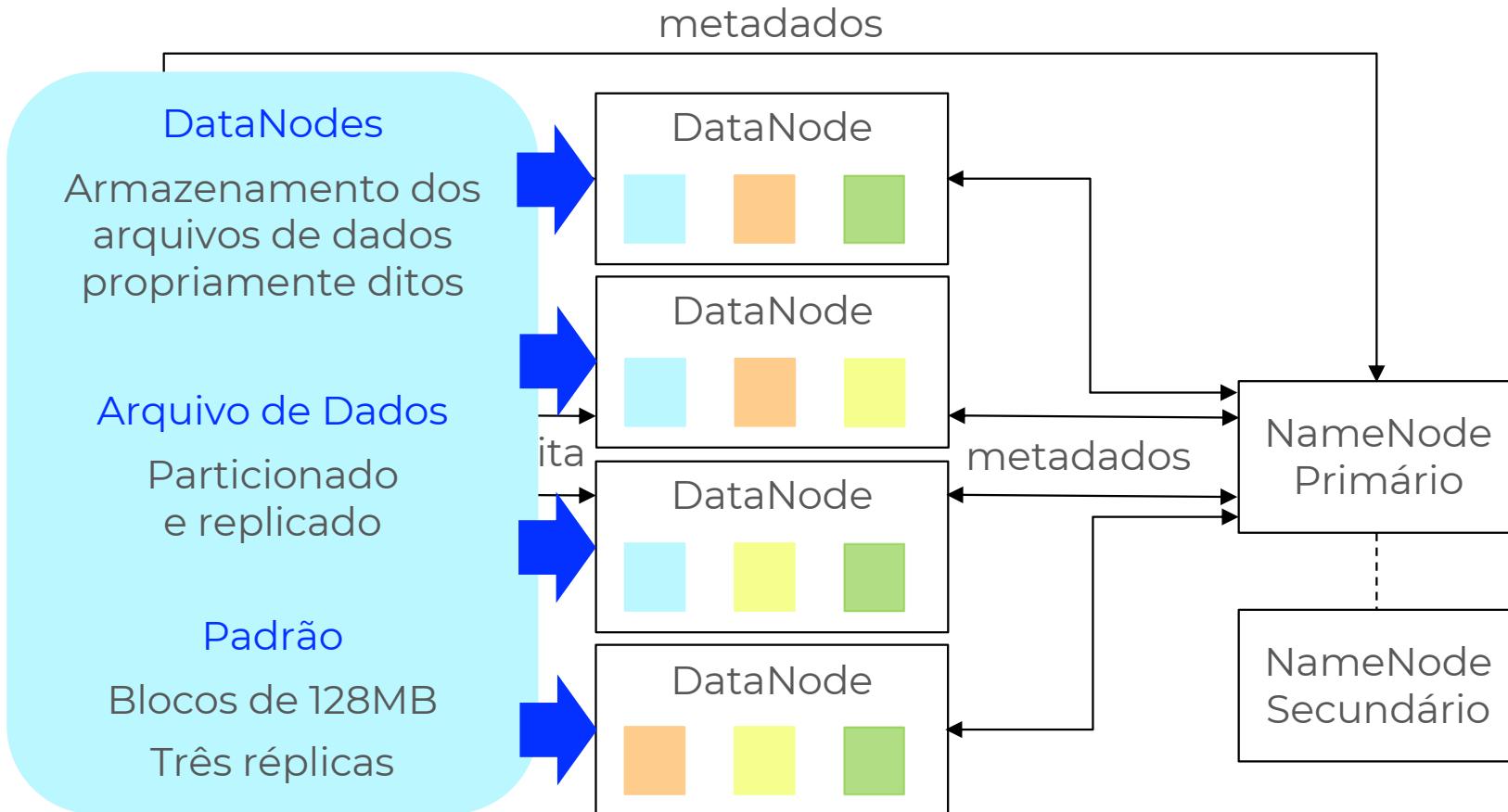
HDFS (Hadoop Distributed File System)

- Sistema de arquivos distribuídos
 - Baseado no Google File System (GoogleFS)
- Características
 - Nó mestre
 - Controle dos outros nós
 - Servidores de dados
 - Armazenamento dos “pedaços” dos arquivos

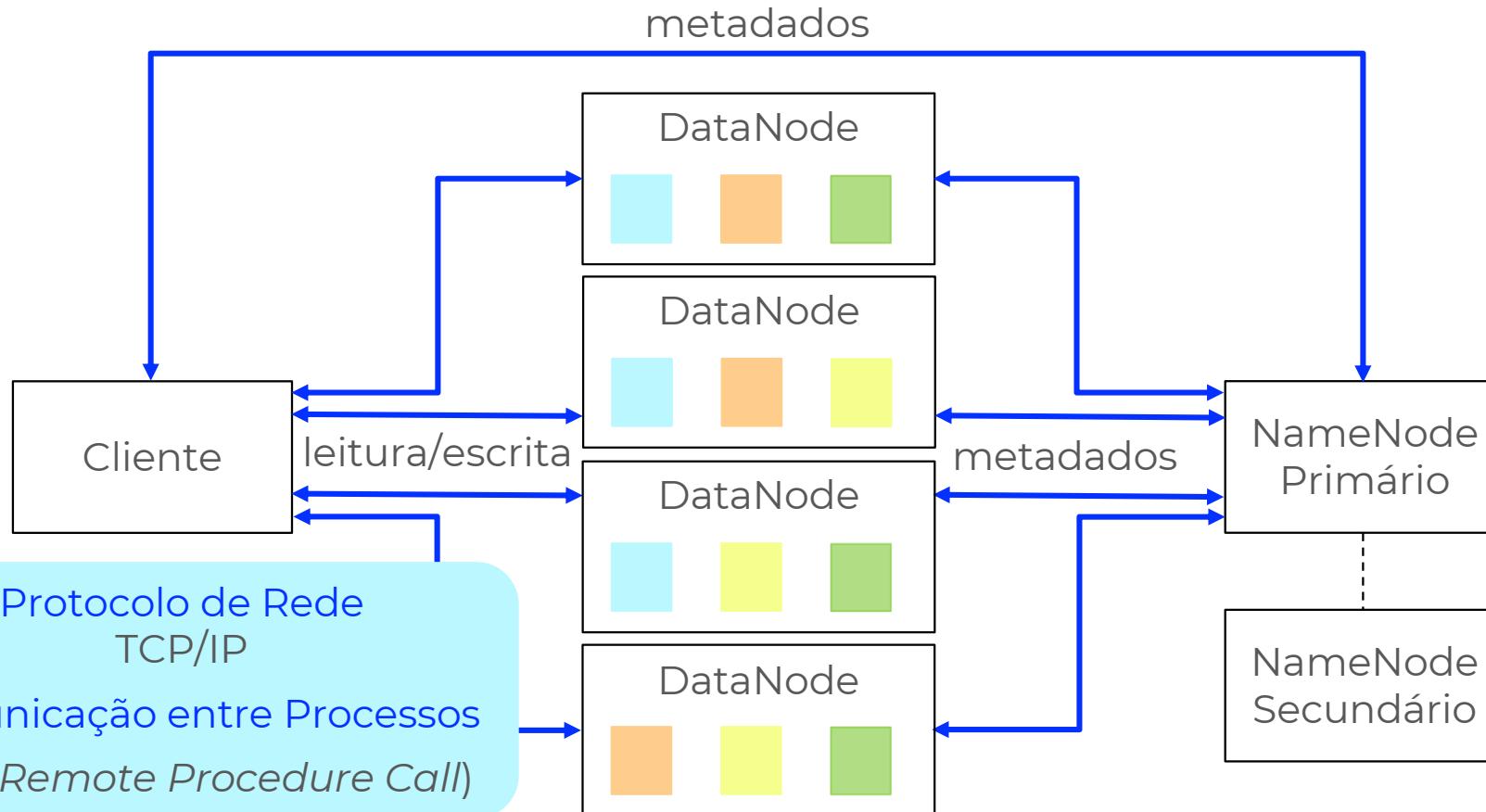
Arquitetura do HDFS



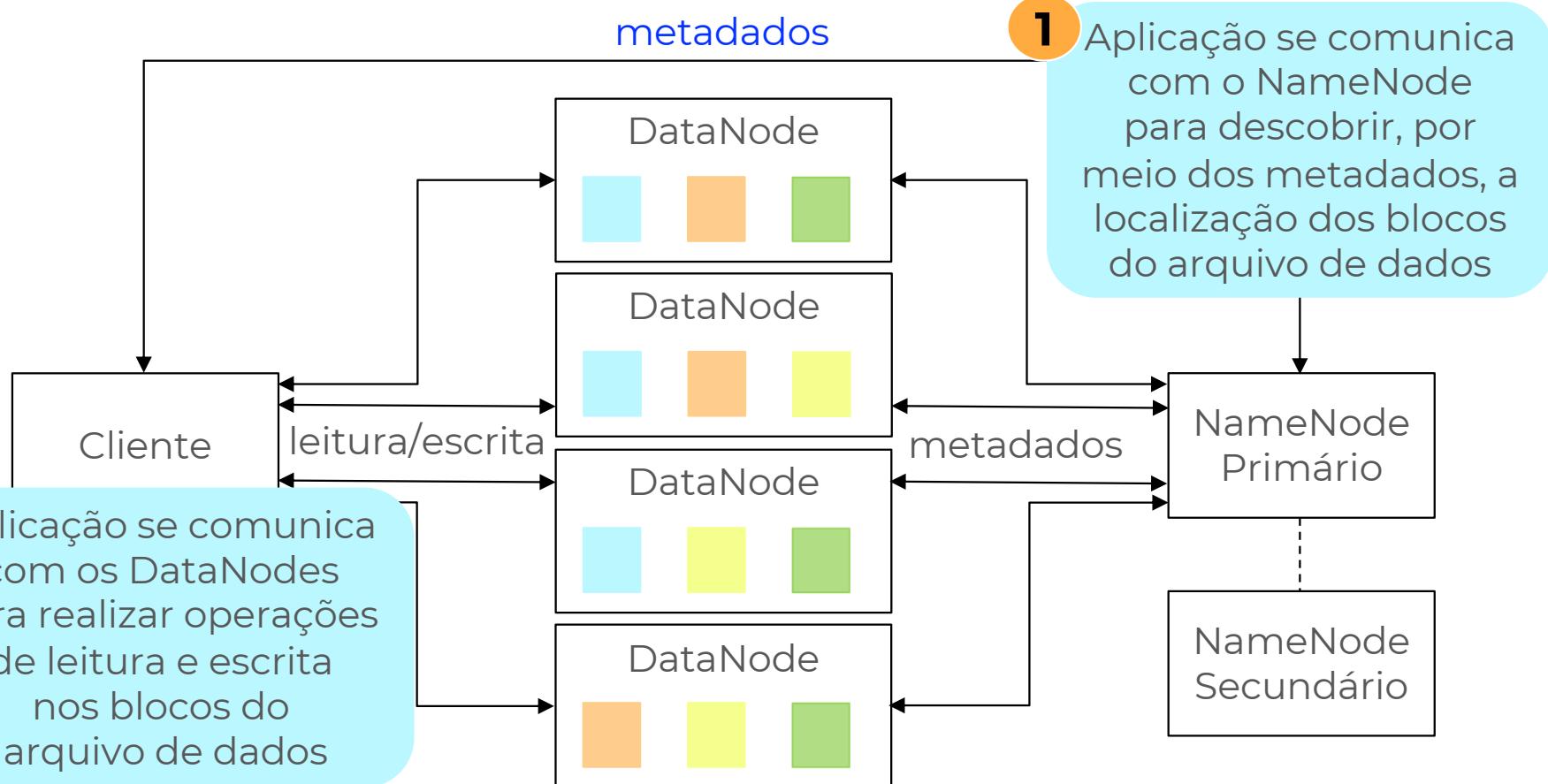
Arquitetura do HDFS



Comunicação



Comunicação



Implementações do MapReduce

- Objetivo
 - Abstrair e facilitar o uso dos conceitos subjacentes
- Frameworks de processamento paralelo e distribuído
 - Hadoop
 - Spark



Apache [Hadoop](#)



Apache [Spark](#)

Apache Hadoop

- Escrito na linguagem de programação **Java**
- Evolução
 - Originalmente projetado para
 - Executar sobre o **HDFS**
 - Incorporar os conceitos relacionados a **MapReduce**
 - Evoluiu para uma **plataforma** que integra um número variado de máquinas de armazenamento e processamento
- Necessidade de incluir **YARN** como um novo componente

Visão Geral da Arquitetura do Hadoop

HDFS

sistema de arquivos
para gerenciar o
armazenamento dos
dados

Yarn

plataforma para
gerenciamento

MapReduce

modelo de
programação

Apache Spark

- Escrito na linguagem de programação [Scala](#)
- Características
 - Executa sobre o [HDFS](#)
 - Incorpora e estende os conceitos relacionados a [MapReduce](#)
 - Baseado no uso de conjuntos de [dados distribuídos e resilientes \(RDDs\)](#)
 - Possibilita o agendamento de tarefas na forma de [grafos acíclicos e direcionados](#)

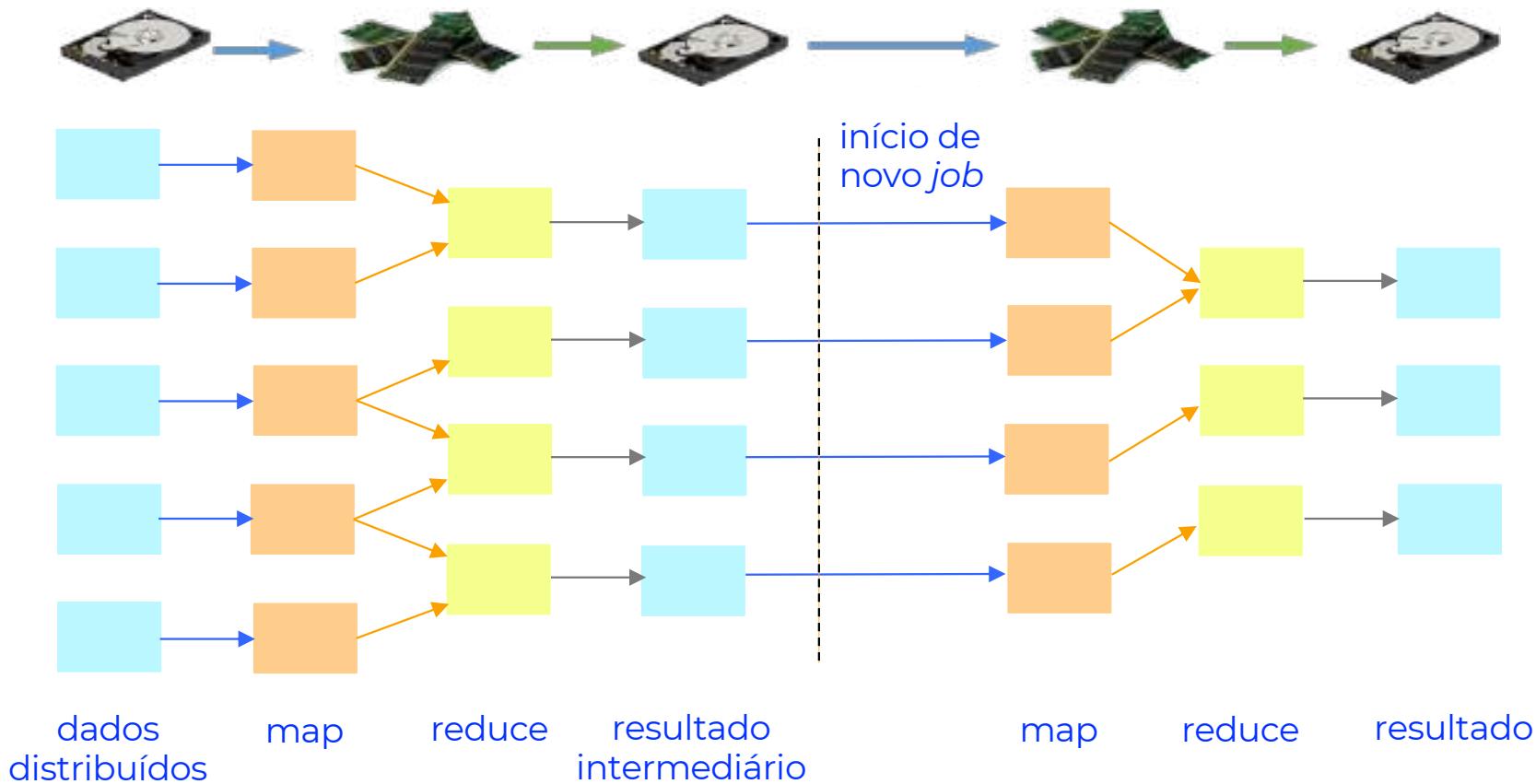
Resilient Distributed Datasets (RDDs)

- Abstrações para a manipulação de dados
 - Toda manipulação de dados deve ser feita sobre os RDDs
- Coleções de blocos de dados distribuídos
 - Capazes de serem reconstruídos em caso de falhas
 - Permitem o armazenamento de resultados intermediários em memória primária
 - Importante quando se deseja reutilizar essas saídas em operações futuras
- Resultado: Uso de RDDs diminui o número de acessos a disco

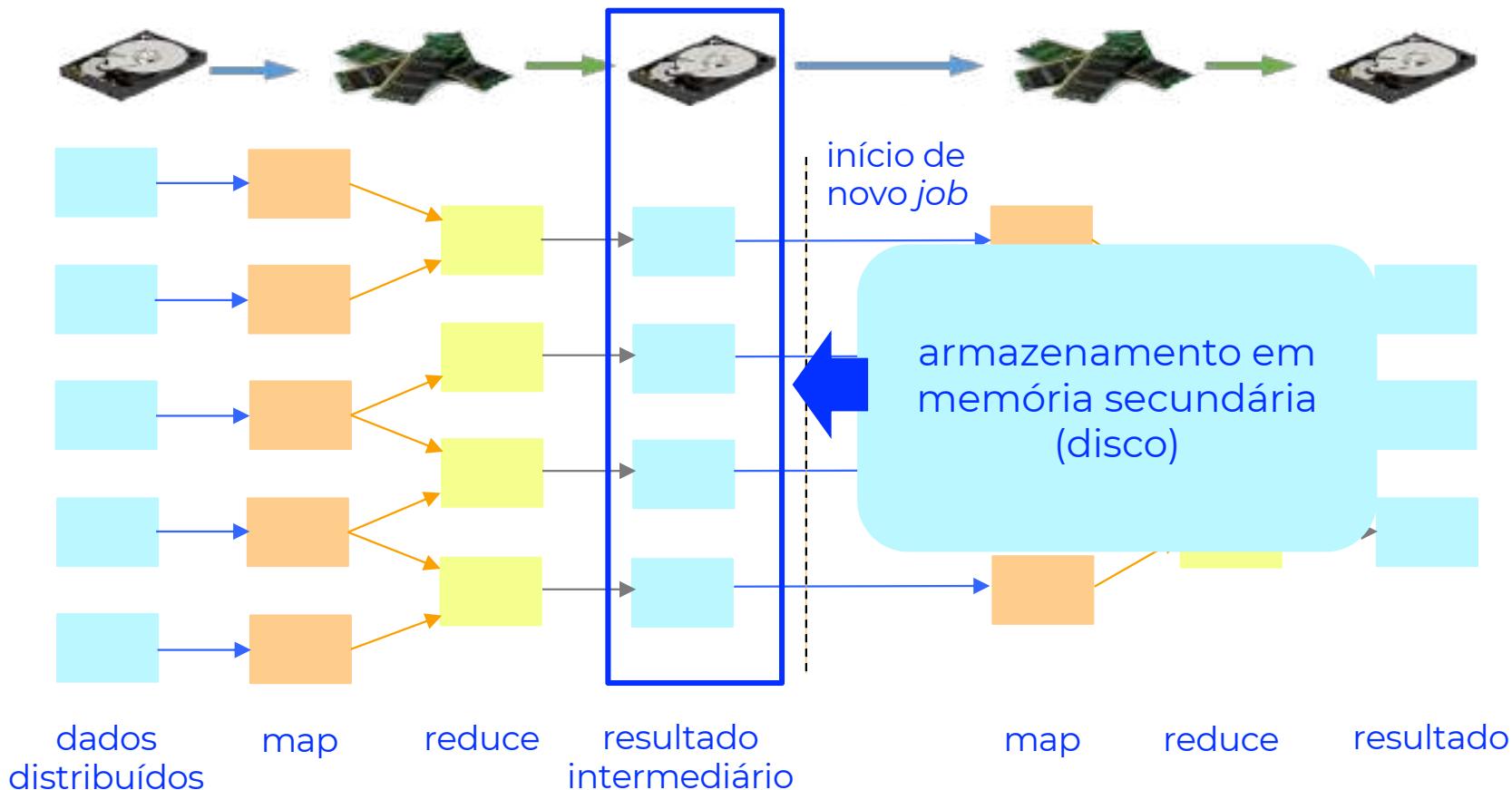
Grafos Acíclicos Direcionados

- Permitem o **agendamento** de estágios
 - Processamento de tarefas consiste de vários estágios
 - Executam os estágios de forma **paralela**
 - Desde que não existam dependências entre os estágios
- Resultado: Uso de grafos acíclicos direcionados melhora o desempenho computacional

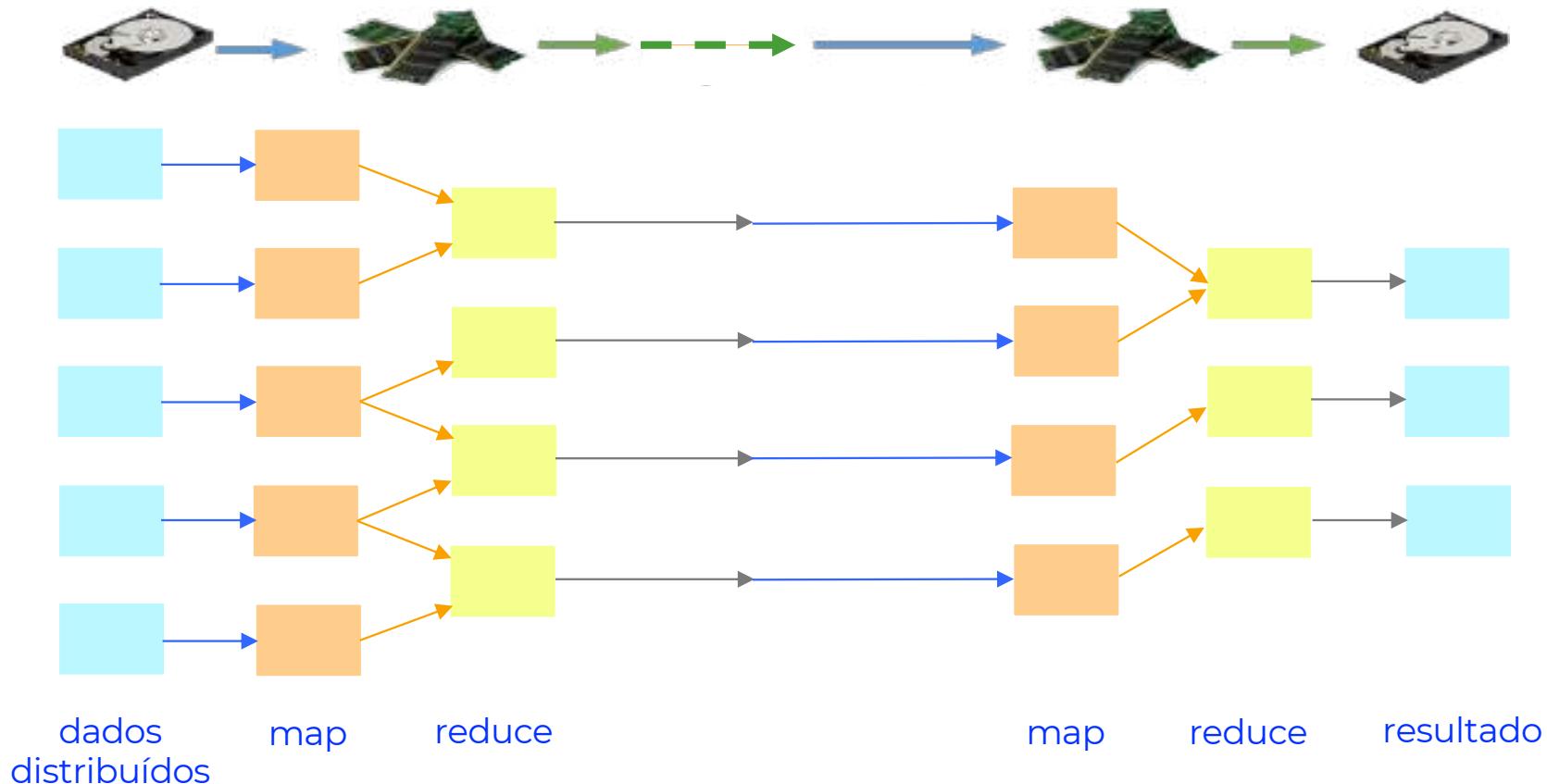
Fluxo de Dados no Hadoop



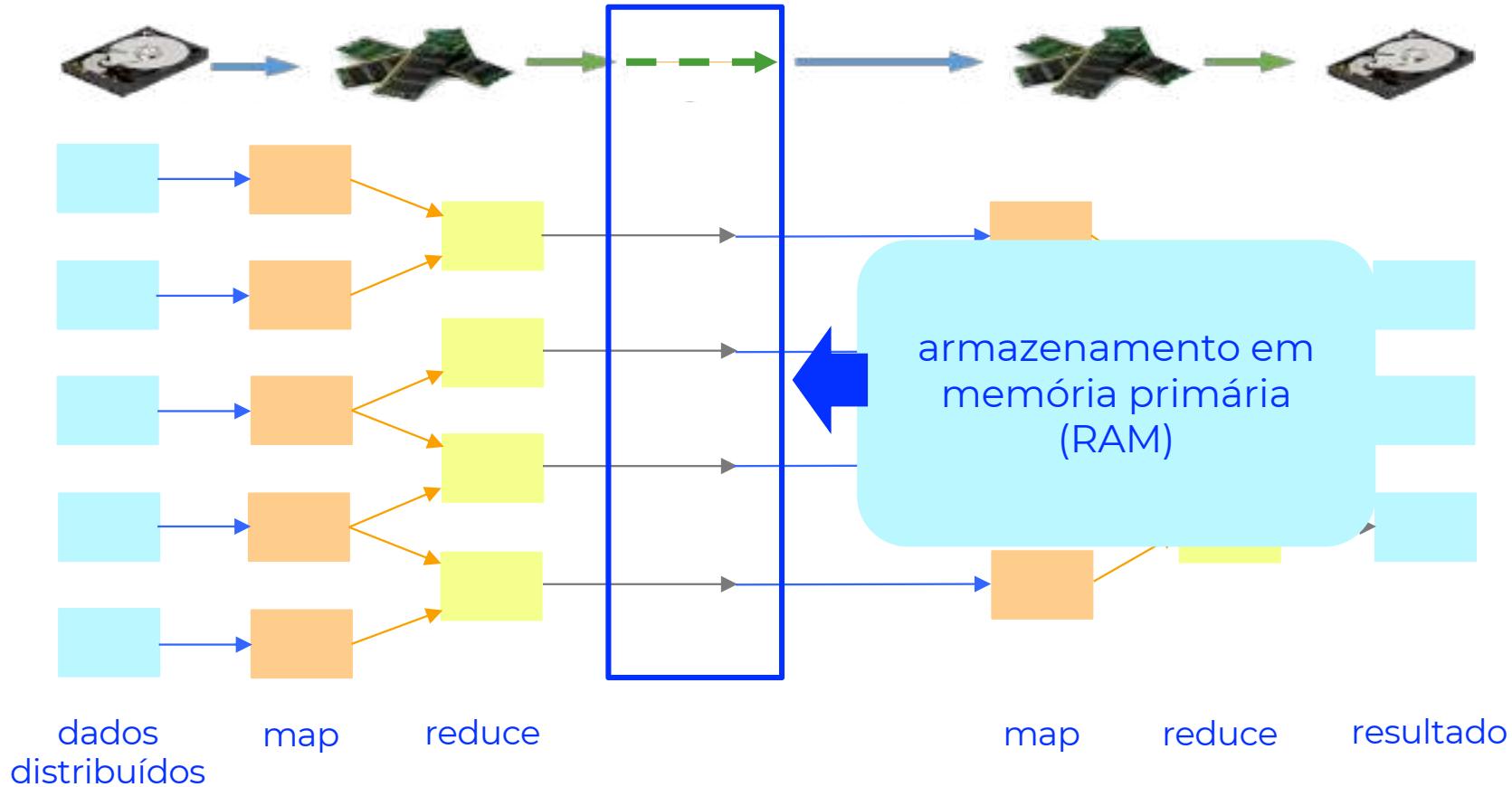
Fluxo de Dados no Hadoop



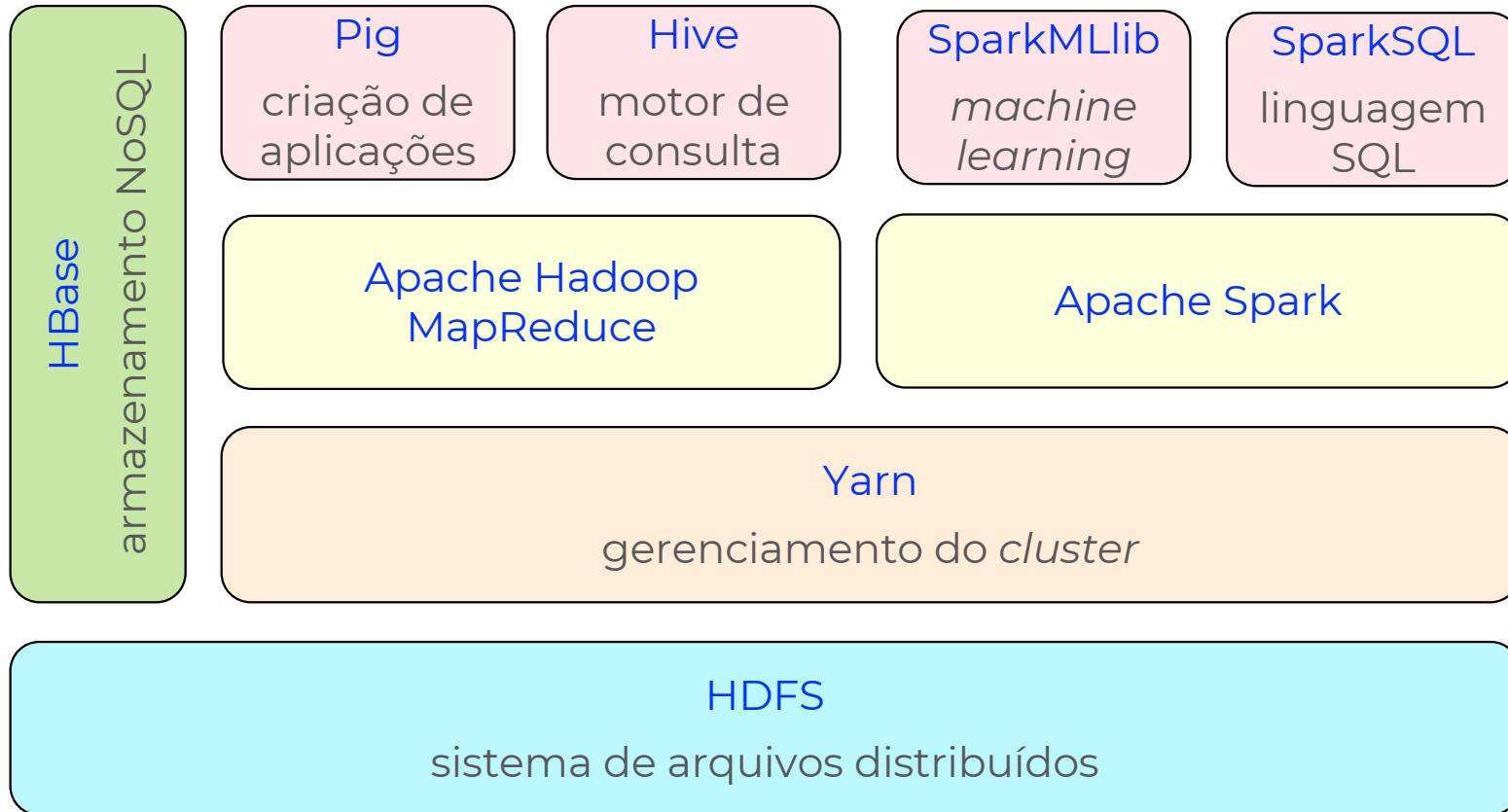
Fluxo de Dados no Spark



Fluxo de Dados no Spark



Ecossistema Hadoop (Tecnologias)



Análise de Dados com Base em Processamento Massivo em Paralelo

Aula 7: Consultas OLAP usando Spark SQL

Cristina Dutra de Aguiar
ICMC/USP
cdac@icmc.usp.br



Agenda

- Linguagem SQL
- Apache Spark SQL

Linguagem SQL

- Estrutura Básica
- Agregação e Agrupamento
- Consultas com Vários Blocos

SQL (Structure Query Language)

- Desenvolvida a partir de 1972
 - Pesquisadores: Donald D. Chamberlin e Raymond F. Joyce
 - Local: laboratório de pesquisa da IBM em San Jose
 - Objetivo: linguagem de consulta para o sistema gerenciador de banco de dados (SGBD) relacional System R
- Evolução contínua desde a sua criação
 - Padronização: American National Standard Institute ([ANSI](#))
International Standard Organization ([ISO](#))
 - Versões: SQL-86, SQL-89, SQL-92, SQL:1999, SQL:2003, SQL:2008, [SQL:2016](#), ...

Características da Linguagem

- Voltada ao **modelo relacional**
- Descreve o problema ao invés da solução
 - Indica **quais** dados devem ser obtidos na resposta da consulta, e não **como** esses dados devem ser obtidos
- Amplamente utilizada
 - **Simplicidade**
 - **Facilidade de ser utilizada**
 - **Grande poder de consulta**

SGBDs Relacionais



IBM Db2

teradata.



MySQL®

IBM
Informix database
software

 Microsoft®
SQL Server®

 MariaDB

Uso de SQL como Tecnologia de Consulta

- NoSQL



Couchbase



- Serviços na Nuvem

Família SQL
do Azure



Apache Presto



Google
BigQuery



AWS Athena

Composição do SQL

- Linguagem de definição de dados (DDL)
- Linguagem de manipulação de dados (DML)
 - Inserção, remoção e atualização dos dados
 - Consulta
- Linguagem para
 - Definição de visões
 - Especificação de restrições de integridade
 - Autorização relacionada aos diretos de acesso para relações e visões
- Linguagem de transação de dados

Linguagem de
Consulta

Comando SELECT

```
SELECT <lista de atributos e funções>
FROM <lista de relações>
[ WHERE predicado de seleção ]
[ GROUP BY <atributos de agrupamento> ]
[ HAVING <condição para agrupamentos> ]
[ ORDER BY <lista de atributos> ]
```

Cláusula SELECT

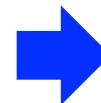
- Especifica **quais dados** são mostrados como resposta
 - Um ou mais **atributos**
 - Uma ou mais **funções**
 - Separados por vírgula
- Atributos
 - Devem estar presentes nas relações especificadas na cláusula FROM

Exemplo de Cláusula SELECT

funcionario

funcPK	funcMatricula	funcNome	funcSexo	funcDataNascimento	funcDiaNascimento	...
1	M-1	ALINE ALMEIDA	F	1/1/1990	1	...
2	M-2	ARAO ALVES	M	2/2/1990	2	...
3	M-3	ARON ANDRADE	M	3/3/1990	3	...
4	M-4	ADA BARBOSA	F	4/4/1990	4	...
5	M-5	ABADE BATISTA	M	5/5/1990	5	...
6	M-6	ABADI BARROS	M	6/6/1990	6	...
...

SELECT funcMatricula, funcNome
FROM funcionario



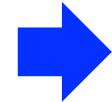
funcMatricula	funcNome
M-1	ALINE ALMEIDA
M-2	ARAO ALVES
M-3	ARON ANDRADE
M-4	ADA BARBOSA
M-5	ABADE BATISTA
M-6	ABADI BARROS
...	...

Exemplo de Cláusula SELECT

funcionario

funcPK	funcMatricula	funcNome	funcSexo	funcDataNascimento	funcDiaNascimento	...
1	M-1	ALINE ALMEIDA	F	1/1/1990	1	...
2	M-2	ARAO ALVES	M	2/2/1990	2	...
3	M-3	ARON ANDRADE	M	3/3/1990	3	...
4	M-4	ADA BARBOSA	F	4/4/1990	4	...
5	M-5	ABADE BATISTA	M	5/5/1990	5	...
6	M-6	ABADI BARROS	M	6/6/1990	6	...
...

SELECT funcSexo
FROM funcionario



funcSexo
F
M
M
F
M
M
...

Cláusula FROM

- Especifica **uma ou mais relações**
 - Contêm os dados solicitados na consulta
 - Devem ser separadas por vírgula
- Realiza um **produto cartesiano** das relações
 - Combina quaisquer tuplas, **independentemente da integridade referencial** existente entre elas
 - Produz tuplas que representam **todas as combinações** de tuplas possíveis entre as relações participantes

mesmos nomes de atributos que aparecem nas duas relações devem ser identificados por nomeRelação.nomeAtributo

Exemplo de Cláusula FROM

funcionario

funcPK	funcMatricula	funcNome	...
1	M-1	ALINE ALMEIDA	...
2	M-2	ARAO ALVES	...
...

pagamento

dataPK	funcPK	equipePK	...
1	1	7	...
1	3	2	...
2	6	7	...
...

SELECT funcionario.funcPK,
funcMatricula,
funcNome,
dataPK,
pagamento.funcPK
FROM funcionario,
pagamento



funcionario .funcPK	funcMatricula	funcNome	dataPK	pagamento .funcPK
1	M-1	ALINE ALMEIDA	1	1
1	M-1	ALINE ALMEIDA	1	3
1	M-1	ALINE ALMEIDA	2	6
2	M-2	ARAO ALVES	1	1
2	M-2	ARAO ALVES	1	3
2	M-2	ARAO ALVES	2	6
...

Cláusula WHERE

- Especifica o **predicado que seleciona** as tuplas
 - Composto por condições
- Condições de seleção
 - Devem ser definidas sobre os atributos das relações na cláusula FROM
 - Incluem condições de junção quando necessário
- Sintaxe de cada condição
 - <atributo> <operador> <valor | atributo | lista de valores | NULL>

Operadores de Comparação

igual a	=
maior que	>
menor que	<
entre dois valores v_1 e v_2	BETWEEN v_1 AND v_2
lista de atributos	IN
valores nulos (NULL)	IS IS NOT

diferente de	<>
maior ou igual a	\geq
menor ou igual a	\leq
cadeia de caracteres	LIKE NOT LIKE
% : substitui qualquer string	
_ : substitui qualquer caractere	

operadores sensíveis ao caso

Predicado de Seleção

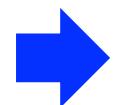
- Sintaxe: <condição₁ > θ <condição₂> θ ... θ <condição_n>
- Operadores lógicos booleanos (θ)
 - conjunção: AND
 - disjunção: OR
 - negação: NOT
- Precedência de operadores
 - NOT; operadores de comparação; AND; OR

Exemplo de Cláusula WHERE

funcionario

funcPK	funcMatricula	funcNome	funcSexo	funcDataNascimento	funcDiaNascimento	...
1	M-1	ALINE ALMEIDA	F	1/1/1990	1	...
2	M-2	ARAO ALVES	M	2/2/1990	2	...
3	M-3	ARON ANDRADE	M	3/3/1990	3	...
4	M-4	ADA BARBOSA	F	4/4/1990	4	...
5	M-5	ABADE BATISTA	M	5/5/1990	5	...
6	M-6	ABADI BARROS	M	6/6/1990	6	...
...

SELECT funcPK, funcMatricula, funcNome
FROM funcionario
WHERE funcPK BETWEEN 2 AND 4



funcPK	funcMatricula	funcNome
2	M-2	ARAO ALVES
3	M-3	ARON ANDRADE
4	M-4	ADA BARBOSA

Cláusula ORDER BY

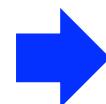
- Ordena as tuplas que aparecem no resultado da consulta
 - `asc` (padrão): ordem ascendente
 - `desc`: ordem descendente
- Ordenação pode ser especificada em `vários atributos`
 - Ordenação referente ao primeiro atributo é prioritária
 - Se houver valores repetidos para o primeiro atributo, então é utilizada a ordenação referente ao segundo atributo
 - E assim por diante

Exemplo de Cláusula ORDER BY

funcionario

funcPK	funcMatricula	funcNome	funcSexo	funcDataNascimento	funcDiaNascimento	...
1	M-1	ALINE ALMEIDA	F	1/1/1990	1	...
2	M-2	ARAO ALVES	M	2/2/1990	2	...
3	M-3	ARON ANDRADE	M	3/3/1990	3	...
4	M-4	ADA BARBOSA	F	4/4/1990	4	...
5	M-5	ABADE BATISTA	M	5/5/1990	5	...
6	M-6	ABADI BARROS	M	6/6/1990	6	...
...

```
SELECT funcPK, funcMatricula, funcNome  
FROM funcionario  
WHERE funcPK BETWEEN 2 AND 4  
ORDER BY funcPK DESC
```



funcPK	funcMatricula	funcNome
4	M-4	ADA BARBOSA
3	M-3	ARON ANDRADE
2	M-2	ARAO ALVES

Cláusula AS

- Renomeia nomes de atributos e relações
 - Sintaxe: `nome_antigo AS nome_novo`
- Atributos
 - Deve aparecer na cláusula `SELECT`
 - Útil para a visualização semântica das respostas
- Relações
 - Deve aparecer na cláusula `FROM`
 - Útil para simplificar os nomes das relações e também quando uma mesma relação é usada mais do que uma vez na consulta

Exemplo de Cláusula AS

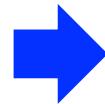
funcionario

funcPK	funcMatricula	funcNome	...
1	M-1	ALINE ALMEIDA	...
2	M-2	ARAO ALVES	...
...

pagamento

dataPK	funcPK	equipePK	...
1	1	7	...
1	3	2	...
2	6	7	...
...

```
SELECT f.funcPK AS `PK de Funcionário`,  
       p.funcPK AS `PK de Pagamento`  
FROM funcionario AS f,  
      pagamento AS p
```



PK de Funcionário	PK de Pagamento
1	1
1	3
1	6
2	1
2	3
2	6
...	...

Resultado da Consulta

- Ordem de apresentação dos atributos
 - Ordem dos atributos na cláusula **SELECT**
- Ordem de apresentação dos dados
 - Ascendente ou descendente de acordo com a cláusula **ORDER BY**
 - Sem ordenação
- Eliminação de valores repetidos
 - Cláusula **SELECT DISTINCT**

Junção (\bowtie)

- Concatena **tuplas relacionadas** de duas relações
 - Com base na **integridade referencial**, ou seja, nos pares (chave estrangeira, chave primária)
- Passos
 - Forma um **produto cartesiano** das relações
 - Faz uma **seleção** forçando a igualdade sobre os atributos que compõem a integridade referencial (e quaisquer outros pares de atributos especificados)



Condição de Junção

- Sintaxe: <condição₁> AND <condição₂> AND ... AND <condição_n>
- Sintaxe de cada condição
 - <atributo da primeira relação> θ <atributo da segunda relação>
 - Theta join: $\theta = \{ =, >, \geq, <, \leq, \neq \}$
- Na prática
 - Equijoin: $\theta = \{ = \}$

Especificando Junção em SQL

- Antes de SQL-92
 - Especificado nas cláusulas **FROM** e **WHERE**
 - **FROM** deve possuir mais do que uma relação
 - **WHERE** deve incluir as condições de junção
- A partir de SQL-92
 - Especificado na cláusula **FROM**
 - Introdução de novas cláusulas **JOIN ... ON <condição de junção>**
 - [INNER] JOIN
 - LEFT [OUTER] JOIN, RIGHT [OUTER] JOIN, FULL [OUTER] JOIN

mesmos nomes de atributos
que aparecem nas duas
relações devem ser
identificados por
nomeRelação.nomeAtributo

Cláusula [INNER] JOIN (R \bowtie S)

- Mantém somente as tuplas de R e S que têm correspondência
- Valores dos atributos das tuplas resultantes
 - Valores obtidos de R e de S

```
SELECT *
FROM funcionario, pagamento
WHERE funcionario.funcPK = pagamento.funcPK
```

```
SELECT *
FROM funcionario JOIN pagamento ON funcionario.funcPK = pagamento.funcPK
```

Exemplo de Cláusula [INNER] JOIN

funcionario

funcPK	funcMatricula	funcNome	...
1	M-1	ALINE ALMEIDA	...
2	M-2	ARAO ALVES	...
...

pagamento

dataPK	funcPK	equipePK	...
1	1	7	...
1	3	2	...
2	6	7	...
...

```
SELECT funcionario.funcPK, funcMatricula, funcNome, dataPK, pagamento.funcPK, equipePK  
FROM funcionario JOIN pagamento ON funcionario.funcPK = pagamento.funcPK
```



funcionario.funcPK	funcMatricula	funcNome	dataPK	pagamento.funcPK	equipePK
1	M-1	ALINE ALMEIDA	1	1	7
...

Cláusula LEFT [OUTER] JOIN ($R \bowtie S$)

- Mantém todas as **tuplas** de **R**
- Valores dos atributos das tuplas resultantes
 - Valores obtidos de **R** e de **S** quando existe correspondência
 - Valores obtidos de **R** e valores **nulo** quando não existe correspondência

Exemplo de Cláusula LEFT [OUTER] JOIN

funcionario

funcPK	funcMatricula	funcNome	...
1	M-1	ALINE ALMEIDA	...
2	M-2	ARAO ALVES	...
...

pagamento

dataPK	funcPK	equipePK	...
1	1	7	...
1	3	2	...
2	6	7	...
...

```
SELECT funcionario.funcPK, funcMatricula, funcNome, dataPK, pagamento.funcPK, equipePK  
FROM funcionario LEFT JOIN pagamento ON funcionario.funcPK = pagamento.funcPK
```



funcionario.funcPK	funcMatricula	funcNome	dataPK	pagamento.funcPK	equipePK
1	M-1	ALINE ALMEIDA	1	1	7
2	M-2	ARAO ALVES	null	null	null
...

Cláusula RIGHT [OUTER] JOIN ($R \bowtie S$)

- Mantém todas as **tuplas de S**
- Valores dos atributos das tuplas resultantes
 - Valores obtidos de **R** e de **S** quando existe correspondência
 - Valores obtidos de **S** e valores **nulo** quando não existe correspondência

Exemplo de Cláusula RIGHT [OUTER] JOIN

funcionario

funcPK	funcMatricula	funcNome	...
1	M-1	ALINE ALMEIDA	...
2	M-2	ARAO ALVES	...
...

pagamento

dataPK	funcPK	equipePK	...
1	1	7	...
1	3	2	...
2	6	7	...
...

```
SELECT funcionario.funcPK, funcMatricula, funcNome, dataPK, pagamento.funcPK, equipePK  
FROM funcionario RIGHT JOIN pagamento ON funcionario.funcPK = pagamento.funcPK
```



funcionario.funcPK	funcMatricula	funcNome	dataPK	pagamento.funcPK	equipePK
1	M-1	ALINE ALMEIDA	1	1	7
null	null	null	1	3	2
null	null	null	2	6	7
...

Cláusula FULL [OUTER] JOIN (R \bowtie S)

- Mantém todas as **tuplas** de R e S mesmo sem correspondência
- Valores dos atributos das tuplas resultantes
 - Valores obtidos de **R** e de **S** quando existe correspondência
 - Valores obtidos de **R** e valores **nulo** quando não existe correspondência
 - Valores obtidos de **S** e valores **nulo** quando não existe correspondência

Exemplo de Cláusula FULL [OUTER] JOIN

funcionario

funcPK	funcMatricula	funcNome	...
1	M-1	ALINE ALMEIDA	...
2	M-2	ARAO ALVES	...
...

pagamento

dataPK	funcPK	equipePK	...
1	1	7	...
1	3	2	...
2	6	7	...
...

```
SELECT funcionario.funcPK, funcMatricula, funcNome, dataPK, pagamento.funcPK, equipePK  
FROM funcionario FULL JOIN pagamento ON funcionario.funcPK = pagamento.funcPK
```



funcionario.funcPK	funcMatricula	funcNome	dataPK	pagamento.funcPK	equipePK
1	M-1	ALINE ALMEIDA	1	1	7
2	M-2	ARAO ALVES	null	null	null
null	null	null	1	3	2
null	null	null	2	6	7
...

Processamento Lógico da Consulta

- 3 SELECT <lista de atributos e funções>
- 1 FROM <lista de relações>
- 2 WHERE predicado de seleção
- 4 ORDER BY <lista de atributos>

Processamento Lógico da Consulta

- 3 SELECT <lista de atributos e funções>
 DISTINCT
- 1 FROM <lista de relações>
 ON
 JOIN
- 2 WHERE predicado de seleção
- 4 ORDER BY <lista de atributos>

Linguagem SQL

Copyright © 2020. Todos os direitos reservados
ao CeMEAI-USP. Proibida a cópia e reprodução
sem autorização



- Estrutura Básica
- Agregação e Agrupamento
- Consultas com Vários Blocos

Funções de Agregação

- Característica
 - Recebem uma coleção de valores como entrada
 - Retornam um único valor como saída
- Funções e resultado retornado
 - **SUM()**: soma
 - **MIN()**: menor
 - **MAX()**: maior
 - **AVG()**: média
 - **COUNT()**: quantidade de tuplas

função de agregação (**DISTINCT ...**)
realiza a função de
agregação considerando apenas
valores distintos

Exemplo de Funções de Agregação

pagamento

dataPK	funcPK	equipePK	cargoPK	salario	quantidadeLancamento
1	1	7	112	2.226,66	1
1	3	2	74	9.169,90	1
2	6	7	112	2.226,66	1
7	1	3	23	3.828,90	1
...

SELECT

```
SUM(salario) AS soma,  
MIN(salario) AS menor,  
MAX(salario) AS maior,  
AVG(salario) AS media,  
COUNT(salario) AS quantidade,  
COUNT(DISTINCT salario) AS diferente  
FROM pagamento
```



soma	menor	maior	media	quantidade	diferente
17.452,12	2.226,66	9.169,90	4.363,03	4	3

Cláusula GROUP BY

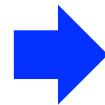
- Aplica uma função de agregação a um **grupo de conjunto de tuplas**
 - Para cada grupo de conjunto de tuplas
 - Retorna um único valor
- Atributos de agrupamento
 - Usados na cláusula **GROUP BY**
 - Têm que ser especificados na cláusula **SELECT**

Exemplo de Cláusula GROUP BY

pagamento

dataPK	funcPK	equipePK	cargoPK	salario	quantidadeLancamento
1	1	7	112	2.226,66	1
1	3	2	74	9.169,90	1
2	6	7	112	2.226,66	1
7	1	3	23	3.828,90	1
...

```
SELECT cargoPK,  
       SUM(salario) AS soma  
FROM pagamento  
GROUP BY cargoPK  
ORDER BY cargoPK
```



cargoPK	soma
23	3.828,90
74	9.169,90
112	4.453,32

Cláusula HAVING

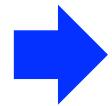
- Especifica **condições de seleção** para as **funções de agregação**
 - Aplicada sobre os grupos de conjunto de tuplas
- Recupera os **valores dos grupos** de conjunto de tuplas
 - Desde que esses valores satisfaçam às condições de seleção
- Usada conjuntamente com a cláusula **GROUP BY**
 - Quando são definidos atributos de agrupamento

Exemplo de Cláusula HAVING

pagamento

dataPK	funcPK	equipePK	cargoPK	salario	quantidadeLancamento
1	1	7	112	2.226,66	1
1	3	2	74	9.169,90	1
2	6	7	112	2.226,66	1
7	1	3	23	3.828,90	1
...

```
SELECT cargoPK,  
       SUM(salario) AS soma  
FROM pagamento  
GROUP BY cargoPK  
HAVING SUM(salario) > 5000  
ORDER BY cargoPK
```



cargoPK	soma
74	9.169,90

Estendendo a Cláusula GROUP BY

- Objetivo
 - Gerar diferentes níveis de agregação dos dados
- Utilidade
 - Aplicações de *data warehousing*
- Extensões
 - ROLLUP
 - CUBE
 - GROUPING SETS

Extensão ROLLUP

- Cria subtotais
 - A partir do nível mais detalhado até o nível menos detalhado
 - Para [combinações dos atributos da lista de agrupamento de acordo com a ordem desses atributos](#)
- Processamento
 - Recebe como argumento uma lista dos [n](#) atributos usados na criação dos subtotais
 - Cria, de forma progressiva, subtotais de nível mais alto, considerando os [n](#) atributos da esquerda para a direita
 - Produz como resultado [n + 1](#) níveis de agregação

Exemplo de Extensão ROLLUP

pagamento

dataPK	funcPK	equipePK	cargoPK	salario	quantidadeLancamento
1	1	7	112	2.226,66	1
1	3	2	74	9.169,90	1
2	1	7	112	2.226,66	1
...

```
SELECT dataPK, funcPK, equipePK, cargoPK,  
       SUM(salario) AS `gastos em salário` ,  
       SUM(quantidadeLancamento) AS `lançamentos`  
FROM pagamento  
GROUP BY ROLLUP (dataPK, funcPK, equipePK, cargoPK)
```

Exemplo de Extensão ROLLUP

GROUP BY **ROLLUP** (dataPK, funcPK, equipePK, cargoPK)

dataPK	funcPK	equipePK	cargoPK	gastos em salário	lançamentos
1	1	7	112	2.226,66	1
1	3	2	74	9.169,90	1
2	1	7	112	2.226,66	1
1	1	7	null	2.226,66	1
1	3	2	null	9.169,90	1
2	1	7	null	2.226,66	1
1	1	null	null	2.226,66	1
1	3	null	null	9.169,90	1
2	1	null	null	2.226,66	1
1	null	null	null	11.396,56	2
2	null	null	null	2.226,66	1
null	null	null	null	13.623,22	3

Exemplo de Extensão ROLLUP

GROUP BY **ROLLUP** (dataPK, funcPK, equipePK, cargoPK)

dataPK	funcPK	equipePK	cargoPK	gastos em salário	lançamentos
1	1	7	112	2.226,66	1
1	3	2	74	9.169,90	1
2	1	7	112	2.226,66	1
1	1	7	"	2.226,66	1
1	3	2	"	9.169,90	1
2	1	7	"	2.226,66	1
1	1	null	"	2.226,66	1
1	3	null	"	9.169,90	1
2	1	null	"	2.226,66	1
1	null	null	"	11.396,56	2
2	null	null	"	2.226,66	1
null	null	null	"	13.623,22	3

$$\begin{aligned} n + 1 \text{ níveis} = \\ 4 + 1 = 5 \end{aligned}$$

Extensão CUBE

- Cria subtotais
 - A partir do nível mais detalhado até o nível menos detalhado
 - Para [todas as combinações dos atributos da lista de agrupamento](#)
- Processamento
 - Recebe como argumento uma lista dos [n](#) atributos usados na criação dos subtotais
 - Cria, de forma progressiva, subtotais de nível mais alto, considerando todas as combinações dos [n](#) atributos
 - Produz como resultado [2ⁿ](#)

Exemplo de Extensão CUBE

pagamento

dataPK	funcPK	equipePK	cargoPK	salario	quantidadeLancamento
1	1	7	112	2.226,66	1
1	3	2	74	9.169,90	1
2	1	7	112	2.226,66	1
...

```
SELECT dataPK, funcPK, equipePK, cargoPK,  
       SUM(salario) AS `gastos em salário`,  
       SUM(quantidadeLancamento) AS `lançamentos`  
FROM pagamento  
GROUP BY CUBE (dataPK, funcPK, equipePK, cargoPK)
```

Exemplo de Extensão CUBE

GROUP BY **CUBE** (dataPK, funcPK, equipePK, cargoPK)

dataPK	funcPK	equipePK	cargoPK	gastos em salário	lançamentos
1	1	7	112	2.226,66	1
1	3	2	74	9.169,90	1
2	1	7	112	2.226,66	1
1	1	7	null	2.226,66	1
1	3	2	null	9.169,90	1
2	1	7	null	2.226,66	1
1	1	null	112	2.226,66	1
1	3	null	74	9.169,90	1
2	1	null	112	2.226,66	1
1	null	7	112	2.226,66	1
1	null	2	74	9.169,90	1
2	null	7	112	2.226,66	1
null	1	7	112	4.453,32	2
null	3	2	74	9.169,90	1
1	1	null	null	2.226,66	1
1	3	null	null	9.169,90	1
2	1	null	null	2.226,66	1
null	1	7	null	4.453,32	2
null	3	2	null	9.169,90	1
1	1	null	null	2.226,66	1
1	3	null	null	9.169,90	1
2	1	null	null	2.226,66	1
null	1	7	null	4.453,32	2
null	3	2	null	9.169,90	1

dataPK	funcPK	equipePK	cargoPK	gastos em salário	lançamentos
1	null	7	null	2.226,66	1
1	null	2	null	9.169,90	1
2	null	7	null	2.226,66	1
1	null	null	112	2.226,66	1
1	null	null	74	9.169,90	1
2	null	null	112	2.226,66	1
null	1	null	112	4.453,32	2
null	3	null	74	9.169,90	1
null	null	7	112	4.453,32	2
null	null	2	74	9.169,90	1
null	null	null	112	4.453,32	2
null	null	null	74	9.169,90	1
null	null	7	null	4.453,32	2
null	null	2	null	9.169,90	1
null	1	null	null	4.453,32	2
null	3	null	null	9.169,90	1
1	null	null	null	11.396,56	2
2	null	null	null	2.226,66	1
null	null	null	null	13.623,22	3

Exemplo de Extensão CUBE

GROUP BY **CUBE** (dataPK, funcPK, equipePK, cargoPK)

dataPK	funcPK	equipePK	cargoPK	gastos em salário	lançamentos
1	1	7	112	2.226,66	1
1	3	2	74	9.169,90	1
2	1	7	112	2.226,66	1
1	1	7	null	2.226,66	1
1	3	2	null	9.169,90	1
2	1	7	null	2.226,66	1
1	1	null	112	2.226,66	
1	3	null	74	9.169,90	
2	1	null	112	2.226,66	
1	null	7	112	2.226,66	
1	null	2	74	9.169,90	
2	null	7	112	2.226,66	
null	1	7	112	4.453,32	2
null	3	2	74	9.169,90	1
1	1	null	null	2.226,66	1
1	3	null	null	9.169,90	1
2	1	null	null	2.226,66	1
null	1	7	null	4.453,32	2
null	3	2	null	9.169,90	1

2ⁿ níveis
= 2⁴ = 16

dataPK	funcPK	equipePK	cargoPK	gastos em salário	lançamentos
1	null	7	null	2.226,66	1
1	null	2	null	9.169,90	1
2	null	7	null	2.226,66	1
1	1	null	112	2.226,66	1
1	1	null	74	9.169,90	1
2	1	null	112	2.226,66	1
1	1	7	112	4.453,32	2
1	1	7	74	9.169,90	1
2	1	7	112	4.453,32	2
1	1	7	74	9.169,90	1
1	3	2	112	4.453,32	2
1	3	2	74	9.169,90	1
2	1	7	112	4.453,32	2
1	1	7	112	4.453,32	2
1	1	7	74	9.169,90	1
2	1	7	74	9.169,90	1
1	1	7	112	11.396,56	2
1	3	2	112	2.226,66	1
2	1	7	112	13.623,22	3

Extensão GROUPING SETS

- Cria subtotais
 - Para quaisquer combinações de atributos desejados
- Subtotais criados
 - Definidos em uma [lista que especifica cada nível de agregação desejado](#)
 - Podem ser equivalentes às extensões ROLLUP e CUBE
 - Podem ser referentes a outros subtotais

Exemplo de Extensão GROUPING SETS

```
SELECT dataPK, funcPK, equipePK, cargoPK,  
       SUM(salario) AS `gastos em salário`,  
       SUM(quantidadeLancamento) AS `lançamentos`  
FROM pagamento  
GROUP BY ROLLUP (dataPK, funcPK, equipePK, cargoPK)
```

```
SELECT dataPK, funcPK, equipePK, cargoPK,  
       SUM(salario) AS `gastos em salário`,  
       SUM(quantidadeLancamento) AS `lançamentos`  
FROM pagamento  
GROUP BY GROUPING SETS ((dataPK, funcPK, equipePK, cargoPK),  
                           (dataPK, funcPK, equipePK), (dataPK, funcPK), (dataPK), ())
```

Exemplo de Extensão GROUPING SETS

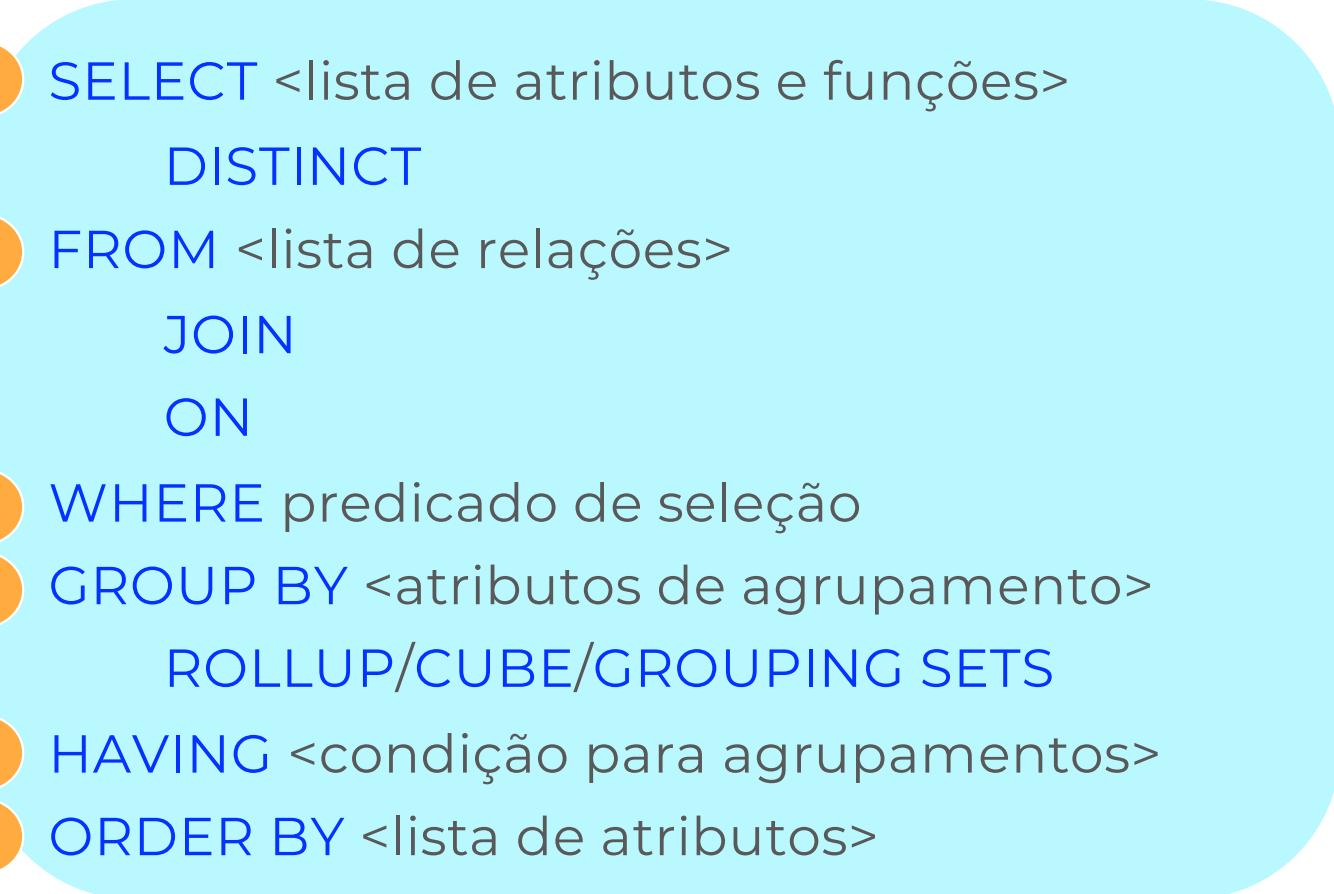
```
SELECT dataPK, funcPK, equipePK, cargoPK,  
       SUM(salario) AS `gastos em salário`,  
       SUM(quantidadeLancamento) AS `lançamentos`  
FROM pagamento  
GROUP BY CUBE (dataPK, funcPK, equipePK, cargoPK)
```

```
SELECT dataPK, funcPK, equipePK, cargoPK,  
       SUM(salario) AS `gastos em salário`,  
       SUM(quantidadeLancamento) AS `lançamentos`  
FROM pagamento  
GROUP BY GROUPING SETS ((dataPK, funcPK, equipePK, cargoPK), (dataPK, funcPK, equipePK),  
                           (dataPK, funcPK, cargoPK), (dataPK, equipePK, cargoPK), (funcPK, equipePK, cargoPK),  
                           (dataPK, funcPK), (dataPK, equipePK), (dataPK, cargoPK), (funcPK, equipePK),  
                           (funcPK, cargoPK), (equipePK, cargoPK), (dataPK), (funcPK), (equipePK), (cargoPK), ())
```

Processamento Lógico da Consulta

- 5 SELECT <lista de atributos e funções>
- 1 FROM <lista de relações>
- 2 WHERE predicado de seleção
- 3 GROUP BY <atributos de agrupamento>
- 4 HAVING <condição para agrupamentos>
- 6 ORDER BY <lista de atributos>

Processamento Lógico da Consulta

- 
- 5 SELECT <lista de atributos e funções>
DISTINCT
 - 1 FROM <lista de relações>
JOIN
ON
 - 2 WHERE predicado de seleção
 - 3 GROUP BY <atributos de agrupamento>
ROLLUP/CUBE/GROUPING SETS
 - 4 HAVING <condição para agrupamentos>
 - 6 ORDER BY <lista de atributos>

Linguagem SQL

- Estrutura Básica
- Agregação e Agrupamento
- Consultas com Vários Blocos

Bloco de Consulta

- Unidade básica
 - Contém uma **única expressão SELECT-FROM-WHERE**
- Tipos de consulta com vários blocos
 - Operações sobre conjuntos
 - Subconsultas aninhadas
 - Consultas complexas

Operações sobre Conjuntos

- Operações

- União
- Intersecção
- Diferença

Duas relações R e S são compatíveis se:

- possuem o mesmo número n de atributos
- os domínios do i -ésimo atributo de R e do i -ésimo atributo de S são os mesmos ($1 \leq i \leq n$)

- Características

- Atuam sobre relações compatíveis
- Eliminam as tuplas repetidas do resultado

Descrição das Operações

- União entre R e S
 - Resultado contém todas as tuplas pertencentes a R, a S, ou a ambas R e S
 - Operação **UNION**
- Intersecção entre R e S
 - Resultado contém todas as tuplas pertencentes a ambas R e S
 - Operação **INTERSECT**
- Diferença entre R e S
 - Resultado contém todas as tuplas pertencentes a R que não pertencem a S
 - Operação **MINUS/EXCEPT**

Exemplo da Operação UNION

funcionario (funcPK BETWEEN 1 AND 2)

funcPK	funcMatricula	funcNome	...
1	M-1	ALINE ALMEIDA	...
2	M-2	ARAO ALVES	...

pagamento (funcPK IN (1, 3, 6))

dataPK	funcPK	equipePK	...
1	1	7	...
1	3	2	...
2	6	7	...

```
SELECT funcPK  
FROM funcionario  
UNION  
SELECT funcPK  
FROM pagamento
```



funcPK
1
2
3
6

Exemplo da Operação INTERSECT

funcionario (funcPK BETWEEN 1 AND 2)

funcPK	funcMatricula	funcNome	...
1	M-1	ALINE ALMEIDA	...
2	M-2	ARAO ALVES	...

pagamento (funcPK IN (1, 3, 6))

dataPK	funcPK	equipePK	...
1	1	7	...
1	3	2	...
2	6	7	...

```
SELECT funcPK  
FROM funcionario  
INTERSECT  
SELECT funcPK  
FROM pagamento
```



funcPK
1

Exemplo da Operação MINUS

funcionario (funcPK BETWEEN 1 AND 2)

funcPK	funcMatricula	funcNome	...
1	M-1	ALINE ALMEIDA	...
2	M-2	ARAO ALVES	...

pagamento (funcPK IN (1, 3, 6))

dataPK	funcPK	equipePK	...
1	1	7	...
1	3	2	...
2	6	7	...

```
SELECT funcPK  
FROM funcionario  
MINUS  
SELECT funcPK  
FROM pagamento
```



funcPK
2

Subconsultas Aninhadas

- Subconsulta
 - Bloco de consulta **aninhado** dentro de outra consulta
 - Aplicações mais comuns
 - Testes para membros de conjuntos (**IN** e **NOT IN**)
 - Cardinalidade de conjuntos (**EXISTS** e **NOT EXISTS**)
- **Conjunto: coleção de valores produzidos pela subconsulta**

Membros de Conjuntos

- Conectivo **IN**
 - Testa se um ou mais atributos **são membros do conjunto**
 - **WHERE** (atributo₁, ... atributo_n) **IN**
(SELECT atributo₁, ..., atributo_n
FROM)
- Conectivo **NOT IN**
 - Testa se um atributo ou mais atributos **não são membros do conjunto**
 - **WHERE** (atributo₁, ... atributo_n) **NOT IN**
(SELECT atributo₁, ..., atributo_n
FROM)

Exemplo do Conectivo IN

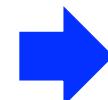
funcionario

funcPK	funcMatricula	funcNome	...
1	M-1	ALINE ALMEIDA	...
2	M-2	ARAO ALVES	...
...

pagamento

dataPK	funcPK	equipePK	...
1	1	7	...
1	3	2	...
2	6	7	...
...

```
SELECT funcPK, funcMatricula, funcNome  
FROM funcionario  
WHERE funcPK IN  
( SELECT funcPK  
    FROM pagamento )
```



funcPK	funcMatricula	funcNome
1	M-1	ALINE ALMEIDA
...

Cardinalidade de Conjuntos

- Construção **EXISTS**
 - A condição é verdadeira quando o conjunto retornado não for vazio
 - WHERE **EXISTS**
(SELECT ...
FROM ...)
- Construção **NOT EXISTS**
 - A condição é verdadeira quando o conjunto retornado for vazio
 - WHERE **NOT EXISTS**
(SELECT ...
FROM ...)

Exemplo da Construção EXISTS

funcionario

funcPK	funcMatricula	funcNome	...
1	M-1	ALINE ALMEIDA	...
2	M-2	ARAO ALVES	...
...

pagamento

dataPK	funcPK	equipePK	...
1	1	7	...
1	3	2	...
2	6	7	...
...

```
SELECT funcPK, funcMatricula, funcNome  
FROM funcionario  
WHERE EXISTS  
( SELECT *  
    FROM pagamento  
    WHERE funcionario.funcPK = pagamento.funcPK  
)
```



funcPK	funcMatricula	funcNome
1	M-1	ALINE ALMEIDA
...

Consultas Complexas

- Consultas difíceis ou impossíveis de se escrever
 - Usando apenas um único bloco de consulta
 - Usando união, intersecção ou diferença entre blocos de consulta
- Relação derivada
 - Subconsulta especificada na cláusula **FROM**
 - Deve possuir um **nome** diferente dos nomes das relações base
 - Pode possuir uma lista de **atributos renomeados**

Exemplo de Consulta Complexa

pagamento

dataPK	funcPK	equipePK	cargoPK	salario	...
1	1	7	112	2.226,66	...
1	3	2	74	9.169,90	...
2	6	7	112	2.226,66	...
7	1	3	23	3.828,90	...

data

dataPK	...	dataAno
1	...	2016
2	...	2016
3	...	2017
7	...	2020

negociacao

dataPK	equipePK	clientePK	receita	...
1	7	2	5.245,00	...
3	3	3	5.431,23	...
7	1	3	5.789,00	...
1	3	4	9.323,00	...

```
SELECT anoGasto, gasto, ganho
FROM
(
    SELECT dataAno, SUM(salario)
    FROM data, pagamento
    WHERE data.dataPK = pagamento.dataPK
    GROUP BY dataAno
) AS pag(anoGasto, gasto),
(
    SELECT dataAno, SUM(receita)
    FROM data, negociacao
    WHERE data.dataPK = negociacao.dataPK
    GROUP BY dataAno
) AS neg(anoGanho, ganho)
WHERE anoGasto = anoGanho
ORDER BY anoGasto
```

Exemplo de Consulta Complexa

pagamento

dataPK	funcPK	equipePK	cargoPK	salario	...
1	1	7	112	2.226,66	...
1	3	2	74	9.169,90	...
2	6	7	112	2.226,66	...
7	1	3	23	3.828,90	...

data

dataPK	...	dataAno
1	...	2016
2	...	2016
3	...	2017
7	...	2020

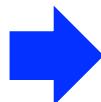
negociacao

dataPK	equipePK	clientePK	receita	...
1	7	2	5.245,00	...
3	3	3	5.431,23	...
7	1	3	5.789,00	...
1	3	4	9.323,00	...

```
SELECT anoGasto, gasto, ganho
FROM
( SELECT dataAno, SUM(salario)
  FROM data, pagamento
 WHERE data.dataPK = pagamento.dataPK
 GROUP BY dataAno
) AS pag(anoGasto, gasto),
(
  SELECT dataAno, SUM(receita)
  FROM data, negociacao
 WHERE data.dataPK = negociacao.dataPK
 GROUP BY dataAno
) AS neg(anoGanho, ganho)
WHERE anoGasto = anoGanho
ORDER BY anoGasto
```

Exemplo de Consulta Complexa

```
SELECT anoGasto, gasto, ganho
FROM
(  SELECT dataAno, SUM(salario)
   FROM data, pagamento
  WHERE data.dataPK = pagamento.dataPK
  GROUP BY dataAno
) AS pag(anoGasto, gasto),
(
  SELECT dataAno, SUM(receita)
   FROM data, negociacao
  WHERE data.dataPK = negociacao.dataPK
  GROUP BY dataAno
) AS neg(anoGanho, ganho)
WHERE anoGasto = anoGanho
ORDER BY anoGasto
```

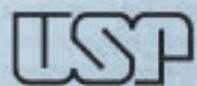


anoGasto	gasto	ganho
2016	13.623,22	14.568,00
2017	0	5.431,23
2020	3.828,90	5.789,00

Análise de Dados com Base em Processamento Massivo em Paralelo

Aula 8: Explorando o Módulo pyspark.sql

Cristina Dutra de Aguiar
ICMC/USP
cdac@icmc.usp.br



Conteúdo das Aulas

	Foco	Detalhamento
Aula 05	Pandas	diversos métodos consultas sobre DataFrames
Aula 06	Apache Spark RDD	diversos métodos manipulação de RDDs
Aula 07	Módulo pyspark.sql	método spark.sql() consultas textuais em SQL
Aula 08	Módulo pyspark.sql	diversos métodos consultas sobre DataFrames

Infraestrutura Computacional

	Processamento	Ambiente
Aula 05 Pandas	centralizado	um computador
Aula 06 Spark RDD		<i>cluster</i> de computadores
Aula 07 spark.sql()	paralelo e distribuído	ou
Aula 08 pyspark.sql		computação em nuvem

Processamento dos Dados

	Persistência em Disco	Computação em RAM
Aula 05 Pandas	sistema de arquivos local	centralizada
Aula 06 Spark RDD		
Aula 07 spark.sql()	sistema de arquivos distribuído (HDFS)	distribuída
Aula 08 pyspark.sql		

Complexidade de Instalação

	Módulos	Outras Configurações
Aula 05 Pandas	Pandas com suporte de Python	-
Aula 06 Spark RDD	Java	criação e configuração de sessão
Aula 07 spark.sql()	Spark findspark	configuração de variáveis de ambiente
Aula 08 pyspark.sql	pyspark	

Volume de Dados

	Volume	Escalabilidade Atual
Aula 05 Pandas	grandes volumes	até 5GB
Aula 06 Spark RDD	gigantescos volumes	até 1TB
Aula 07 spark.sql()		
Aula 08 pyspark.sql		

Velocidade de Dados

	Latência	Módulo
Aula 05 Pandas	lote	-
Aula 06 Spark RDD	lote e streaming	Apache Spark Streaming
Aula 07 spark.sql()	lote	Apache Spark SQL
Aula 08 pyspark.sql		

Variedade de Dados

	Tipo de Dados	Conceito Subjacente
Aula 05 Pandas	estruturados	DataFrames
Aula 06 Spark RDD	estruturados, semiestruturados e não estruturados	RDDs
Aula 07 spark.sql()	estruturados	DataFrames construídos sobre RDDs
Aula 08 pyspark.sql		

Abstração do Conceito de RDD

	Nível	Detalhamento
Aula 05 Pandas	-	-
Aula 06 Spark RDD	baixo	RDDs manipulados diretamente
Aula 07 spark.sql()	alto	uso de comandos SQL
Aula 08 pyspark.sql	alto	uso de métodos funcionais

Abstração Problema e Solução

	Programação	Detalhamento
Aula 05 Pandas	funcional	como os dados devem ser obtidos
Aula 06 Spark RDD		
Aula 07 spark.sql()	declarativa	quais dados devem ser obtidos
Aula 08 pyspark.sql	funcional	como os dados devem ser obtidos

Grau de Conhecimento do Usuário

Experiente em	
Aula 05 Pandas	resolução de consultas passo a passo
Aula 06 Spark RDD	uso de métodos de baixo nível programação paralela e distribuída
Aula 07 spark.sql()	uso da linguagem SQL
Aula 08 pyspark.sql	resolução de consultas passo a passo programação paralela e distribuída

Agenda

- Métodos de Interesse
- Consultas OLAP
- Comparativo Pandas, Spark RDD e Spark SQL

AS1

Questão 1

Correto

Atingiu 2,00 de 2,00

Texto da questão

As siglas DW, OLAP e ETL correspondem, respectivamente, a:

- a.
Data warehousing, online accurate processing, extract-transform-load.
- b.
Nenhuma das alternativas anteriores.
- c.
Data warehousing, online analytical processing, extract-transform-load.
- d.
Data warehouse, online analytical processing, extract-transform-load.
- e.
Data warehouse, online accurate processing, extract-transform-load.

Feedback

Sua resposta está correta.

A resposta correta é:

Data warehouse, online analytical processing, extract-transform-load.

Questão 2

Correto

Atingiu 2,00 de 2,00

Texto da questão

Marque a alternativa correta:

- a.
Uma prática bastante comum utilizada no ambiente informacional é a normalização.
- b.
No ambiente informacional, os tipos de operação mais frequentes são de inserção, remoção e atualização dos dados.
- c.
Quando se pensa no ambiente operacional, pensa-se na modelagem multidimensional dos dados e no armazenamento dos dados em diferentes níveis de detalhe.
- d.
O ambiente operacional é caracterizado majoritariamente por aplicações que realizam a análise de dados voltada à tomada de decisão estratégica.
- e.
No ambiente informacional, o foco de desempenho é a produtividade das consultas.

Feedback

Sua resposta está correta.

A resposta correta é:

No ambiente informacional, o foco de desempenho é a produtividade das consultas.

Questão 3

Correto

Atingiu 2,00 de 2,00

Texto da questão

Marque a alternativa correta:

a.

O uso exploratório das informações com possibilidade de identificar tendências e realizar previsões é um exemplo de necessidade relacionada a business intelligence.

b.

Um dos objetivos de business intelligence é produzir a informação certa para a pessoa certa. Porém, não é necessário que esta informação seja entregue na hora certa.

c.

Informação advém da interpretação, da análise e do processamento do conhecimento gerado. Ela possui o valor mais agregado e pode ser usada para orientar as ações das empresas, possibilitando a tomada de decisão estratégica.

d.

Business intelligence consiste no processo de transformação dos dados brutos primeiro em conhecimento e depois em informação.

e.

Quando se pensa em business intelligence e em data warehousing, pensa-se em transações (ou seja, inserções, atualizações e remoções) ao invés de geração de conhecimento.

Feedback

Sua resposta está correta.

A resposta correta é:

O uso exploratório das informações com possibilidade de identificar tendências e realizar previsões é um exemplo de necessidade relacionada a business intelligence.

Questão 4

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere as seguintes afirmações:

I - Data warehouse se refere ao local onde os dados estão fisicamente armazenados, enquanto data warehousing é um conceito mais abrangente, que se refere ao ambiente como um todo.

II - O data warehouse armazena dados integrados e históricos.

III - Exemplos de usuários do ambiente operacional incluem administradores do sistema, projetistas e usuários finais, enquanto exemplos de usuários do ambiente informacional incluem analistas, gerentes e executivos.

Assinale a alternativa que contém as afirmações corretas.

- a.
I, II, e III.
- b.
I e III.
- c.
II e III.
- d.
Somente II.
- e.
I e II.

Feedback

Sua resposta está correta.

A resposta correta é:

I, II, e III.

Questão 5

Correto

Atingiu 2,00 de 2,00

Texto da questão

Marque a alternativa que é composta por exemplos comuns de aplicações em um ambiente informacional:

- a.
Planejamento estratégico, análise de tendências e empréstimos de livros.
- b.
Transações bancárias, planejamento de marketing e análise de tendências.
- c.
Transações bancárias, empréstimos de livros, matrículas em cursos.
- d.
Análises históricas, planejamento estratégico e matrículas em cursos.
- e.
Planejamento de marketing, tomada de decisão e análise financeira.

Feedback

Sua resposta está correta.

A resposta correta é:

Planejamento de marketing, tomada de decisão e análise financeira.

[Terminar revisão](#)

AS2

Questão 1

Incorreto

Atingiu 0,00 de 2,00

Texto da questão

Considerando as camadas da arquitetura de data warehousing, assinale a alternativa **incorrecta**:

a.

O data lake é o componente da camada de data warehouse que armazena dados não estruturados, semiestruturados e estruturados.

b.

Os bancos de dados operacionais fazem parte da camada de fonte de dados.

c.

O data mart é o componente da camada de data warehouse que consiste em um pequeno data warehouse com escopo limitado quando comparado ao data warehouse propriamente dito.

d.

O repositório de metadados é o componente da camada de data warehouse que é responsável por armazenar os metadados de todos os dados e processos envolvidos no ambiente de data warehousing.

e.

As ferramentas OLAP são exemplos de ferramentas que fazem parte da camada de ferramentas de análise e consulta.

Feedback

Sua resposta está incorreta.

A resposta correta é:

O data lake é o componente da camada de data warehouse que armazena dados não estruturados, semiestruturados e estruturados.

Questão 2

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considerando os conceitos relativos às ferramentas de análise e consulta, assinale a alternativa **incorrecta**:

a.

Geradores de relatório são os tipos mais simples de ferramentas e têm como objetivo principal produzir relatórios periódicos.

b.

Modelos de machine learning são ferramentas que reúnem diversos dados e indicadores por meio de gráficos e tabelas, permitindo o monitoramento simultâneo de um grande número de informações.

c.

Ferramentas de análise estatística permitem que usuários de sistemas de suporte à decisão analisem os dados usando métodos estatísticos.

d.

As ferramentas de análise e consulta são a forma pela qual os usuários de sistemas de suporte à decisão interagem com o ambiente de data warehousing.

e.

Ferramentas OLAP (on-line analytical processing) são caracterizadas por permitir que usuários de sistemas de suporte à decisão analisem os dados usando visões multidimensionais complexas e elaboradas.

Feedback

Sua resposta está correta.

A resposta correta é:

Modelos de machine learning são ferramentas que reúnem diversos dados e indicadores por meio de gráficos e tabelas, permitindo o monitoramento simultâneo de um grande número de informações.

Questão 3

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considerando os conceitos de data warehouse e data mart, assinale a alternativa **incorreta**:

a.

Construir um data warehouse pode ser mais barato do que construir um data mart, uma vez que o data warehouse armazena dados referentes a um escopo limitado do projeto.

b.

Os dados de um data mart são caracterizados por serem organizados multidimensionalmente.

c.

É possível construir vários data marts primeiro e, depois, construir um data warehouse corporativo completo.

d.

A criação de data marts independentes pode conduzir a problemas de integração, gerando inconsistências entre os dados.

e.

Os dados armazenados em data marts compartilham as mesmas características dos dados armazenados no data warehouse.

Feedback

Sua resposta está correta.

A resposta correta é:

Construir um data warehouse pode ser mais barato do que construir um data mart, uma vez que o data warehouse armazena dados referentes a um escopo limitado do projeto.

Questão 4

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considerando os conceitos de data staging area, data lake e data warehouse, assinale a alternativa **incorreta**:

a.

O processamento de consultas analíticas em um data lake é usualmente mais custoso do que o processamento de consultas analíticas em um data staging area.

b.

Para povoar o data warehouse, os dados precisam primeiro passar pelo processo de ELT. Já os dados armazenados no data lake são decorrentes do processo de ETL.

c.

O data lake contém um grande volume de dados extraídos das fontes de dados em seu formato nativo (raw data), incluindo dados estruturados, semiestruturados e não estruturados.

d.

O data lake pode atuar também como data staging area.

e.

O data staging area contém os dados das fontes de dados que vão passando por sucessivas modificações até que estejam prontos para serem carregados no data warehouse.

Feedback

Sua resposta está correta.

A resposta correta é:

Para povoar o data warehouse, os dados precisam primeiro passar pelo processo de ELT. Já os dados armazenados no data lake são decorrentes do processo de ETL.

Questão 5

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere as seguintes afirmações:

I - De acordo com o modelo de 7Vs, o conceito de big data se refere não apenas a quão volumosos são os dados e a quão rápida é feita a coleta desses dados, mas também a vários outros fatores.

II - Ambientes de computação em nuvem são ideais para lidar com big data porque eles são caracterizados por serem ambientes computacionais com grande capacidade de armazenamento e processamento.

III - No modelo de 7Vs, o conceito de variabilidade se refere ao fato de que os dados podem ser estruturados, semiestruturados e não estruturados.

Assinale a alternativa que contém as afirmações **incorrectas**.

- a.
- I e II.
- b.
- Somente III.
- c.
- Somente II.
- d.
- I e III.
- e.
- II e III.

Feedback

Sua resposta está correta.

A resposta correta é:

Somente III.

AS3

Questão 1

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere 3 diferentes fontes de dados sobre funcionários. Os itens a seguir indicam algumas características dessas fontes:

(I) Fonte 1: Funcionario.csv

Fonte 2: Colaborador.json

Fonte 3: Empregado.xlsx

(II) Fonte 1!Sexo: 'Feminino'

Fonte 2!Sexo: 'F'

Fonte 3!Sexo: 0

(III) Fonte 1!NomeFuncionario: 'João da Silva'

Fonte 2!NomeColaborador: 'João da Silva'

Fonte 3!NomeEmpregado: 'João da Silva'

(IV) Fonte 1!DataNascimento: '1990-01-01'

Fonte 2!DataNasc: '01/01/1990'

Fonte 3!Nascimento: '01 JAN 1990'

(V) Fonte 1!EstadoSigla: 'SP'

Fonte 2!EstadoSigla: 'SP'

Fonte 3!EstadoSigla: 'SP'

Considere que o projeto do data mart é representado pela Fonte 1. Indique quais itens necessitam de integração de esquemas e/ou instâncias:

a.

Integração de Esquemas: (I) e (II)

Integração de Instâncias: (III), (IV) e (V)

b.

Integração de Esquemas: (I) e (IV)

Integração de Instâncias: (II) e (IV)

c.

Integração de Esquemas: (I), (III) e (IV)

Integração de Instâncias: (II) e (IV)

d.

Integração de Esquemas: (I) e (V)

Integração de Instâncias: (II), (III) e (IV)

e.

Integração de Esquemas: (I) e (IV)

Integração de Instâncias: (II), (III) e (V)

Feedback

Sua resposta está correta.

A resposta correta é:

Integração de Esquemas: (I), (III) e (IV)

Integração de Instâncias: (II) e (IV)

Questão 2

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere as seguintes afirmações a respeito do processo ETL (extract, transform, load) de geração de dados de alunos a partir de fontes de dados presentes no ambiente operacional.

(I) Devem ser definidos mapeamentos que especificam como cada esquema de cada fonte se relaciona com o esquema do data mart.

(II) A integração deve ser feita pelo atributo NUSP. Alunos que possuem o mesmo valor desse atributo nas diferentes fontes referem-se ao mesmo aluno.

(II) O sexo dos alunos no data mart deve ser representado por 'feminino' e 'masculino', apesar deles serem representados por '0' e '1' em algumas fontes e por 'F' e 'M' em outras fontes.

Assinale a alternativa que representa qual etapa da operação de integração é responsável pelos itens I, II e III, respectivamente.

a.

Resolução de conflitos de valores de atributos, identificação de entidades, resolução de conflitos de valores de atributos.

b.

Integração de esquemas, resolução de conflitos de valores de atributos, identificação de entidades.

c.

Resolução de conflitos de valores de atributos, identificação de entidades, identificação de entidades.

d.

Integração de esquemas, identificação de entidades, identificação de entidades.

e.

Integração de esquemas, identificação de entidades, resolução de conflitos de valores de atributos.

Feedback

Sua resposta está correta.

A resposta correta é:

Integração de esquemas, identificação de entidades, resolução de conflitos de valores de atributos.

Questão 3

Correto

Atingiu 2,00 de 2,00

Texto da questão

O método merge() da biblioteca Pandas pode ser aplicado sobre dois DataFrames com objetivo de:

a.

Converter tipos de dados.

b.

Excluir aleatoriamente linhas da união de duas tabelas.

c.

Unir duas tabelas por meio de uma coluna chave.

d.

Padronizar os dados, modificando letras maiúsculas para minúsculas.

e.

Unir os dados aleatoriamente.

Feedback

Sua resposta está correta.

A resposta correta é:

Unir duas tabelas por meio de uma coluna chave.

Questão 4

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere o código a seguir. O que acontece com o DataFrame dfA após o método apply() ser aplicado?

a.

Será feito o preenchimento da coluna estadoSigla com valores nulos.

b.

Será feito o preenchimento da coluna estado com os dados vindo de mapear_estado_sigla.

c.

Será feito o preenchimento da coluna estadoSigla com os dados vindo de mapear_estado_sigla.

d.

Será feito o preenchimento aleatório da coluna estado.

e.

Será feito o preenchimento aleatório da coluna estadoSigla.

Feedback

Sua resposta está correta.

A resposta correta é:

Será feito o preenchimento da coluna estadoSigla com os dados vindo de mapear_estado_sigla.

Questão 5

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere o código a seguir. Qual o resultado esperado no DataFrame dfR após a aplicação do método merge()?

a.

dfR:

	cpf	nome	sexo	salario
0	43778936450	Joao Souza	M	3000
1	12752689438	Maria Silva	F	10000
2	43778936450	Joao Souza	M	3000
3	12752689438	Andre Nunes	F	10000

b.

dfR:

	cpf	nome	sexo	salario
0	43778936450	Joao Souza	M	3000
1	12752689438	Maria Silva	F	10000

c.

dfR:

	cpf	nome	sexo	salario
0	43778936450	Andre Nunes	M	5000

d.

dfR:

	cpf	nome	sexo	salario
0	43778936450	Joao Souza	M	3000

e.

dfR:

None

Feedback

Sua resposta está correta.

A resposta correta é:

dfR:

	cpf	nome	sexo	salario
0	43778936450	Joao Souza	M	3000
1	12752689438	Maria Silva	F	10000

AS4

Questão 1

Correto

Atingiu 2,00 de 2,00

Texto da questão

Assinale a alternativa correta acerca da modelagem conceitual do processo de ETL (Extração, Transformação e Carga):

a.

Não há necessidade de se documentar um workflow de ETL se ele for feito por um bom desenvolvedor.

b.

A modelagem conceitual do processo de ETL deve ocorrer após a implementação desse processo ter sido finalizada, pois dessa forma o desenvolvedor consegue visualizar melhor todo o processo e então criar uma modelagem mais completa.

c.

Não é recomendado que a modelagem conceitual de ETL seja feita para processos complexos.

d.

A modelagem conceitual do processo de ETL baseada em operadores contribui para facilitar a análise de impacto em casos de evolução do sistema.

e.

Uma boa modelagem sempre é complexa e sempre tem múltiplos fluxos.

Feedback

Sua resposta está correta.

A resposta correta é:

A modelagem conceitual do processo de ETL baseada em operadores contribui para facilitar a análise de impacto em casos de evolução do sistema.

Questão 2

Correto

Atingiu 2,00 de 2,00

Texto da questão

Assinale a alternativa correta sobre operadores de manipulação de dados do Modelo Intuitive:

a.

O operador Sort é utilizado para copiar um conjunto de dados.

b.

O operador Split é usado para combinar dois conjuntos de dados.

c.

O operador Update calcula a diferença entre dois conjuntos de dados.

d.

O operador Diff é usado para atualizar os valores de um ou mais atributos.

e.

O operador Filter é usado para filtrar os itens de dados de um conjunto de dados com base em condições definidas.

Feedback

Sua resposta está correta.

A resposta correta é:

O operador Filter é usado para filtrar os itens de dados de um conjunto de dados com base em condições definidas.

Questão 3

Correto

Atingiu 2,00 de 2,00

Texto da questão

A Figura 1 ilustra o funcionamento do operador Fork (Figura 1a) e do operador Junction (Figura 1b) do Modelo Intuitive. Considerando estes operadores, assinale a alternativa correta.

a.

O operador Fork pode ser usado para representar uma situação na qual um conjunto de dados, vindo de um repositório ou resultante de alguma tarefa, é direcionado para dois ou mais fluxos que são executados de forma concorrente no workflow.

b.

O operador Junction é o temporizador que representa uma pausa no fluxo de trabalho por um período de tempo especificado. Ele é usado para controlar o fluxo de trabalho e garantir que as tarefas sejam executadas em uma ordem específica ou em intervalos regulares.

c.

Os fluxos gerados pelo operador Fork não podem ser executados de forma concorrente no workflow.

d.

A saída de um operador Junction sempre deve ser direcionada para um repositório de log de erros.

e.

A saída do operador Fork é, obrigatoriamente, unária.

Feedback

Sua resposta está correta.

A resposta correta é:

O operador Fork pode ser usado para representar uma situação na qual um conjunto de dados, vindo de um repositório ou resultante de alguma tarefa, é direcionado para dois ou mais fluxos que são executados de forma concorrente no workflow.

Questão 4

Correto

Atingiu 2,00 de 2,00

Texto da questão

A Figura 2 ilustra os operadores de agregação do modelo Intuitive. Considerando estes operadores, assinale a alternativa correta:

- a.
- O operador Avg não é necessário ao modelo, desde que para calcular o resultado produzido por esse operador basta somar os resultados produzidos pelos operadores Max e Min e dividir por 2.
- b.
- O operador MinGroup produz, para cada grupo, o menor valor de cada atributo.
- c.
- Não existem diferenças práticas entre o operador Sum e o operador SumGroup.
- d.
- Para cada atributo, o operador Count soma os seus valores e produz um único valor como resultado.
- e.
- O operador Max produz, para cada grupo, o maior valor de cada atributo.

Feedback

Sua resposta está correta.

A resposta correta é:

O operador MinGroup produz, para cada grupo, o menor valor de cada atributo.

Questão 5

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere o diagrama conceitual modelado na Figura 3. Assinale a alternativa que corresponde ao que está sendo modelado no diagrama.

- a.
- O workflow representa a sincronização dos fluxos para o tratamento dos atributos "Order.status", "Order.custKey", "Order.ordKey" com "L.status", "L.cust" e "L.ord", respectivamente, sendo que a condição de finalização é a atualização de "DW.Order".
- b.
- O workflow representa a separação dos atributos do conjunto "Order" (ou seja, "Order.status", "Order.custKey", "Order.ordKey"), direcionando-os primeiro para sucessivos fluxos até serem sincronizados e atualizados em "DW.Order".
- c.
- Os atributos "Order.status", "Order.custKey", "Order.ordKey" são usados para junções sucessivas com "L.status", "L.cust" e "L.ord", respectivamente, sendo que o conjunto de dados resultante é usado para a atualização de "DW.Order".
- d.
- "DW.Order" representa o ponto de encontro das tarefas que realizam de forma concorrente o tratamento dos atributos "Order.status", "Order.custKey", "Order.ordKey" com "L.status", "L.cust" e "L.ord", respectivamente.
- e.
- Os atributos "Order.status", "Order.custKey", "Order.ordKey" são usados sucessivamente por meio da aplicação do operador Fork com "L.status", "L.cust" e "L.ord", respectivamente, sendo que os conjuntos de dados resultantes são direcionados para fluxos que são executados de forma concorrente no workflow, até serem usados para a atualização de "DW.Order".

Feedback

Sua resposta está correta.

A resposta correta é: Os atributos "Order.status", "Order.custKey", "Order.ordKey" são usados para junções sucessivas com "L.status", "L.cust" e "L.ord", respectivamente, sendo que o conjunto de dados resultante é usado para a atualização de "DW.Order".

AS5

Questão 1

Correto

Atingiu 2,00 de 2,00

Texto da questão

Assinale a alternativa correta acerca dos dados de um data warehouse:

- a.
Dados menos detalhados possuem grão grande.
- b.
A variedade de consultas que podem ser respondidas independe da granularidade dos dados.
- c.
A granularidade não impacta no volume de dados armazenado no data warehouse.
- d.
Dados mais detalhados são dados mais agregados.
- e.
O data warehouse é mais volumoso quando o grão é grande.

Feedback

Sua resposta está correta.

A resposta correta é:

Dados menos detalhados possuem grão grande.

Questão 2

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere as seguintes afirmações acerca das operações OLAP:

- I. A operação dice aumenta a dimensionalidade do cubo de dados.
- II. A operação slice diminui a dimensionalidade do cubo de dados.
- III. A operação pivot está relacionada com a seleção de dados.
- IV. A operação roll-up está relacionada com agrupamentos dos dados.
- V. A operação drill-across compara métricas de dois cubos de dados diferentes que compartilham a mesma dimensão em níveis de agregação distintos.

Considerando V (Verdadeiro) e F (Falso), assinale a alternativa correta:

- a.
F - V - V - F - V.
- b.
F - F - F - V - V.
- c.
F - V - F - V - F.

- d.
V - V - V - V - F.
- e.
V - F - V - F - V.

Feedback

Sua resposta está correta.

A resposta correta é:
F - V - F - V - F.

Questão 3

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere as seguintes afirmações relacionadas aos tipos de esquema utilizados para organizar a visão multidimensional dos dados do data warehouse.

- I. O esquema estrela possui uma tabela de fatos dominante em seu centro e um conjunto de tabelas de dimensão nas extremidades.
- II. O esquema floco de neve consiste em uma extensão do esquema estrela na qual as extremidades do esquema estrela são normalizadas utilizando como base hierarquias de atributos.
- III. Em contrate às tabelas de dimensão, que usualmente possuem um menor número de linhas e uma grande quantidade de colunas, as tabelas de fatos geralmente possuem um grande número de linhas e uma quantidade reduzida de colunas.
- IV. Uma constelação de fatos possui duas ou mais tabelas de fatos e representa as tabelas de dimensão em comum apenas uma vez.

Assinale o número de afirmações corretas:

- a.
0
- b.
1
- c.
2
- d.
4
- e.
3

Feedback

Sua resposta está correta.

A resposta correta é:
4

Questão 4

Correto
Atingiu 2,00 de 2,00

Texto da questão

Considere a constelação de fatos da BI Solutions. Assinale a alternativa que corresponde à resposta para a seguinte consulta analítica: "Liste a quantidade de negociações realizadas no segundo trimestre do ano de 2019".

- a.
1012
- b.
306
- c.
313
- d.
1222
- e.
223

Feedback

Sua resposta está correta.

A resposta correta é:

306

Questão 5

Correto
Atingiu 2,00 de 2,00

Texto da questão

Considere a constelação de fatos da BI Solutions. Assinale a alternativa que corresponde à resposta para a seguinte consulta analítica: "Liste o total de receita referente à equipe de BI & ANALYTICS no ano de 2019".

- a.
R\$ 2.382.301,70
- b.
R\$ 19.522.346,45
- c.
R\$ 35.353.318,30
- d.
R\$ 23.577.872,90
- e.
R\$ 43.100.219,35

Feedback

Sua resposta está correta.

A resposta correta é:

R\$ 23.577.872,90

AS6

Questão 1

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere um cenário no qual Apache Spark está sendo usado para processar um grande conjunto de dados. Foram realizadas várias operações de transformação, como map e reduce, porém nenhuma dessas operações foi executada imediatamente. Em seguida, é realizada uma operação de ação e, de repente, os cálculos começam. Assinale a alternativa que ilustra o conceito do Apache Spark que foi empregado.

- a.
Conjuntos de dados distribuídos e resilientes (RDDs).
- b.
Garantia de tolerância a falhas.
- c.
Lazy-evaluation.
- d.
Garantia de disponibilidade dos dados.
- e.
Agendador de grafos acíclicos e direcionados (DAGs).

Feedback

Sua resposta está correta.

A resposta correta é:

Lazy-evaluation.

Questão 2

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere um cenário no qual existe um grande arquivo de texto que é utilizado como entrada para a tarefa de contar a frequência com que cada palavra aparece utilizando o Apache Spark. Considere também que a implementação desta tarefa é feita utilizando o modelo de programação funcional MapReduce. Adicionalmente, considere que o método spark.textFile() já foi aplicado para ler o arquivo e retorná-lo como um RDD (conjuntos de dados distribuídos resilientes) de strings. Assinale a alternativa que especifica qual sequência de métodos descreve a implementação da tarefa.

- a.
reduceByKey(), map(), flatMap(), collect().

- b.
flatMap(), reduceByKey(), map(), collect().
- c.
flatMap(), map(), reduceByKey(), collect().
- d.
map(), reduceByKey(), flatMap(), collect().
- e.
reduceByKey(), flatMap(), map(), collect().

Feedback

Sua resposta está correta.

A resposta correta é:

flatMap(), map(), reduceByKey(), collect().

Questão 3

Correto

Atingiu 2,00 de 2,00

Texto da questão

Uma empresa de desenvolvimento de software está planejando mover suas operações para a nuvem. Eles querem gerenciar suas próprias aplicações e dados, mas não querem se preocupar com o tempo de execução, middleware, sistema operacional, virtualização, servidores, armazenamento ou rede. Qual modelo de serviço de computação em nuvem é o mais apropriado para esta empresa?

- a.
Função como Serviço (FaaS).
- b.
Infraestrutura como Serviço (IaaS).
- c.
Plataforma como Serviço (PaaS).
- d.
Software como Serviço (SaaS).
- e.
Banco de Dados como Serviço (DBaaS).

Feedback

Sua resposta está correta.

A resposta correta é:

Plataforma como Serviço (PaaS).

Questão 4

Incorreto

Atingiu 0,00 de 2,00

Texto da questão

Assinale a alternativa correta acerca do sistema de arquivos distribuídos HDFS (Hadoop Distributed File System).

- a.
O HDFS é baseado no uso de conjuntos de dados distribuídos e resilientes (RDDs).
- b.
O HDFS não é baseado em uma arquitetura mestre-escravo porque existem apenas DataNodes.
- c.
O NameNode é considerado o nó mestre dentro da arquitetura HDFS, tendo como uma de suas responsabilidades armazenar metadados sobre os dados armazenados.
- d.
Os DataNodes fazem a manutenção de metadados sobre os dados armazenados.
- e.
Os NameNodes armazenam os arquivos de dados de forma particionada e replicada.

Feedback

Sua resposta está incorreta.

A resposta correta é:

O NameNode é considerado o nó mestre dentro da arquitetura HDFS, tendo como uma de suas responsabilidades armazenar metadados sobre os dados armazenados.

Questão 5

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere o código especificado a seguir. Considere que a variável “lista_de_tweets” é uma lista de conteúdos textuais em inglês com aproximadamente 1 milhão de tweets que contêm a palavra bitcoin. Considere também que a função “analisa_texto” retorna o valor 1 no caso do tweet apresentar uma consideração positiva sobre bitcoins ou retorna o valor -1 em caso contrário.

Considere as seguintes afirmações:

- I A única operação do tipo ação presente no código é o método reduce. As demais operações são transformações.

II O código funciona sem erros e seu objetivo principal é estimar o interesse sobre bitcoin.

III No caso da configuração a seguir, o valor de output é igual a 1.

IV Se o valor de output for negativo, então a maioria dos tweets analisados apresentam considerações desfavoráveis sobre bitcoins.

V Se o valor de output for positivo, então a maioria dos tweets analisados apresentam considerações desfavoráveis sobre bitcoins.

Assinale a alternativa que contém as afirmações corretas:

- a.
I, II, IV, V.
- b.
I, III, IV.
- c.
I, II, III, V.
- d.
I, II, IV.
- e.
II, III, V.

Feedback

Sua resposta está correta.

A resposta correta é:

I, II, III, V.

AS7

Questão 1

Correto

Atingiu 2,00 de 2,00

Texto da questão

Assinale a alternativa que apresenta a ordem correta de processamento lógico das cláusulas em uma consulta SQL.

- a.
WHERE, FROM, SELECT, GROUP BY, HAVING, ORDER BY
- b.
SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY

- c.
FROM, HAVING, GROUP BY, WHERE, SELECT, ORDER BY
d.
FROM, SELECT, WHERE, GROUP BY, ORDER BY, HAVING
e.
FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY

Feedback

Sua resposta está correta.

A resposta correta é:

FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY

Questão 2

Correto

Atingiu 2,00 de 2,00

Texto da questão

Assinale a alternativa que mostra a quantidade correta de níveis de agregação gerados pelas seguintes consultas:

i) SELECT dataPK, funcPK, cargoPK, SUM(salario) AS `Salário (R\$)`
FROM pagamento
GROUP BY ROLLUP (dataPK, funcPK, cargoPK)
ii) SELECT dataPK, funcPK, cargoPK, SUM(salario) AS `Salário (R\$)`
FROM pagamento
GROUP BY CUBE (dataPK, funcPK, cargoPK)
iii) SELECT dataPK, funcPK, cargoPK, SUM(salario) AS `Salário (R\$)`
FROM pagamento
GROUP BY GROUPING SETS
(
(dataPK, funcPK, cargoPK), (dataPK, funcPK), (dataPK), ()
)

- a.
i) 3 níveis, ii) 27 níveis, iii) 4 níveis.
b.
i) 3 níveis, ii) 9 níveis, iii) 6 níveis.
c.
i) 4 níveis, ii) 8 níveis, iii) 6 níveis.
d.
i) 3 níveis, ii) 9 níveis, iii) 7 níveis.
e.
i) 4 níveis, ii) 8 níveis, iii) 4 níveis.

Feedback

Sua resposta está correta.

A resposta correta é:

i) 4 níveis, ii) 8 níveis, iii) 4 níveis.

Questão 3

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere as seguintes afirmações acerca da linguagem SQL:

- I. A linguagem SQL é amplamente utilizada por sua simplicidade, facilidade de utilização e grande poder de consulta.
- II. A especificação de duas relações na cláusula FROM sem a especificação de uma condição de junção referente a essas duas relações gera um produto cartesiano das relações.
- III. A linguagem SQL descreve o problema ao invés da solução, indicando quais dados devem ser obtidos na resposta da consulta, e não como esses dados devem ser obtidos.
- IV. A cláusula WHERE especifica o predicado que seleciona as tuplas enquanto que a cláusula ORDER BY agrupa os dados exibidos como resposta à consulta.
- V. No processamento lógico das cláusulas de SQL, a cláusula SELECT é sempre processada antes da cláusula FROM.

Assinale a alternativa que contém as afirmações corretas.

- a.
III, IV e V
- b.
II, III e IV
- c.
II e III
- d.
I e II
- e.
I, II e III

Feedback

Sua resposta está correta.

A resposta correta é: I, II e III

Questão 4

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considerando a constelação de fatos da BI Solutions, assinale a alternativa que corresponde à consulta SQL para: "Qual a média de salários no ano de 2019, considerando cada cargo e seus respectivos níveis? Ordene o resultado final pela média de salários, da menor média para a maior média."

- a.

```

SELECT cargoNome, cargoNivel, AVG(salario)
FROM cargo JOIN pagamento ON cargo.cargoPK = pagamento.cargoPK
JOIN data ON data.dataPK = pagamento.dataPK
WHERE dataAno = 2019
GROUP BY cargoNome, cargoNivel
ORDER BY AVG(salario) ASC
b.
SELECT cargoNome, AVG(salario)
FROM cargo JOIN pagamento ON cargo.cargoPK = pagamento.cargoPK
JOIN data ON data.dataPK = pagamento.dataPK
WHERE dataAno = 2019
GROUP BY cargoNome
ORDER BY AVG(salario)
c.
SELECT cargoNome, cargoNivel, AVG(salario)
FROM cargo JOIN pagamento ON cargo.cargoPK = pagamento.cargoPK
WHERE dataAno = 2019
GROUP BY cargoNome, cargoNivel
ORDER BY AVG(salario) ASC
d.
SELECT cargoNome, cargoNivel, AVG(salario)
FROM cargo JOIN pagamento ON cargo.cargoPK = pagamento.cargoPK
JOIN data ON data.dataPK = pagamento.dataPK
GROUP BY cargoNome, cargoNivel
ORDER BY AVG(salario)
e.
SELECT cargoNome, cargoNivel, AVG(salario)
FROM cargo JOIN pagamento ON cargo.cargoPK = pagamento.cargoPK
JOIN data ON data.dataPK = pagamento.dataPK
WHERE dataAno = 2019 AND cargoNivel = 'SENIOR'
GROUP BY cargoNome, cargoNivel
ORDER BY AVG(salario) ASC

```

Feedback

Sua resposta está correta.

A resposta correta é:

```

SELECT cargoNome, cargoNivel, AVG(salario)
FROM cargo JOIN pagamento ON cargo.cargoPK = pagamento.cargoPK
JOIN data ON data.dataPK = pagamento.dataPK
WHERE dataAno = 2019
GROUP BY cargoNome, cargoNivel
ORDER BY AVG(salario) ASC

```

Questão 5

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considerando a constelação de fatos da BI Solutions, assinale a alternativa que corresponde à resposta para a seguinte consulta analítica: "Qual a quantidade de lançamentos realizados no mês de dezembro de 2019?"

a.

180

b.

150

c.

130

d.

200

e.

120

Feedback

Sua resposta está correta.

A resposta correta é:

200

AS8

Questão 1

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere o trecho de código em pyspark ilustrado na Figura 1:

Figura 1: Trecho de código em pyspark.

Assinale a alternativa que contém a afirmação correta.

a.

O resultado da execução completa do trecho de código inclui apenas as frutas que estão presentes em ambos df1 e df2.

b.

A operação filter() é usada para selecionar apenas as linhas cuja contagem dos elementos da segunda coluna seja maior do que 2, de forma que filtered_df possua 2 linhas.

c.

A operação join() é utilizada para combinar df1 e df2 com base na coluna fruit, de forma que joined_df possua as mesmas linhas e as mesmas colunas que df1.

d.

A operação select() é usada para selecionar apenas as colunas fruit e count de df1, de forma que result_df contenha as duas últimas colunas de filtered_df.

e.

O resultado da execução completa do trecho de código inclui todas as frutas presentes em df1.

Feedback

Sua resposta está correta.

A resposta correta é:

A operação filter() é usada para selecionar apenas as linhas cuja contagem dos elementos da segunda coluna seja maior do que 2, de forma que filtered_df possua 2 linhas.

Questão 2

Correto

Atingiu 2,00 de 2,00

Texto da questão

Suponha que você seja um cientista de dados que trabalha para uma grande empresa de comércio eletrônico. A empresa possui um enorme conjunto de dados de transações de clientes e detalhes de produtos. Suponha também que existam dois DataFrames Spark especificados conforme segue:

- transactions_df, o qual possui as seguintes colunas:
customer_id, product_id, transaction_date, quantity e price.
- products_df, o qual possui as seguintes colunas:
product_id, product_name, product_category

Considere a seguinte solicitação de consulta: "Liste a soma das receitas que cada categoria de produto gerou no ano de 2023", sendo que receita é obtida considerando a multiplicação da coluna quantity pela coluna price.

Assinale a alternativa que corresponde à consulta em pyspark.

a.

```
transactions_df.filter(year("transaction_date") == 2023)
.join(products_df, on="product_id")
.groupBy("product_category")
.agg(avg("price") * "quantity")
```

b.

```
transactions_df.join(products_df, on="product_id")  
.filter(year("transaction_date") == 2023)  
.groupBy("product_category")  
.agg(sum("quantity"))  
C.  
transactions_df.join(products_df, on="product_id")  
.filter(year("transaction_date") == 2023)  
.groupBy("product_category")  
.agg(avg("price" / "quantity"))  
d.  
transactions_df.join(products_df, on="product_id")  
.filter(year("transaction_date") == 2023)  
.groupBy("product_category")  
.agg(sum("price" / "quantity"))  
e.  
transactions_df.filter(year("transaction_date") == 2023)  
.join(products_df, on="product_id")  
.groupBy("product_category")  
.agg(sum("price"))
```

Feedback

Sua resposta está correta.

A resposta correta é:

```
transactions_df.join(products_df, on="product_id")  
.filter(year("transaction_date") == 2023)  
.groupBy("product_category")  
.agg(sum("price" / "quantity"))
```

Questão 3

Correto

Atingiu 2,00 de 2,00

Texto da questão

Assinale a alternativa correta acerca das comparações entre Pandas, Spark RDD e Spark SQL.

a.

Especificar uma consulta usando a linguagem SQL como um parâmetro do método spark.sql() indica que o usuário deve especificar como os dados devem ser obtidos ao invés de quais dados devem ser obtidos.

b.

Tanto a biblioteca Pandas quanto o módulo pyspark.sql utilizam o conceito de DataFrames, sendo que não existem diferenças entre um DataFrame em Pandas e um DataFrame em Spark.

c.

É possível responder à consulta “Qual é a média dos salários recebidos por nível do cargo e por sexo no ano de 2019?” definida sobre a constelação de fatos da BI Solutions utilizando Pandas, especificando a consulta SQL textual como um parâmetro do método spark.sql() e utilizando métodos do módulo pyspark.sql, obtendo, entretanto, respostas diferentes.

d.

Especificar uma consulta usando a linguagem SQL como um parâmetro do método spark.sql() indica que o usuário deve especificar como os dados devem ser obtidos ao invés de quais dados devem ser obtidos.

e.

É possível responder à consulta "Qual é a média dos salários recebidos por nível do cargo e por sexo no ano de 2019?" definida sobre a constelação de fatos da BI Solutions especificando a consulta SQL textual como um parâmetro do método spark.sql() e utilizando métodos do módulo pyspark.sql. A consulta não pode ser respondida utilizando Pandas porque essa linguagem não oferece suporte para o processamento paralelo e distribuído.

Feedback

Sua resposta está correta.

As respostas corretas são:

É possível responder à consulta "Qual é a média dos salários recebidos por nível do cargo e por sexo no ano de 2019?" definida sobre a constelação de fatos da BI Solutions utilizando Pandas, especificando a consulta SQL textual como um parâmetro do método spark.sql() e utilizando métodos do módulo pyspark.sql, obtendo, entretanto, respostas diferentes.,

Tanto a biblioteca Pandas quanto o módulo pyspark.sql utilizam o conceito de DataFrames, sendo que não existem diferenças entre um DataFrame em Pandas e um DataFrame em Spark.,

Especificar uma consulta usando a linguagem SQL como um parâmetro do método spark.sql() indica que o usuário deve especificar como os dados devem ser obtidos ao invés de quais dados devem ser obtidos.,

Especificar uma consulta usando a linguagem SQL como um parâmetro do método spark.sql() indica que o usuário deve especificar como os dados devem ser obtidos ao invés de quais dados devem ser obtidos.,

É possível responder à consulta "Qual é a média dos salários recebidos por nível do cargo e por sexo no ano de 2019?" definida sobre a constelação de fatos da BI Solutions especificando a consulta SQL textual como um parâmetro do método spark.sql() e utilizando métodos do módulo pyspark.sql. A consulta não pode ser respondida utilizando Pandas porque essa linguagem não oferece suporte para o processamento paralelo e distribuído.

Questão 4

Correto

Atingiu 2,00 de 2,00

Texto da questão

Assinale a alternativa correta acerta dos métodos do módulo pyspark.sql.

a.

O método join() somente realiza junção do tipo inner, não sendo possível especificar outros tipos de junção.

b.

O método `orderBy()` ordena os dados exibidos como resposta à consulta de modo descendente por padrão, sendo necessário definir explicitamente o modo ascendente caso seja necessário.

c.

O método `filter()` filtra linhas de um DataFrame de acordo com uma condição especificada como parâmetro.

d.

Os métodos `rollup()` e `cube()` criam vários níveis de agregação usando como base as colunas especificadas como parâmetro, gerando o mesmo resultado.

e.

O método `select()` seleciona linhas específicas de um DataFrame de acordo com uma condição especificada como parâmetro.

Feedback

Sua resposta está correta.

A resposta correta é:

O método `filter()` filtra linhas de um DataFrame de acordo com uma condição especificada como parâmetro.

Questão 5

Correto

Atingiu 2,00 de 2,00

Texto da questão

Considere as seguintes afirmações a respeito dos métodos de Pandas, as cláusulas da linguagem SQL e os métodos de pyspark.

- I A filtragem de dados pode ser feita usando o método `query()` em Pandas, a cláusula WHERE em SQL e o método `filter()` em pyspark.
- II A ordenação de dados pode ser feita usando o método `sort_values()` em Pandas, a cláusula ORDER BY em SQL e o método `orderBy()` em pyspark.
- III O agrupamento de dados pode ser feito usando o método `groupby()` em Pandas, a cláusula GROUP BY em SQL e o método `groupBy()` em pyspark.
- IV A renomeação de colunas pode ser feito usando o método `rename()` em Pandas, a cláusula AS em SQL e o método `reduce()` em pyspark.
- V A seleção de colunas a serem exibidas pode ser feita usando o método `sel()` em Pandas, a cláusula SELECT em SQL e o método `select()` em pyspark.

Assinale a alternativa que contém as afirmações corretas.

a.

I e II

b.

II e III

c.

II, III e IV

- d.
- III, IV e V
- e.
- I, II e III

Feedback

Sua resposta está correta.

A resposta correta é:
I, II e III

AF

Questão 1

Correto

Atingiu 1,00 de 1,00

Texto da questão

Uma empresa de transporte público de uma grande capital brasileira está enfrentando um desafio para analisar seus dados. A empresa possui sensores em cada um de seus ônibus, os quais enviam sua localização em tempo real para um banco de dados relacional localizado na sede da empresa. Além disso, a empresa também gerencia dados cadastrais relacionados aos ônibus, estações, motoristas e pontos disponíveis, os quais são armazenados em diferentes planilhas Excel, cada uma com um formato de padronização distinto, e são atualizados uma vez por semana. Os gestores precisam fazer diferentes tipos de análises, incluindo traçar rotas em tempo real para os ônibus e realizar investigações históricas dos dados cadastrais. Considerando este contexto, os gestores da empresa solicitaram a implementação de uma aplicação de big data warehousing englobando todas as fontes de dados supracitadas com o intuito de executar consultas analíticas.

Assinale a alternativa que corresponde à melhor estratégia de implementação.

- a.
Deve-se utilizar tanto um data lake quanto um data warehouse para se carregar os dados. O data lake pode receber os dados dos sensores em tempo real via streaming de dados devido à sua frequência de atualização, enquanto recebe o restante dos dados em cargas diárias. Uma vez carregados no data lake, os dados passam por transformações e são carregados no data warehouse a fim de otimizar as consultas analíticas. Desta forma, tem-se consultas em tempo real para os dados dos sensores dos ônibus no data lake e consultas otimizadas para todos os dados históricos no data warehouse.
- b.
Deve-se criar um ambiente informacional para a empresa contendo tanto um data warehouse quanto um data lake. O data warehouse deve armazenar os dados em seu formato bruto, permitindo que dados que são gerados em tempo real, como os dados dos sensores, sejam consultados imediatamente. Já o data lake deve armazenar os dados após um processo de transformação para um esquema estrela, permitindo maior otimização no tempo de resposta das consultas analíticas.

c.

Somente um data lake deve ser utilizado na implementação do ambiente. No data lake, tanto os dados cadastrais quanto os dados de sensores devem ser armazenados segundo o esquema estrela, possibilitando o carregamento e a análise dos dados em tempo real e de maneira otimizada.

d.

Deve-se criar uma solução capaz de carregar os dados gerados pelos sensores dos ônibus diretamente nas planilhas Excel que contêm os dados cadastrais. Assim, é possível aproveitar o poder de processamento das máquinas utilizadas diariamente pela empresa para executar consultas analíticas diretamente nas planilhas, reduzindo o custo de implementação do projeto de forma considerável.

e.

Somente um data warehouse deve ser utilizado na implementação do ambiente. Como o data warehouse permite o carregamento dos dados em um esquema estrela em tempo real, tanto os dados de sensores em seu formato bruto quanto os dados cadastrais das planilhas podem ser armazenados imediatamente para a execução de consultas analíticas.

Feedback

Sua resposta está correta.

A resposta correta é:

Deve-se utilizar tanto um data lake quanto um data warehouse para se carregar os dados. O data lake pode receber os dados dos sensores em tempo real via streaming de dados devido à sua frequência de atualização, enquanto recebe o restante dos dados em cargas diárias. Uma vez carregados no data lake, os dados passam por transformações e são carregados no data warehouse a fim de otimizar as consultas analíticas. Desta forma, tem-se consultas em tempo real para os dados dos sensores dos ônibus no data lake e consultas otimizadas para todos os dados históricos no data warehouse.

Questão 2

Correto

Atingiu 1,00 de 1,00

Texto da questão

Considere a etapa de integração de instâncias do processo de ETL relativa ao código em Pandas ilustrado a seguir. Considere que a função `processar_conflitos` recebe como parâmetro três fontes de dados, os índices e a coluna a ser processada. Considere também que a função retorna uma tabela em Pandas com o resultado final, processando apenas a coluna especificada no parâmetro. Adicionalmente, considere alguns exemplos de dados armazenados nas fontes "fonte1", "fonte2" e "fonte3".

Assinale a alternativa que corresponde à execução do processamento de conflitos.

a.

De acordo com a lógica da função `processar_conflitos`, nunca haverá registro com o valor vazio. No resultado final da variável “`df`”, o registro de código “03” contém o nome “José” e o e-mail “`joao@email.com`”.

b.

De acordo com a lógica da função `processar_conflitos`, nunca haverá registro com o valor vazio. No resultado final da variável “`df`”, o registro de código “03” contém o nome “Maria” e o e-mail “`mariaaa@email.com`”.

c.

De acordo com a lógica da função processar_conflitos, nunca haverá registro com o valor vazio. No resultado final da variável "df", o registro de código "02" contém o nome "João" e o e-mail "None".

d.

De acordo com a lógica da função processar_conflitos, pode haver registro com o valor vazio. No resultado final da variável "df", o registro de código "01" contém o nome "Maria" e o e-mail "mariaaaa@email.com".

e.

De acordo com a lógica da função processar_conflitos, pode haver registro com o valor vazio. No resultado final da variável "df", o registro de código "01" contém o nome "Mariah" e o e-mail "mariaaaa@email.com".

Feedback

Sua resposta está correta.

A resposta correta é:

De acordo com a lógica da função processar_conflitos, pode haver registro com o valor vazio. No resultado final da variável "df", o registro de código "01" contém o nome "Mariah" e o e-mail "mariaaaa@email.com".

Questão 3

Correto

Atingiu 1,00 de 1,00

Texto da questão

Considere o diagrama conceitual modelado na Figura 1. Assinale a alternativa que corresponde ao que está sendo modelado no diagrama.

Figura 1: Diagrama conceitual.

a.

Os atributos "PK", "Destino", "Data" e "Preço" passam por uma operação de junção com os dados da fonte "passagens aereas" e também por uma função de tratamento que adiciona o atributo "Região", que usa como base o atributo "Destino" previamente extraído. Na sequência, os dados são primeiramente utilizados para a execução de tarefas paralelas de agregação e depois armazenados em um único data mart final contendo os dados de todas as regiões do Brasil.

b.

Os atributos "PK", "Destino", "Data" e "Preço" passam por uma operação de junção com os dados da fonte "passagens aereas" e também por uma função de tratamento que adiciona o atributo "Região", que usa como base o atributo "Preço" previamente extraído. Na sequência, os dados são primeiramente utilizados para a execução de tarefas paralelas de agregação, depois sincronizados por meio da espera da execução de tarefas de agregação e finalmente filtrados usando o novo atributo "Região", para então serem armazenados em seus respectivos data marts regionais.

c.

Os atributos "PK", "Destino", "Data" e "Preço" são extraídos da fonte "passagens aereas" e passam por uma função de tratamento que adiciona o atributo "Região", que usa como base o atributo "Destino" previamente extraído. Na sequência, os dados são utilizados para a execução de tarefas paralelas de agregação e depois filtrados usando o novo atributo "Região", para então serem armazenados em seus respectivos data marts regionais.

d.

Os atributos "PK", "Destino", "Data" e "Preço" são extraídos da fonte "passagens aereas" e passam por uma função de tratamento que adiciona o atributo "Região", que usa como base o atributo "Preço" previamente extraído. Na sequência, os dados são primeiramente utilizados para a execução de tarefas paralelas de agregação e depois filtrados usando o atributo "Destino", para então serem armazenados em seus respectivos data marts regionais.

e.

Os atributos "PK", "Destino", "Data" e "Preço" passam por uma operação de junção com os dados da fonte "passagens aereas" e também por uma função de tratamento que adiciona o atributo "Região", que usa como base o atributo "Data" previamente extraído. Na sequência, os dados são primeiramente utilizados para a execução de tarefas paralelas de agregação e depois filtrados usando o atributo "Destino", para então serem armazenados em seus respectivos data marts regionais.

Feedback

Sua resposta está correta.

A resposta correta é:

Os atributos "PK", "Destino", "Data" e "Preço" são extraídos da fonte "passagens aereas" e passam por uma função de tratamento que adiciona o atributo "Região", que usa como base o atributo "Destino" previamente extraído. Na sequência, os dados são utilizados para a execução de tarefas paralelas de agregação e depois filtrados usando o novo atributo "Região", para então serem armazenados em seus respectivos data marts regionais.

Questão 4

Correto

Atingiu 1,00 de 1,00

Texto da questão

Considere o código especificado a seguir, o qual utiliza o método merge() sobre três DataFrames criados a partir dos dicionários "alunos", "turma1" e "turma2".

Assinale a alternativa que corresponde ao resultado esperado no DataFrame dfR após a aplicação das duas ocorrências do método merge().

a.

dfR:

cpf	nota_progamacao	nota_calculo	nome	idad
0 204.927.060-77	9.0	NaN	Joao Paulo	21
1 116.948.760-20	5.0	4.0	Jose Carlos	23
2 327.639.610-61	10.0	8.0	Maria Eduarda	20

3 904.716.030-40	8.0	9.0	Ana Julia	21
4 750.286.140-83	NaN	3.0	Carlos Alberto	22

b.

dfR:

cpf	nota_progamacao	nota_calculo	nome	idade
0 204.927.060-77	9.0	NaN	Joao Paulo	21
1 116.948.760-20	5.0	NaN	Jose Carlos	23
2 327.639.610-61	10.0	8.0	Maria Eduarda	20
3 904.716.030-40	8.0	9.0	Ana Julia	21

c.

dfR:

cpf	nota_progamacao	nota_calculo	nome	idade
0 204.927.060-77	9.0	NaN	Joao Paulo	21
1 327.639.610-61	10.0	8.0	Maria Eduarda	20
2 904.716.030-40	8.0	9.0	Ana Julia	21

d.

dfR:

cpf	nota_progamacao	nota_calculo	nome	idade
0 327.639.610-61	10.0	8.0	Maria Eduarda	20
1 904.716.030-40	8.0	9.0	Ana Julia	21

e.

dfR:

cpf	nota_progamacao	nota_calculo	nome	idade
0 116.948.760-20	5.0	4.0	Jose Carlos	23
1 327.639.610-61	10.0	8.0	Maria Eduarda	20
2 904.716.030-40	8.0	9.0	Ana Julia	21

Feedback

Sua resposta está correta.

A resposta correta é:

dfR:

cpf	nota_progamacao	nota_calculo	nome	idade
0 327.639.610-61	10.0	8.0	Maria Eduarda	20
1 904.716.030-40	8.0	9.0	Ana Julia	21

Questão 5

Correto

Atingiu 1,00 de 1,00

Texto da questão

Considere o trecho de código especificado a seguir.

Considere as seguintes afirmações.

- I A transformação flatMap() é usada para criar um novo RDD a partir do retorno da função split() sobre cada elemento do RDD.
- II O código realiza a contagem de cada palavra, exceto a palavra Spark.
- III As palavras mais frequentes retornadas pelo comando especificado na linha 17 são [('É', 2), ('Olá', 2)].
- IV A transformação filter() é usada para filtrar apenas a palavra Spark e adicioná-la ao RDD.
- V A ação collect() é usada para retornar uma lista com elementos do RDD e suas respectivas contagens.

Assinale a alternativa que contém apenas as afirmações corretas:

- a.
- I, II, III.
- b.
- I e V.
- c.
- II, IV.

- d.
- I, II, V.
- e.
- II, IV, V.

Feedback

Sua resposta está correta.

A resposta correta é:
I, II, V.

Questão 6

Correto
Atingiu 1,00 de 1,00

Texto da questão

Considere a constelação de fatos da BI Solutions e a seguinte solicitação de consulta:

"Liste a média dos salários recebidos por escolaridade mínima e por sexo em cada ano". Arredonde a soma dos salários para até duas casas decimais. Devem ser exibidas as colunas na ordem e com os nomes especificados a seguir: "ANO", "ESCOLARIDADE", "SEXO", "Média dos Salários (R\$)". Ordene as linhas exibidas primeiro por ano em ordem ascendente, depois por escolaridade em ordem ascendente e depois por sexo em ordem ascendente.

Assinale a alternativa que corresponde à consulta em SQL.

- a.

```
SELECT dataAno AS ANO,
cargoEscolaridadeMinima AS ESCOLARIDADE,
funcSexo AS SEXO,
ROUND(AVG(salario),2) AS `Média dos Salários (R$)`
FROM pagamento JOIN data ON data.dataPK = pagamento.dataPK
JOIN cargo ON cargo.cargoPK = pagamento.cargoPK
JOIN funcionario ON funcionario.funcPK = pagamento.funcPK
GROUP BY ANO, ESCOLARIDADE, SEXO
ORDER BY ANO, ESCOLARIDADE, SEXO
```
- b.

```
SELECT dataAno AS ANO,
cargoEscolaridadeMinima AS ESCOLARIDADE,
funcSexo AS SEXO,
ROUND(PERCENTILE(salario, 0.5),2) AS `Mediana dos Salários (R$)`
FROM pagamento JOIN data ON data.dataPK = pagamento.dataPK
JOIN cargo ON cargo.cargoPK = pagamento.cargoPK
JOIN funcionario ON funcionario.funcPK = pagamento.funcPK
GROUP BY ANO, ESCOLARIDADE, SEXO
ORDER BY ANO, ESCOLARIDADE, SEXO
```
- c.

```
SELECT funcAnoNascimento AS ANO,
cargoEscolaridadeMinima AS ESCOLARIDADE,
funcSexo AS SEXO,
ROUND(AVG(salario),2) AS `Média dos Salários (R$)`
FROM pagamento JOIN data ON data.dataPK = pagamento.dataPK
JOIN cargo ON cargo.cargoPK = pagamento.cargoPK
JOIN funcionario ON funcionario.funcPK = pagamento.funcPK
```

```

GROUP BY ANO, ESCOLARIDADE, SEXO
ORDER BY ANO, ESCOLARIDADE, SEXO
d.
SELECT data.dataAno AS ANO,
cargo.cargoNivel AS ESCOLARIDADE,
funcionario.funcSexo AS SEXO,
ROUND(AVG(pagamento.salario),2) AS `Média dos Salários (R$)`
FROM pagamento JOIN data ON data.dataPK = pagamento.dataPK
JOIN cargo ON cargo.cargoPK = pagamento.cargoPK
JOIN funcionario ON funcionario.funcPK = pagamento.funcPK
GROUP BY ANO, ESCOLARIDADE, SEXO
ORDER BY ANO, ESCOLARIDADE, SEXO
e.
SELECT funcAnoNascimento AS ANO,
cargoNivel AS ESCOLARIDADE,
funcSexo AS SEXO,
ROUND(AVG(salario),2) AS `Média dos Salários (R$)`
FROM pagamento JOIN data ON data.dataPK = pagamento.dataPK
JOIN cargo ON cargo.cargoPK = pagamento.cargoPK
JOIN funcionario ON funcionario.funcPK = pagamento.funcPK
GROUP BY ANO, ESCOLARIDADE, SEXO
ORDER BY ANO, ESCOLARIDADE, SEXO

```

Feedback

Sua resposta está correta.

A resposta correta é:

```

SELECT dataAno AS ANO,
cargoEscolaridadeMinima AS ESCOLARIDADE,
funcSexo AS SEXO,
ROUND(AVG(salario),2) AS `Média dos Salários (R$)`
FROM pagamento JOIN data ON data.dataPK = pagamento.dataPK
JOIN cargo ON cargo.cargoPK = pagamento.cargoPK
JOIN funcionario ON funcionario.funcPK = pagamento.funcPK
GROUP BY ANO, ESCOLARIDADE, SEXO
ORDER BY ANO, ESCOLARIDADE, SEXO

```

Questão 7

Correto

Atingiu 1,00 de 1,00

Texto da questão

Considere a constelação de fatos da BI Solutions e a seguinte solicitação de consulta:

"Liste em ordem decrescente os totais de receitas que cada equipe (representada por equipeNome e filialNome) gerou no ano de 2018."

Assinale a alternativa que corresponde à consulta em pyspark.

a.

```

pagamenton
.join(data, on="dataPK")n
.join(equipe, on="equipePK")n
.where("dataAno = 2018")n
.select("equipeNome", "filialNome", "salario")n
.groupBy("equipeNome", "filialNome")n
.sum("salario")n
.orderBy(desc("sum(salario)"))

```

```

b.
negociacao
.join(equipe, on="equipePK")
.join(cliente, on="clientePK")
.select("equipeNome", "filialNome", "receita")
.groupBy("equipeNome", "filialNome")
.sum("receita")
.orderBy(desc("sum(receita)"))

C.
pagamento
.join(data, on="dataPK")
.join(equipe, on="equipePK")
.where("dataPK BETWEEN 367 AND 731")
.select("equipeNome", "filialNome", "salario")
.groupBy("equipeNome", "filialNome")
.sum("salario")
.orderBy(desc("sum(salario)"))

d.
negociacao
.join(data, on="dataPK")
.join(equipe, on="equipePK")
.where("dataAno = 2018")
.select("equipeNome", "filialNome", "receita")
.groupBy("equipeNome", "filialNome")
.sum("receita")
.orderBy(desc("sum(receita)"))

e.
negociacao
.join(data, on="dataPK")
.join(equipe, on="equipePK")
.where("dataPK BETWEEN 367 AND 731")
.select("equipeNome", "filialNome", "receita")
.groupBy("equipeNome", "filialNome")
.sum("receita")
.orderBy(desc("sum(receita)"))

```

Feedback

Sua resposta está correta.

A resposta correta é:

```

negociacao
.join(data, on="dataPK")
.join(equipe, on="equipePK")
.where("dataAno = 2018")
.select("equipeNome", "filialNome", "receita")
.groupBy("equipeNome", "filialNome")
.sum("receita")
.orderBy(desc("sum(receita)"))

```

Questão 8

Correto

Atingiu 1,00 de 1,00

Texto da questão

Considere a constelação de fatos da BI Solutions e a seguinte solicitação de consulta:

"Qual a quantidade de negociações por ano e pela cidade da equipe, considerando equipes localizadas na mesma cidade de seus clientes?" Devem ser exibidas as colunas

na ordem e com os nomes especificados a seguir: ANO, CIDADE e TOTALNEGOCIACOES. Ordene as linhas exibidas primeiro por ano em ordem ascendente e depois por cidade em ordem ascendente.

Considerando a resposta para a consulta analítica, assinale a alternativa correta:

- a.
Com exceção do ano de 2020, as negociações realizadas pelas equipes localizadas na cidade de SAO PAULO aumentaram paulatinamente.
- b.
As equipes localizadas na cidade do RIO DE JANEIRO possuem menos negociações do que as demais equipes em todos os anos.
- c.
Três cidades distintas possuem equipes que negociaram mais de uma vez em 2019.
- d.
Quatro cidades distintas possuem equipes que negociaram mais de uma vez em 2018.
- e.
As equipes localizadas na cidade de SAO PAULO possuem mais negociações do que as demais equipes em todos os anos.

Feedback

Sua resposta está correta.

A resposta correta é:

Com exceção do ano de 2020, as negociações realizadas pelas equipes localizadas na cidade de SAO PAULO aumentaram paulatinamente.

Questão 9

Correto

Atingiu 1,00 de 1,00

Texto da questão

Considere a constelação de fatos da BI Solutions e a seguinte solicitação de consulta:

"Liste todas as agregações que podem ser geradas a partir da média dos salários dos funcionários que moram na região SUDESTE por sexo e por ano." Arredonde a média dos salários para até duas casas decimais. Devem ser exibidas as colunas na ordem e com os nomes especificados a seguir: SEXO, ANO e MEDIASALARIO. Ordene as linhas exibidas primeiro por sexo, depois por ano, depois por média dos salários, todos em ordem ascendente.

Considerando a resposta para a consulta analítica, assinale a alternativa correta:

- a.
São retornadas 18 linhas, das quais 6 linhas são referentes ao sexo feminino e 6 linhas são referentes ao sexo masculino.
- b.
A quarta e a quinta linhas retornadas contêm o mesmo valor de média dos salários e referem-se a anos diferentes.

c.

São retornadas 10 linhas, das quais 5 linhas são referentes ao sexo feminino e 5 linhas são referentes ao sexo masculino.

d.

São retornadas 13 linhas, das quais 6 linhas são referentes ao sexo feminino e 6 linhas são referentes ao sexo masculino.

e.

As médias dos salários das funcionárias de sexo feminino para os anos de 2017 e 2018 são menores do que as médias dos salários dos funcionários do sexo masculino para os anos de 2017 e 2018, respectivamente.

Feedback

Sua resposta está correta.

A resposta correta é:

São retornadas 18 linhas, das quais 6 linhas são referentes ao sexo feminino e 6 linhas são referentes ao sexo masculino.

Questão 10

Correto

Atingiu 1,00 de 1,00

Texto da questão

Considere a constelação de fatos da BI Solutions e a seguinte solicitação de consulta:

"Qual o lucro ou prejuízo médio de cada equipe localizada na região SUDESTE do BRASIL?", sendo que lucro representa a diferença entre a média das receitas e a média dos salários. Arredonde o lucro ou prejuízo para até duas casas decimais. Devem ser exibidas as colunas na ordem e com os nomes especificados a seguir: NOME DA EQUIPE, NOME DA FILIAL, CIDADE DA FILIAL, LUCRO OU PREJUÍZO.

Considerando a resposta para a consulta analítica, assinale a alternativa correta:

a.

As equipes localizadas na cidade de RIO DE JANEIRO garantem mais lucro à BI Solutions do que as equipes localizadas na cidade de SAO PAULO.

b.

A soma dos lucros gerados pelas equipes que possuem APP - MOBILE em seu nome são maiores do que a soma dos lucros gerados pelas equipes que possuem APP - DESKTOP em seu nome, independentemente da filial e da cidade na qual estão localizadas.

c.

Equipes que possuem WEB em seu nome, independentemente da filial e da cidade na qual estão localizadas, proveem mais lucro à BI Solutions do que as demais equipes.

d.

Todas as equipes tiveram lucro, independentemente da cidade na qual elas estão localizadas.

e.

Os lucros gerados pelas equipes que possuem BI & ANALYTICS em seu nome, independentemente da filial e da cidade na qual estão localizadas, são maiores do que a soma dos lucros gerados pelas demais equipes.

Feedback

Sua resposta está correta.

A resposta correta é:

Os lucros gerados pelas equipes que possuem BI & ANALYTICS em seu nome, independentemente da filial e da cidade na qual estão localizadas, são maiores do que a soma dos lucros gerados pelas demais equipes.