

BUSSINESS INTELIGENCE

Prof. Orlando Oliveira dos
Santos

APRESENTAÇÃO

Disciplina

- Business Intelligence para Gestores Públicos - Teoria e Prática

Carga horária:

- 40 horas

Objetivo da disciplina

- Capacitar servidores para atuar na concepção, gerenciamento, produção, manutenção e utilização de soluções para apoio a tomada de decisão.

Professor

- Orlando Oliveira dos Santos (MSC)
- Orlando.santos@presidencia.gov.br - (61) 3411 1919 / 99646 6151

Formação Acadêmica

- Mestrado em Computação Aplicada - UnB (2014 – 2016)
- MBA, Administração Estratégica de Sistemas de Informação – FGV (2011 – 2013)
- Tecnólogo em Processamento de Dados, Análise e Desenvolvimento de Sistemas – FAETEC/SP (2000-2002)

APRESENTAÇÃO

Certificações

- PMP - Project Manager Professional (Project Management Institute – PMI);
- PSM-I Professional Scrum Master I (Scrum.org)
- ITIL V3 Foundation Certified Professional (EXIN Institute)
- ISO/IEC 20000 Foundation Certified Professional (EXIN Institute);
- ISO/IEC 27002 Foundation Certified Professional (EXIN Institute);
- COBIT Foundation Certified Professional (Information Systems Audit and Control Association - ISACA).

Experiência Profissional

- Diretor de Gestão da Informação – PR/CCIVIL/SE
- Coordenador Geral de Gestão da Informação – MP/SEGES/GABIN (2017)
- Chefe de Equipe de Análise de dados e Informações – MP/SEGES/DETRV (2016)
- Chefe de Divisão de Informações Estratégicas MP/SE/DTI (2015)
- Gerenciamento de projetos de desenvolvimento e implantação de sistemas para autarquias de fiscalização profissional (2010 – 2014)
- Gerenciamento de projetos de desenvolvimento e implantação de sistemas tributários para prefeituras (2004-2010)

Currículo Lattes: <http://lattes.cnpq.br/5082932373540854>

CONTEÚDO PROGRAMÁTICO

Teórico

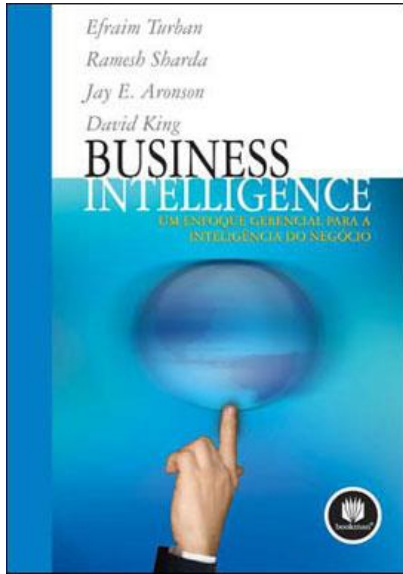
- Informação para tomada de Decisão
- Introdução ao Business Intelligence – BI
- Natureza dos dados e Taxonomia dos dados;
- Pré-processamento de dados
- Data Warehouse - DW e Data Mart
- Modelagem estatística:
- Visualização de dados
- Estratégia de seleção de tipo de visualização
- Linguagem SQL

CONTEÚDO PROGRAMÁTICO

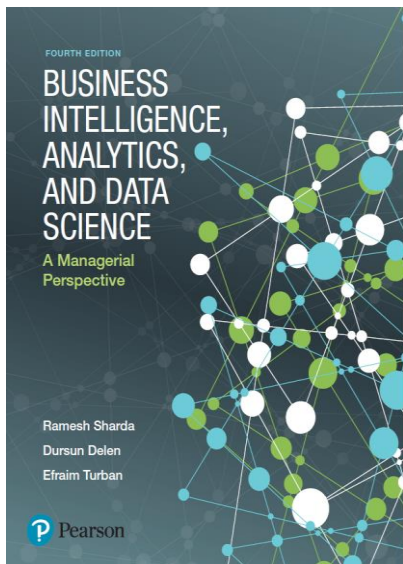
Prático

- Apresentação da ferramenta Qlik Sense Desktop
- Apresentação do conjunto de dados a ser utilizado
- Carga e pré-processamento dos dados;
- Realização de operações de manipulação de dados utilizando linguagem de Script Qlik Sense.
- Definição de metadados (métricas e dimensões); e
- Criação de visualizações de dados utilizando a ferramenta de Qlik Sense Desktop.
- Realização de operações de manipulação de dados em bancos de dados utilizando linguagem SQL; e
- Atividade em duplas de desenvolvimento de um painel de BI com dados escolhidos pela dupla.

BIBLIOGRAFIA



(TURBAN ,2009): Turban, Efraim; et al. Business Intelligence: Um Enfoque Gerencial Para a Inteligência do Negócio. São Paulo: Bookman, 2009



(SHARDA,DELEN,TURBAN ,2017): Sharda, Ramesh; Delen, Dursun; Turban, Efraim. Business Intelligence, Analytics, and Data Science: A Managerial Perspective. Fourth Edition – Pearson Education - 2016

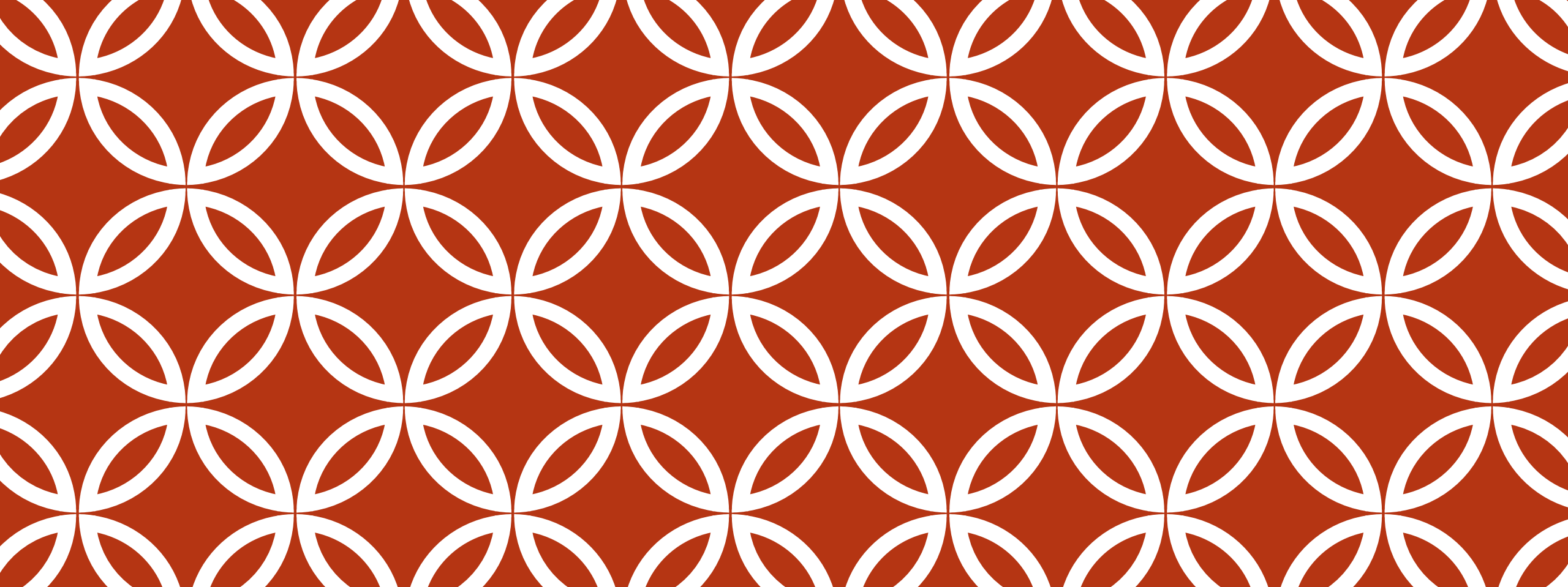
CONTEÚDO PROGRAMÁTICO

Tutoriais Qlik Sense

- Disponíveis em:
 - <http://help.qlik.com/pt-BR/sense/1.1/Content/TutorialsExamples.htm>
- Tutorial - Começando com o básico
 - Aprenda as noções básicas do Qlik Sense, como fazer seleções e como interpretar os resultados.
- Tutorial - Criação de um aplicativo
 - Aprenda a criar um aplicativo desde o início.
- Tutorial - Scripts para iniciantes
 - Comece com o básico do script para poder carregar e manipular dados no Qlik Sense.
- Tutorial - Próximas etapas do script
 - Avance para o uso de scripts mais avançado no Qlik Sense.



Sense™



BUSSINESS INTELIGENCE — BI E QLIKSENSE

AULA 01

INFORMAÇÃO E TOMADA DE DECISÃO



**“In God we trust.
All others must bring data.”**

- Dr. W. Edwards Deming

INFORMAÇÃO E TOMADA DE DECISÃO

Dados

- São elementos estruturados, provenientes de uma coleta ou pesquisa. Podem ser classificados em termos de fatos, sinais, ou símbolos e identificados como palavras, números, códigos, tabelas ou base de dados;

Informação

- A informação surge a partir da estruturação ou organização de dados processados para um fim/contexto específico. Permite identificar o “o que”. Se apresenta no formato de sentenças, equações, conceitos e ideias;

INFORMAÇÃO E TOMADA DE DECISÃO

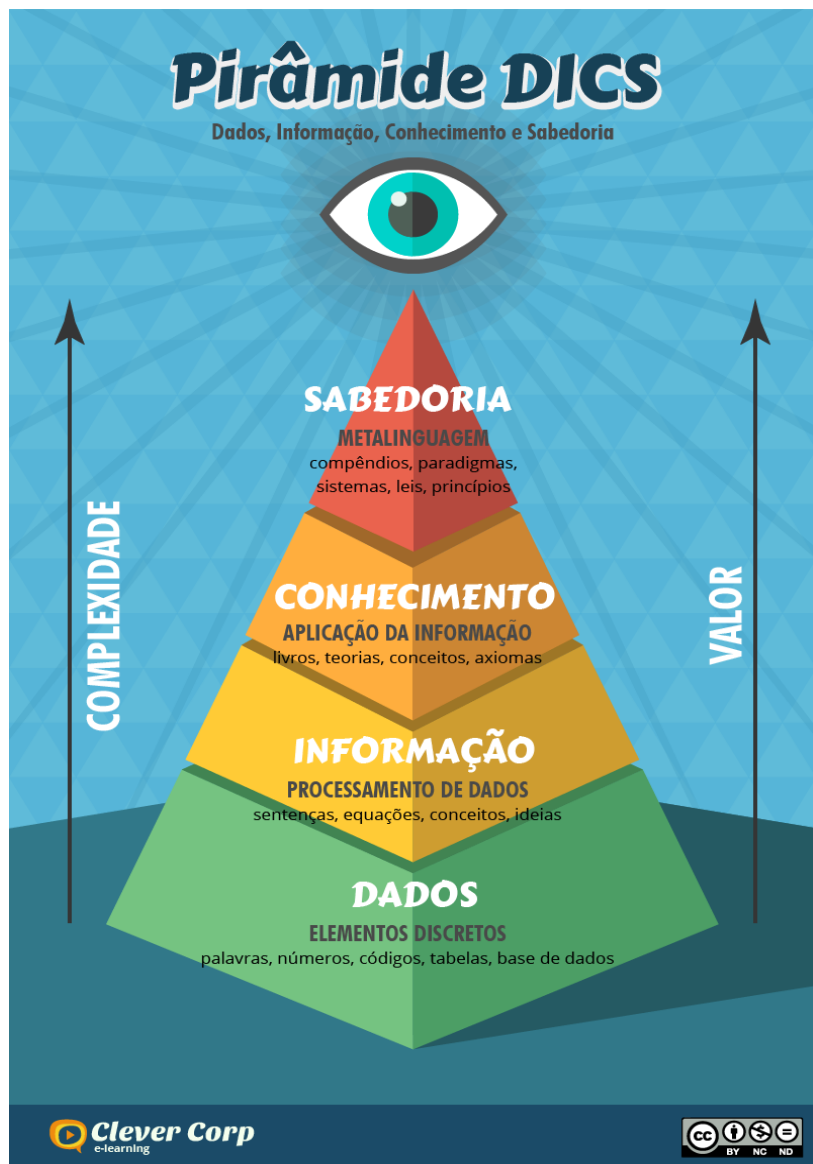
Conhecimento

- É composto por uma mescla de informação contextualizada, valores experiências e regras. Permite identificar o “como”. Estão na forma de livros, teorias, conceitos e axiomas;

Sabedoria

- É o estágio mais complexo de se definir e ocorre quando há a resignificação dos outros níveis em combinações metalinguísticas. Permite identificar o “por que”. Expressa-se em compêndios, paradigmas, sistemas, leis e princípios.

INFORMAÇÃO E TOMADA DE DECISÃO



Dados, Informação e Conhecimento

	Dado	Informação	Conhecimento
Definição	Simple observação sobre o estado do mundo	Dado dotado de relevância e propósito	Informação valiosa da mente humana, incluindo reflexão, síntese e contexto
Características	<ul style="list-style-type: none">• Facilmente estruturado• Facilmente obtido por máquinas• Frequentemente quantificado• Facilmente transferível	<ul style="list-style-type: none">• Requer unidade de análise• Exige consenso em relação ao significado• Exige necessariamente a mediação humana	<ul style="list-style-type: none">• Difícil estruturação• Difícil captura em máquinas• Frequentemente tácito• Difícil transferência

Davenport (1998)

Pirâmide DICS-

Fonte: http://www.clevercorp.com.br/wp-content/uploads/2015/07/piramide_03-01.png

SISTEMAS DE INFORMAÇÃO

Sistema de Informação (SI):

- Sistema cujo elemento principal é a informação. Seu objetivo é armazenar, tratar e fornecer informações de tal modo a apoiar as funções ou processos de uma organização.
- Um SI é composto de um subsistema social e de um subsistema automatizado.
- O primeiro inclui as pessoas, processos, informações e documentos.
- O segundo consiste dos meios automatizados (máquinas, computadores, redes de comunicação) que interligam os elementos do subsistema social

<http://www.ifba.edu.br/professores/pablovf/repositorio/siTiposSi.pdf>

SISTEMAS DE INFORMAÇÃO DO GOVERNO FEDERAL

Siorg - Sistema de Informações Organizacionais do Governo Federal

- O Siorg é o sistema que trata as informações sobre as estruturas, competências, finalidades, jurisdição, histórico legal e titulares dos Órgãos da Administração Federal. Disponibiliza o catálogo de órgãos que é utilizado por todos os sistemas estruturantes do governo federal.

Siape - Sistema Integrado de Administração de Recursos Humanos

- O Sistema Integrado de Administração de Recursos Humanos (Siape) é um sistema de abrangência nacional criado com a missão de integrar todas as plataformas de gestão da folha de pessoal dos servidores públicos. Hoje, o Siape é um dos principais sistemas estruturadores do governo e é responsável pela produção das folhas de pagamento dos mais de 200 órgãos federais.

Fonte: <https://intra.serpro.gov.br/linhas-negocio/catalogo-de-solucoes>

SISTEMAS DE INFORMAÇÃO DO GOVERNO FEDERAL

Sigepe - Sistema de Gestão de Pessoas do Governo Federal

- O Sistema de Gestão de Pessoas do Governo Federal (Sigepe), desenvolvido em parceria entre o Serpro e a Dataprev, foi criado para subsidiar, por meio dos recursos de Tecnologia da Informação, as políticas de gestão de pessoas e as operações de recursos humanos do Governo Federal.

SCDP - Sistema de Concessão de Diárias e Passagens

- Com o objetivo de automatizar o fluxo do processo de concessão de diárias e passagens a servidores da Administração Pública federal, foi criado o Sistema de Concessão de Diárias e Passagens (SCDP). Implantado em 2004, com uso tornado obrigatório em 2009 (com fundamento legal no Decreto n.º 6.258, de novembro de 2007).

Fonte: <https://intra.serpro.gov.br/linhas-negocio/catalogo-de-solucoes>

SISTEMAS DE INFORMAÇÃO DO GOVERNO FEDERAL

Siest - Sistema de Informações das Empresas Estatais

- O Siest é o sistema que oferece suporte ao Departamento de Coordenação e Controle das Empresas Estatais (Dest), do Ministério da Economia, na captação de propostas de investimentos das empresas estatais para o exercício financeiro subsequente.

Siasg - Sistema Integrado de Administração de Serviços Gerais:

- O Siasg é o sistema utilizado para facilitar e agilizar os processos de compra e aquisição de materiais e serviços do Governo Federal. Sua finalidade é integrar os órgãos da Administração Pública Federal direta, autárquica e fundacional.

Fonte: <https://intra.serpro.gov.br/linhas-negocio/catalogo-de-solucoes>

SISTEMAS DE INFORMAÇÃO DO GOVERNO FEDERAL

Portal de Compras do Governo Federal (Comprasnet)

- O Comprasnet é um sistema web destinado à realização de licitações, contratações e aquisições promovidas pelas instituições do governo federal. A tecnologia disponibiliza uma série de facilidades voltadas para o controle e execução dos processos de contratação, além de disponibilizar a legislação que regulamenta o processo de serviços gerais e de contratação com informações gerais que torna desburocratizada a participação em processos licitatórios.

CPF - Cadastro de Pessoas Físicas

- O Cadastro de Pessoas Físicas (CPF) é um banco de dados gerenciado pela Secretaria da Receita Federal do Brasil (RFB) que armazena informações de contribuintes, brasileiros e estrangeiros, obrigados pela legislação vigente à inscrição no CPF, ou de cidadãos que se inscreveram voluntariamente

Fonte: <https://intra.serpro.gov.br/linhas-negocio/catalogo-de-solucoes>

SOLUÇÕES DE APOIO A TOMADA DE DECISÃO - NECESSIDADE

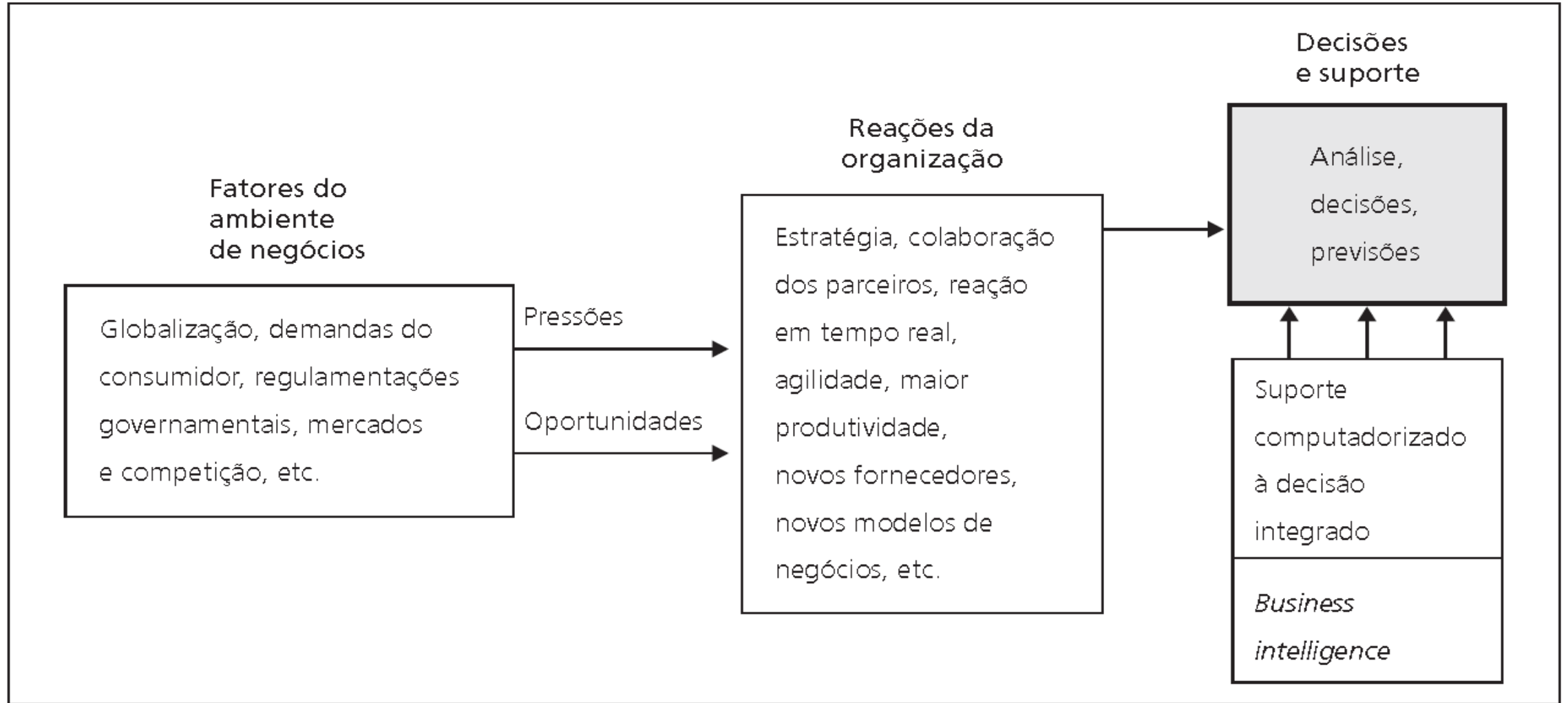


Figura 1.1 Modelo de pressões – reações – suporte de negócios - Fonte: (TURBAN ,2009)

SOLUÇÕES DE APOIO À TOMADA DE DECISÃO - MOTIVADORES

Aumento da disponibilidade de hardware, software e recursos de rede

Intensificação comunicação e colaboração em grupo

Evolução das soluções de gerenciamento de dados

Aumento da capacidade de gerenciamento de grandes repositórios de dados e Big Data

Disponibilidade softwares para suporte analítico

Superação dos limites cognitivos no processamento e armazenamento de informações

Melhoria na gestão do conhecimento

Disponibilidade de suporte a qualquer momento em qualquer lugar

SOLUÇÕES DE APOIO A TOMADA DE DECISÃO - EVOLUÇÃO

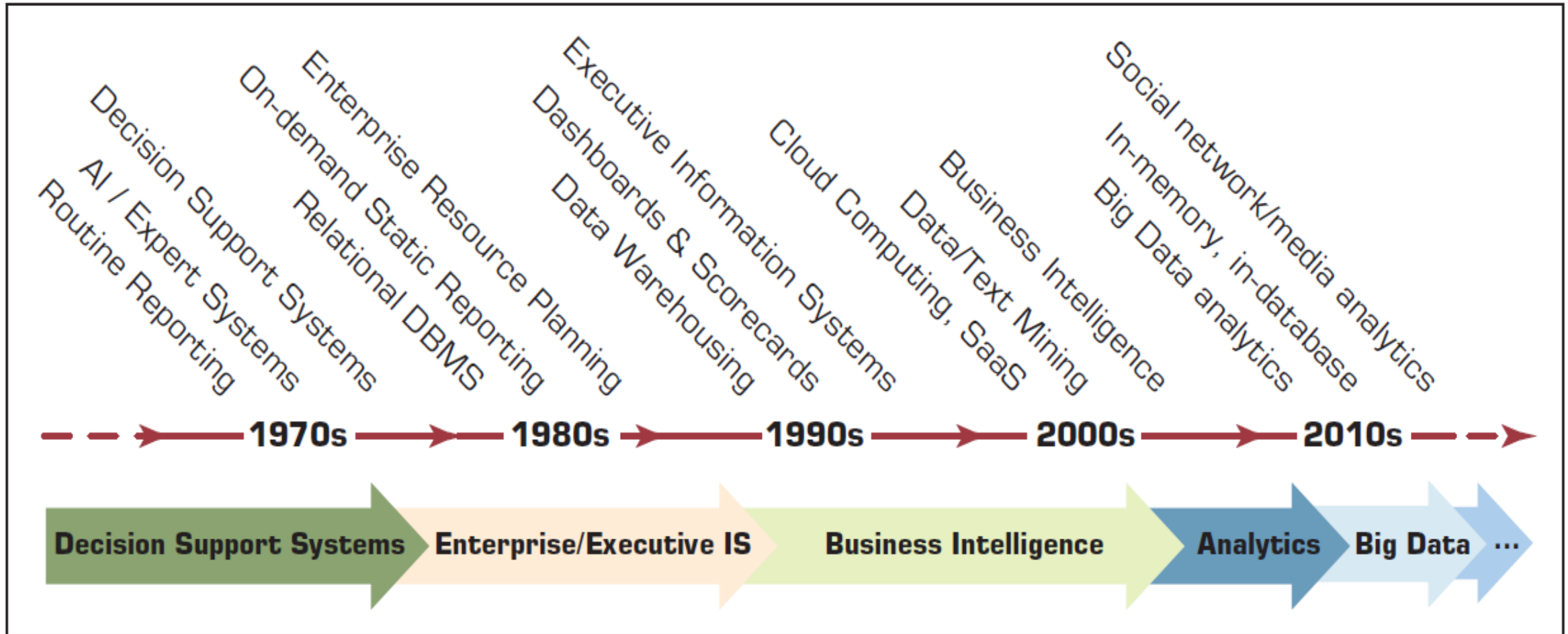


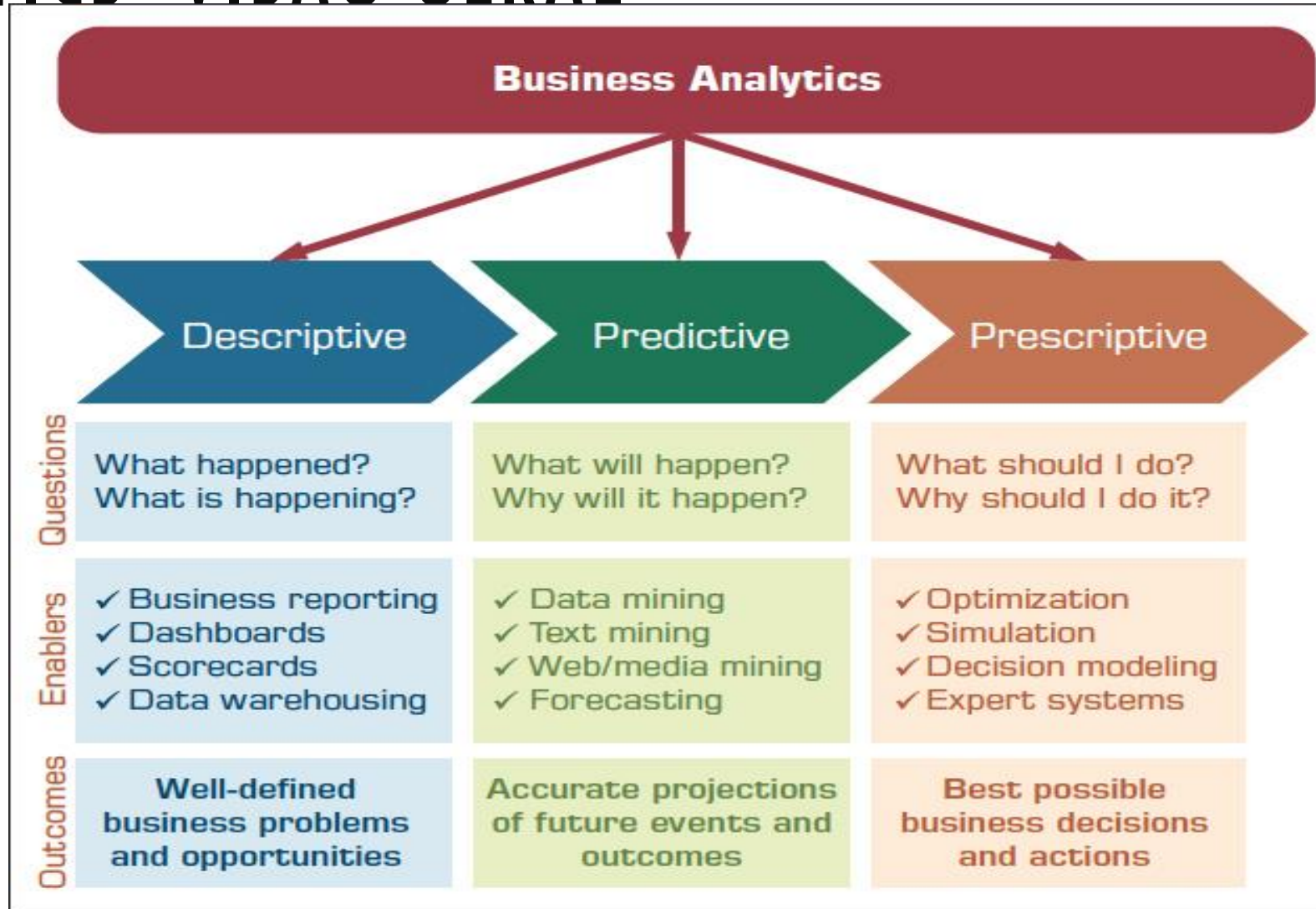
Figura 1.1 Modelo de pressões – reações – suporte de negócios - Fonte: (TURBAN ,2009)

ANALYTICS- VISÃO GERAL

Benefícios

- Um termo relativamente novo / palavra-chave;
- Processo de desenvolvimento de decisões ou recomendações para ações baseadas em insights gerados a partir de dados históricos;
- De acordo com Institute for Operations Research and Management Science (INFORMS) :
 - Analytics representa a combinação de tecnologia informática, técnicas de gestão e estatística para resolver problemas reais.

ANALYTICS- VISÃO GERAL



Copyright © 2018, 2014, 2011, 2008 by Pearson Education, Inc.

Figura 1.11 Three Types of Analytics - Fonte: (SHARDA,DELEN,TURBAN ,2017)

ANALYTICS- ANÁLISE DESCRITIVA

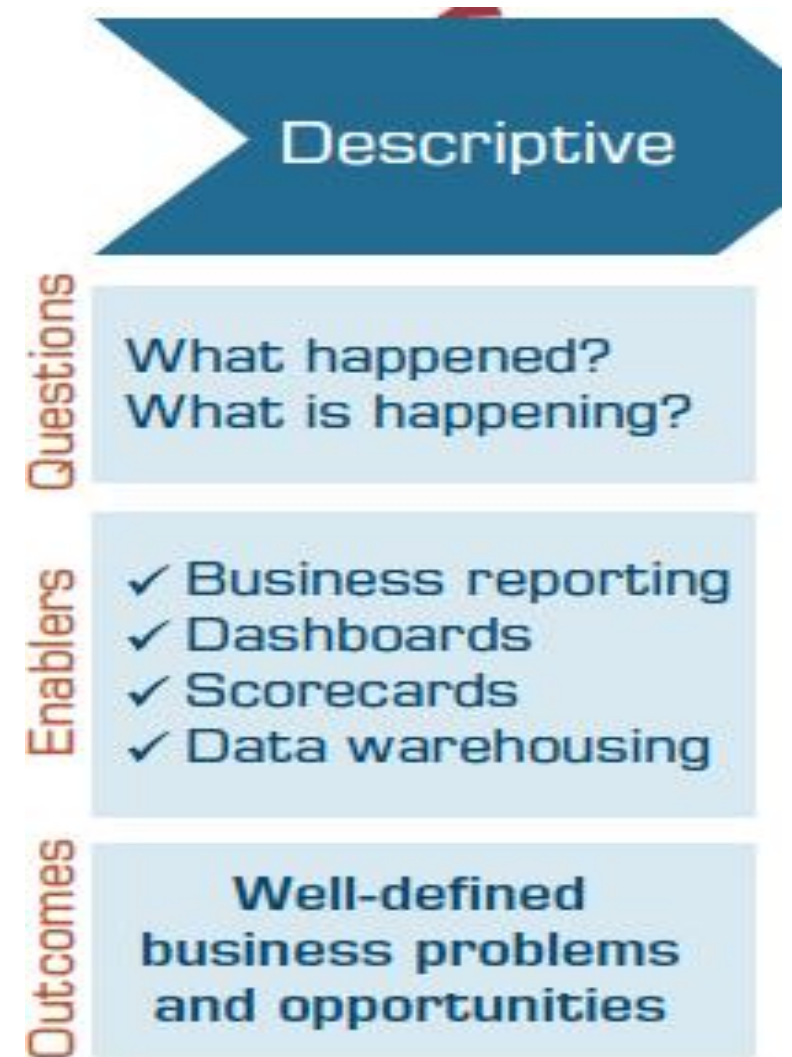
Análise descritiva ou de relatórios

Responde à pergunta sobre o que aconteceu

Análise retrospectiva de dados históricos

Habilitadores:

- OLAP / DW
- Visualização de dados
 - Painéis e Scorecards
- Estatísticas descritivas



ANALYTICS- ANÁLISE PREDITIVA

Pretende determinar o que provavelmente acontecerá no futuro (prevendo os eventos futuros)

Olhando os dados passados para prever o futuro

Habilitadores

- Mineração de dados
- Mineração de texto / Web mining
- Previsão (séries temporais)



What will happen?
Why will it happen?

- ✓ Data mining
- ✓ Text mining
- ✓ Web/media mining
- ✓ Forecasting

Accurate projections
of future events and
outcomes

ANALYTICS- ANÁLISE PRESCRITIVA

Pretende determinar a melhor decisão possível

Usa métodos descritivos e preditivos para criar as alternativas, e depois indica a melhor

Habilitadores

- Otimização
- Simulação
- Modelagem de decisão multi-critérios
- Sistemas inteligentes

Análise aplicada a muitos domínios



ANALYTICS OU DATA SCIENCE?

Duas disciplinas separadas, mas interconectadas

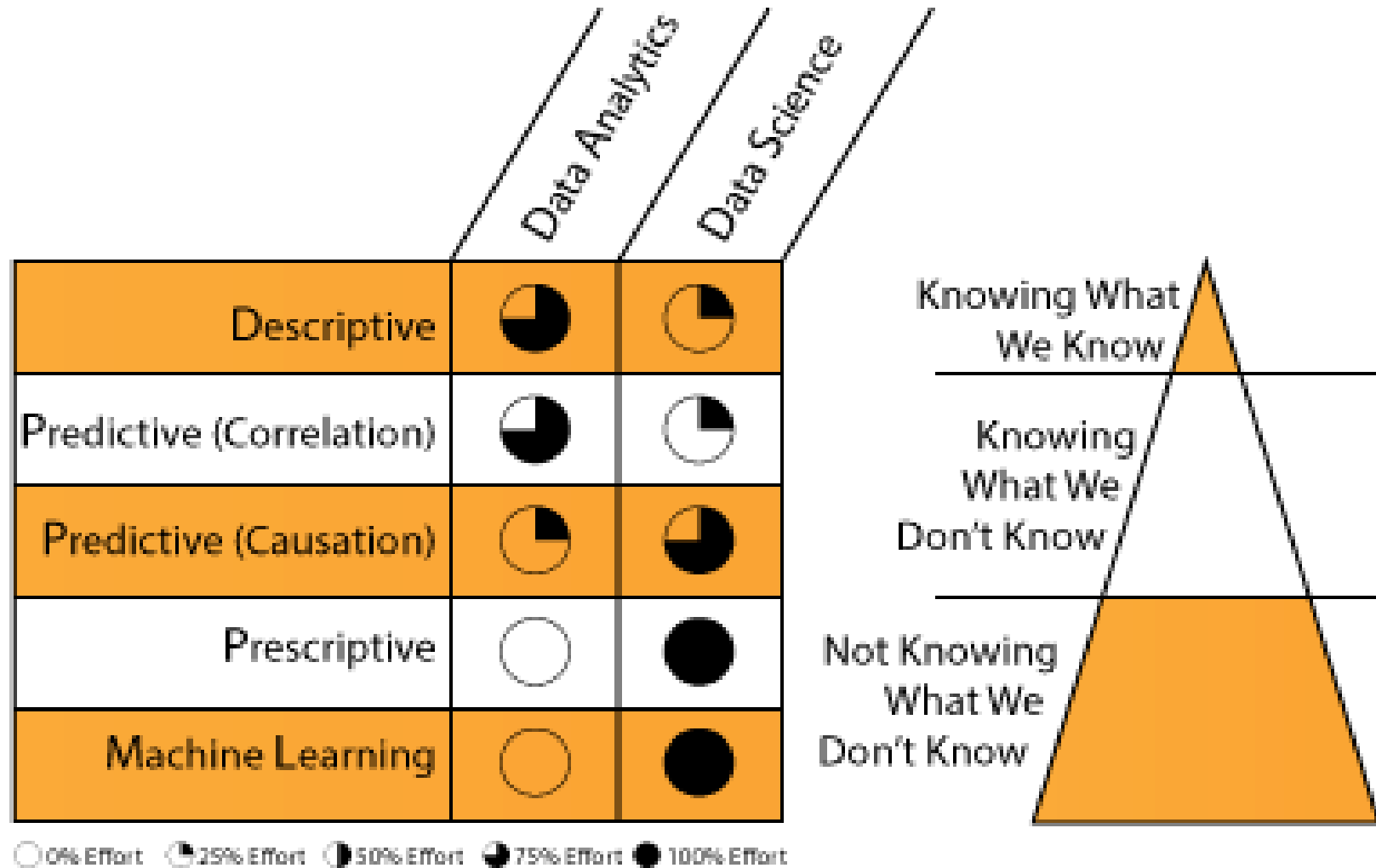
Analytics:

- Procura fornecer observações operacionais em questões que sabemos ou sabemos que não sabemos.

Data Science:

- Procura fornecer insights estratégicos em questões onde não sabemos o que não sabemos.

ANALYTICS OU DATA SCIENCE?



ANALYTICS - CIENTISTA DE DADOS

Cientista de dados:

- De acordo Gartner, o papel do cientista de dados é fundamental para as organizações que buscam extrair informações dos ativos de informação e requer uma ampla combinação de habilidades que podem ser cumpridas melhor como equipe".

Um bom cientista de dados deve ter os seguintes traços:

- Curiosidade – Explora os dados para encontrar as respostas
- Intuição - Um bom senso de negócios.
- Habilidade para coleta de dados - Como encontrar os dados relevantes
- Conhecimento de estatística - Compreender as correlações
- Habilidade para Modelagem analítica – Avaliar dados históricos para prever o futuro.
- Boa comunicação - Capacidade de compreender e expressar os resultados obtidos com as análises.

BUSINESS INTELLIGENCE

- O BI era tudo que se tinha relacionado ao uso de dados para suporte à tomada de decisão gerencial
- Agora, é parte da Business Analytics
 - BI = Análise descritiva

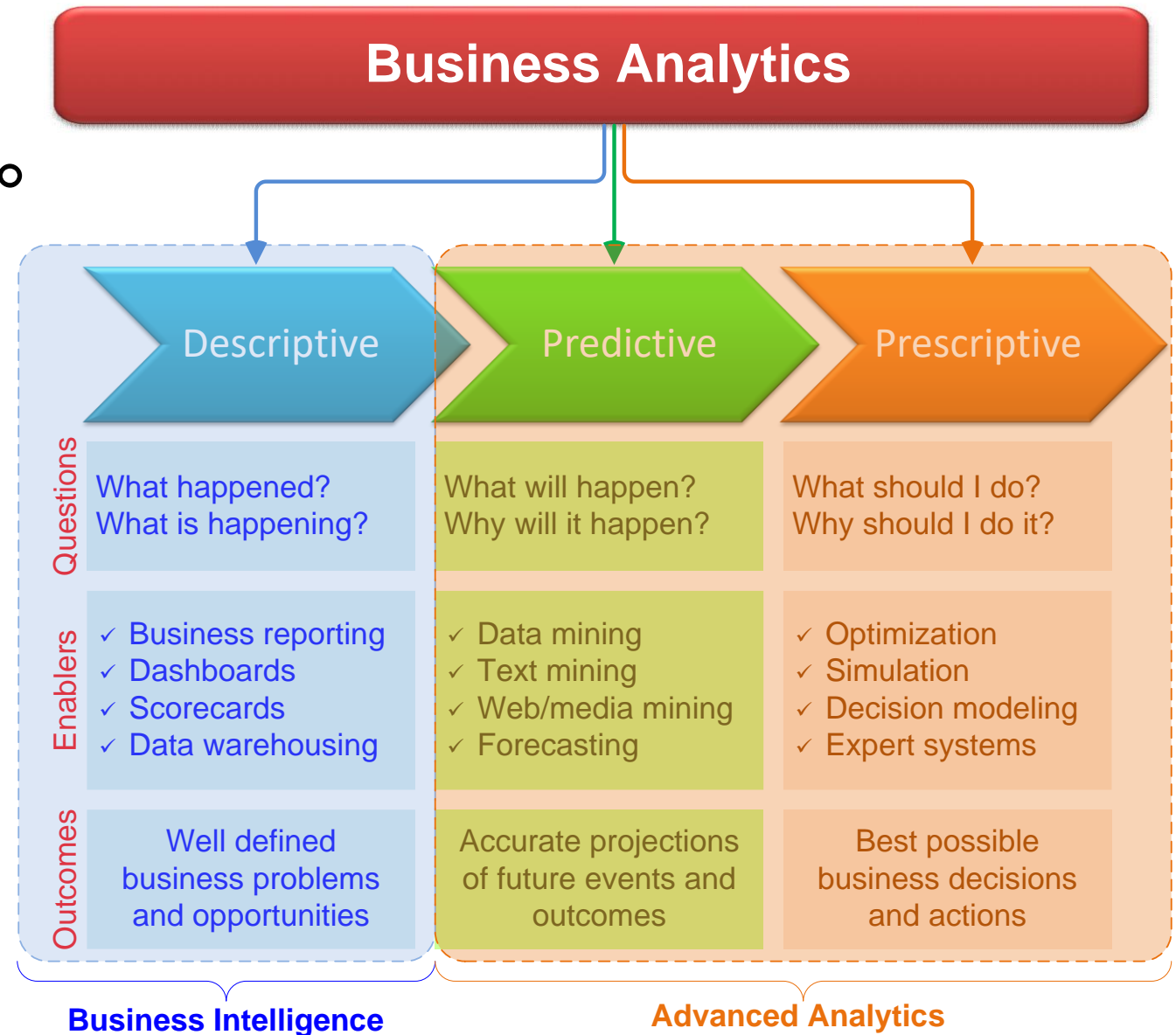


Figura 1.11 Three Types of Analytics - Fonte: (SHARDA,DELEN,TURBAN ,2017)

INTELIGÊNCIA DE NEGÓCIOS - DEFINIÇÃO

[Definição ampla] Um termo guarda-chuva que combina arquiteturas, ferramentas, bancos de dados, ferramentas analíticas, aplicações e metodologias;

[Definição restrita] Ferramentas e técnicas de análise descritiva ;

INTELIGÊNCIA DE NEGÓCIOS - EVOLUÇÃO

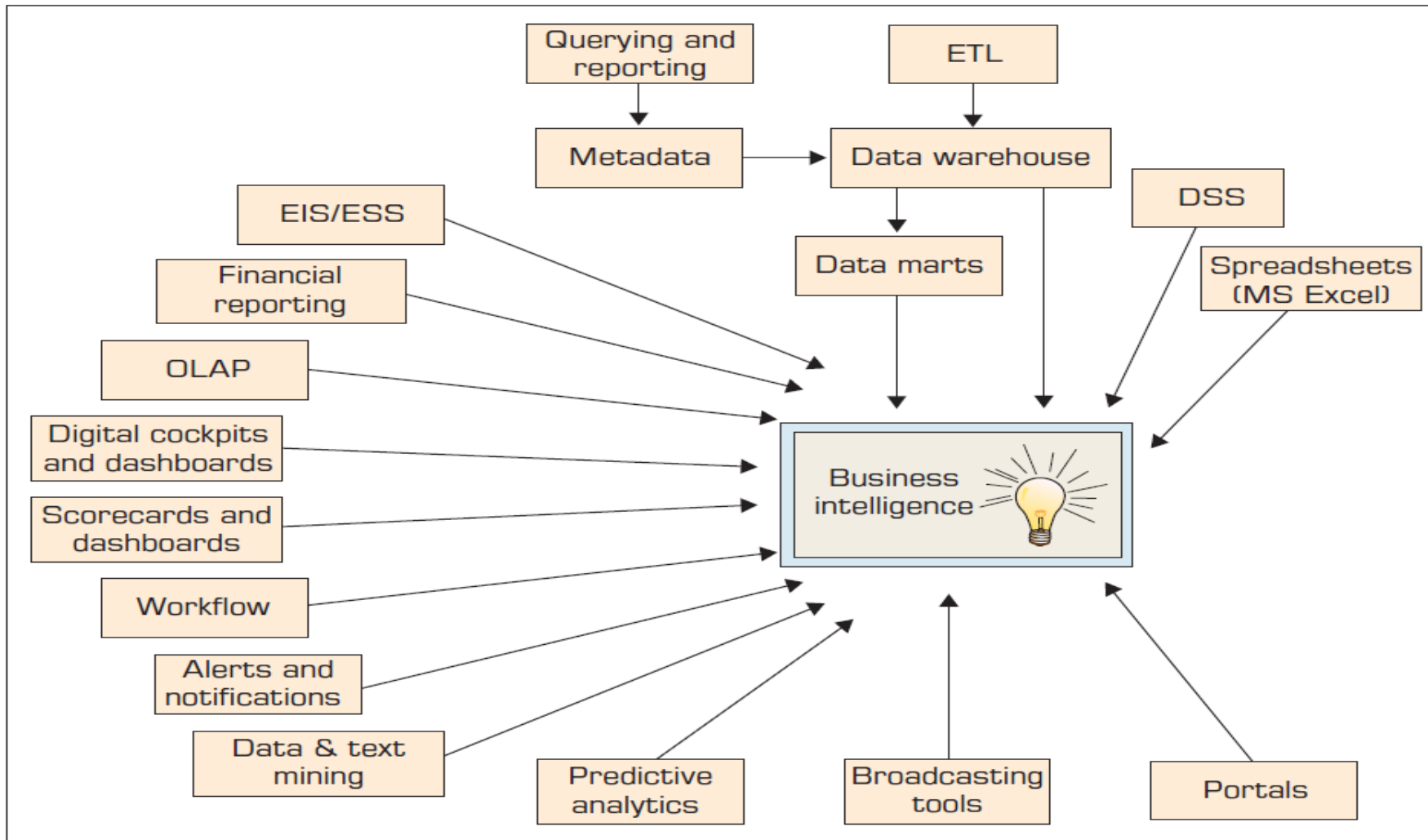


Figura 1.9 Evolution of Business Intelligence (BI) - Fonte: (SHARDA,DELEN,TURBAN ,2017)

INTELIGÊNCIA DE NEGÓCIOS - COMPONENTES

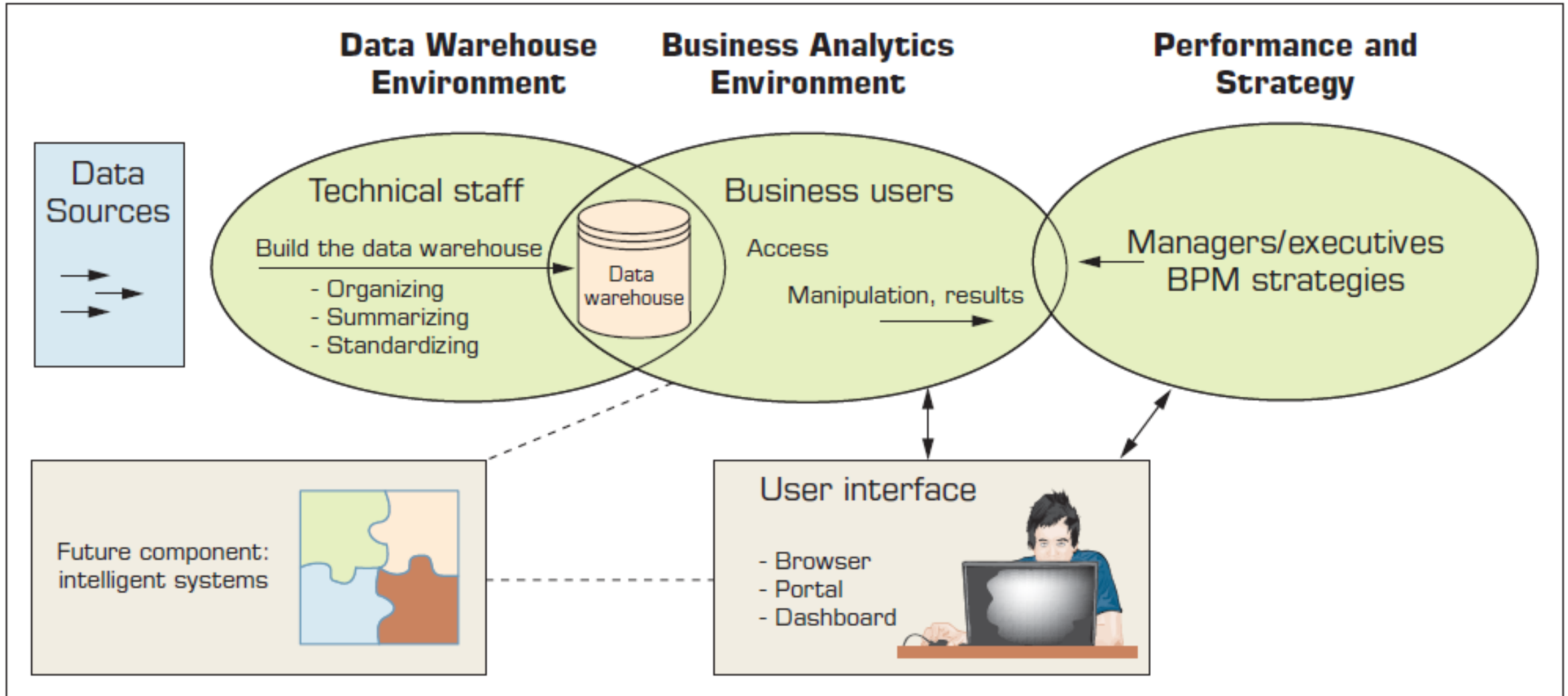


Figura 1.10 A High-Level Architecture of BI - Fonte: (SHARDA,DELEN,TURBAN ,2017)

INTELIGÊNCIA DE NEGÓCIOS – PROCESSO DE CRIAÇÃO

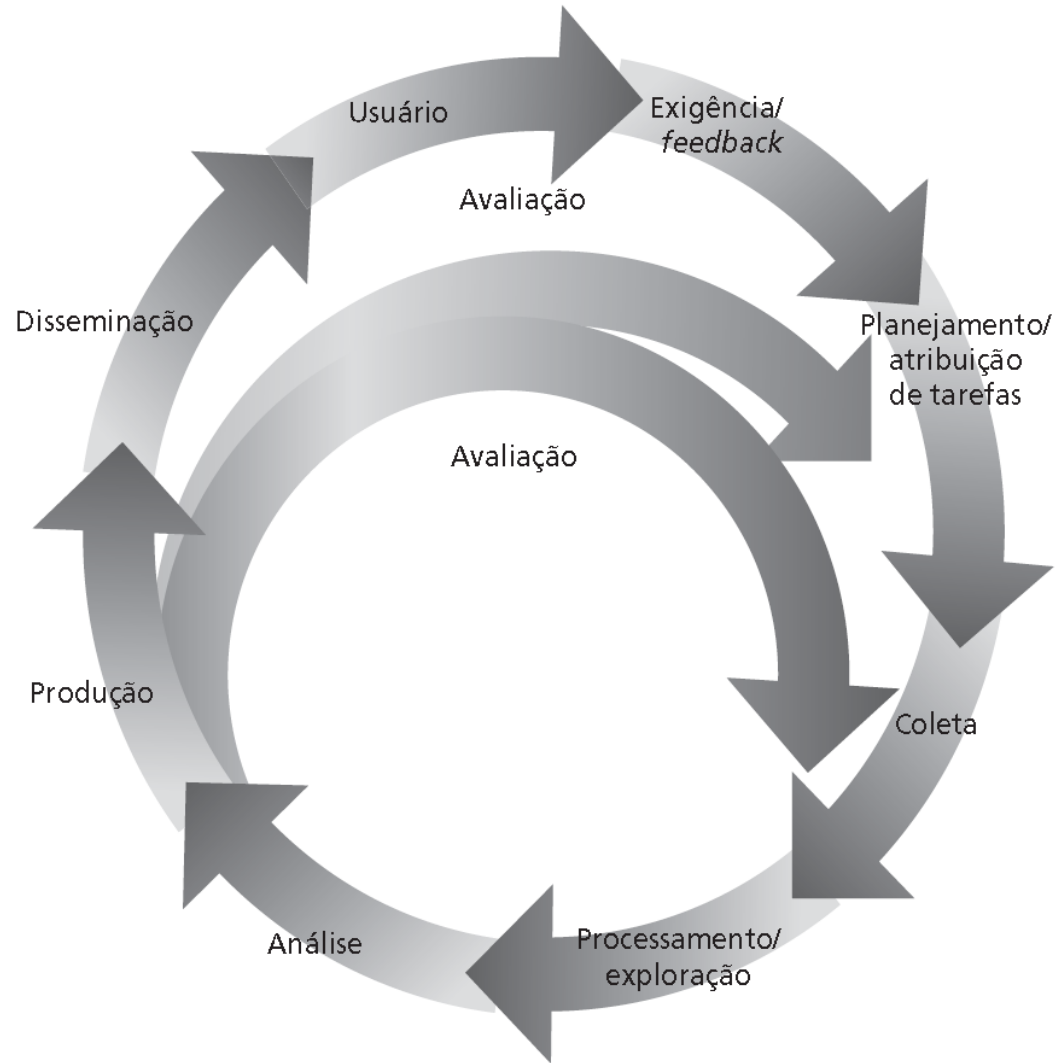


Figura 1.4 Processo de criação e uso da inteligência.- Fonte: Fonte: (TURBAN ,2009)

INTELIGÊNCIA DE NEGÓCIOS - ADOÇÃO

Planejamento e alinhamento com a estratégia de negócios

- Definir como o BI está ligado à definição e execução da estratégia
- Planejamento e execução de soluções de BI com foco nas necessidades estratégicas
- Escolher áreas chaves a serem atendidas com competências em BI
- Incentivar a interação entre as potenciais comunidades de usuários;
- Utilizar um centro de competências em BI para servir como um repositório e disseminador de melhores práticas de BI entre as diferentes área da organização.

INTELIGÊNCIA DE NEGÓCIOS - ADOÇÃO

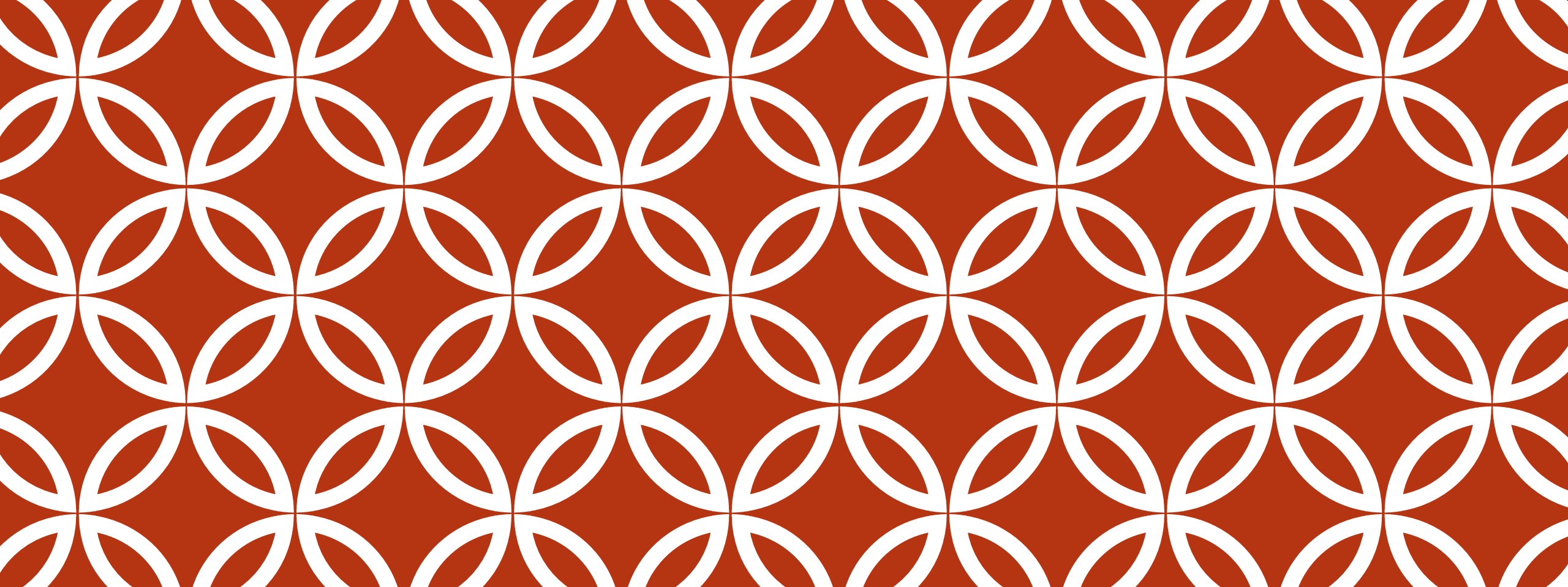
Aspectos Críticos a Serem Considerados

- Desenvolver ou adquirir soluções de BI
 - Fazer versus comprar
- Justificar e analisar o custo-benefício
 - Um esforço desafiador, por quê?
- Garantir a Segurança
- Proteção de Privacidade
- Integração em outros sistemas e aplicativos

INTELIGÊNCIA DE NEGÓCIOS - ADOÇÃO

Benefícios

- Economia de tempo
- Versão única da verdade
- Melhoria em estratégias e planos
- Processos mais eficientes
- Redução de custos
- Relatórios mais rápidos e precisos
- Melhoria da tomada de decisões
- Melhor serviço ao cidadão
- Aumento de receitas



BUSSINESS INTELIGENCE — BI E QLIKSENSE

AULA 02

AULA PRÁTICA

APRESENTAÇÃO DAS FERRAMENTAS

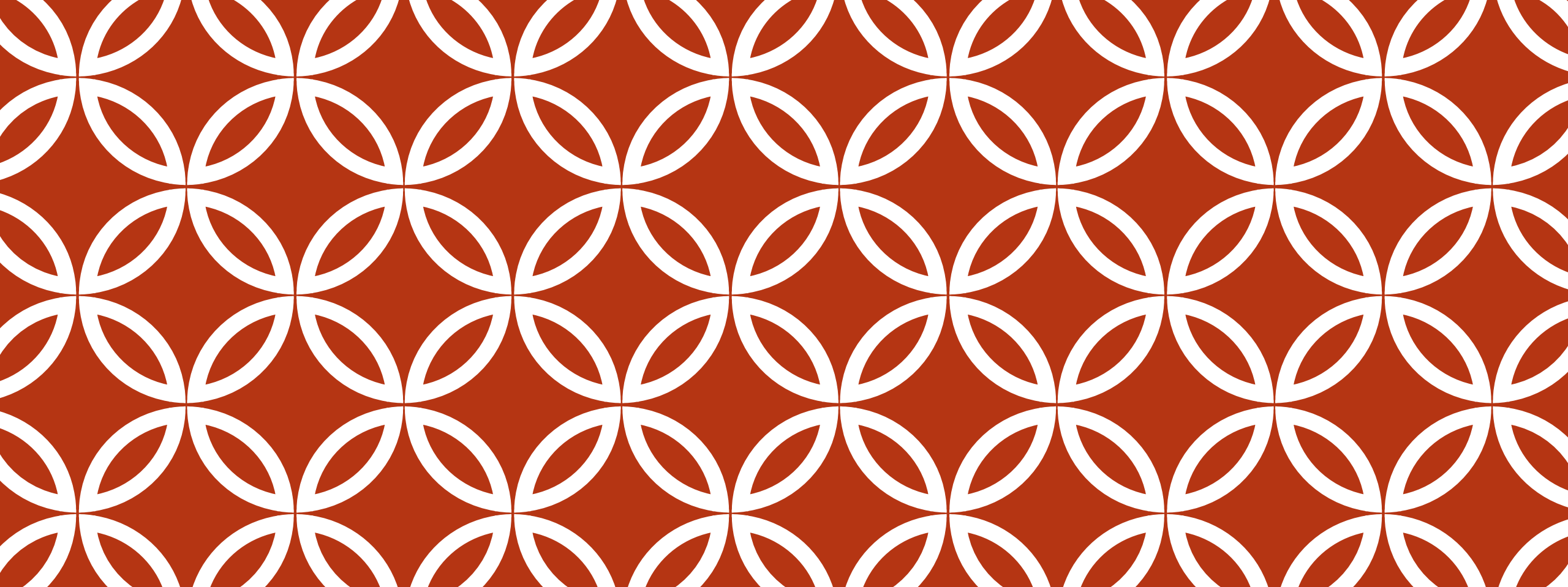
Apresentação do Software Qlik Sense Desktop (Uso gratuito);

Arquivo de dados:

<https://drive.google.com/file/d/1QK1-M0ubxJzko4mviz0EMhqAUZwSP4M/view?usp=sharing>

Arquivo do painel

<https://drive.google.com/file/d/1INzVUAIOQB0-6yp2bijAEkfcrrvaMnKgl/view?usp=sharing>



BUSSINESS INTELIGENCE — BI E QLIKSENSE

AULA 03

INTELIGÊNCIA DE NEGÓCIOS - COMPONENTES

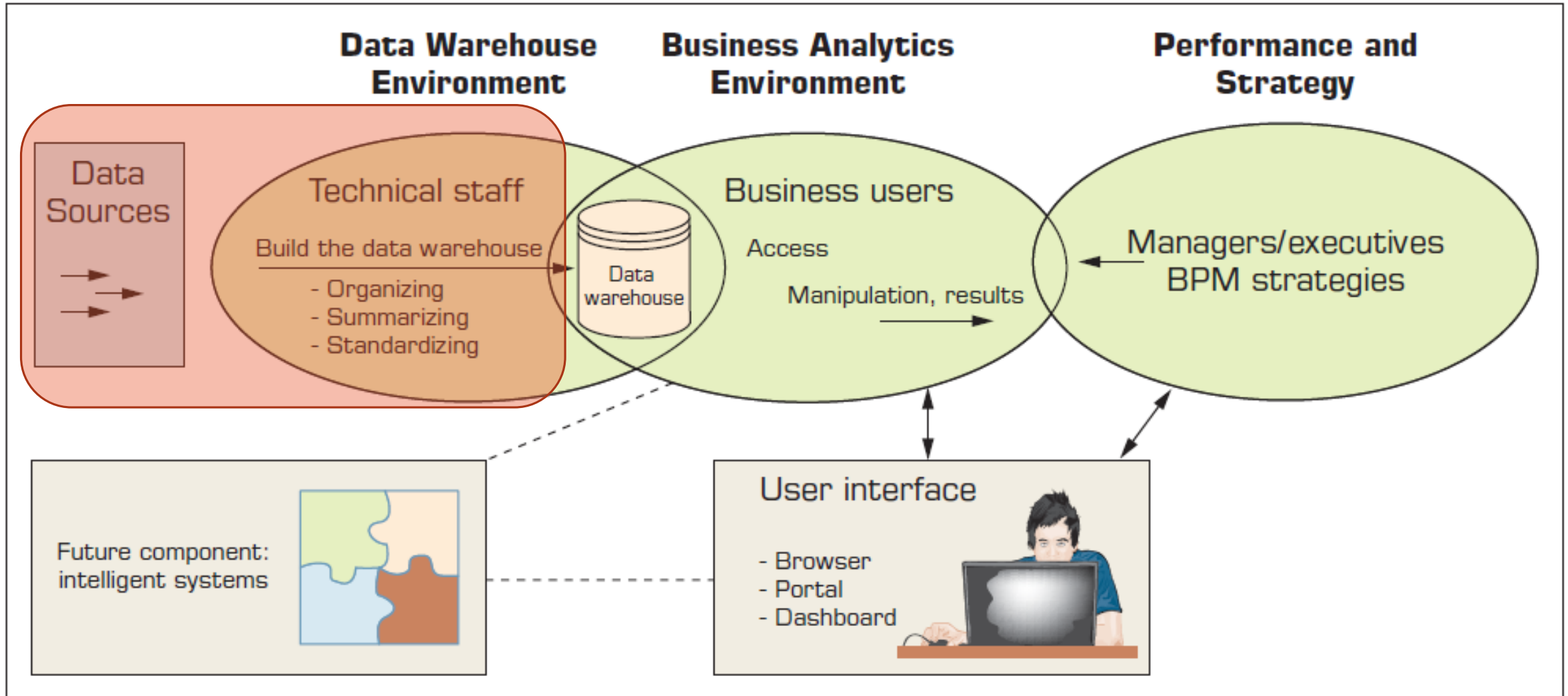


Figura 1.10 A High-Level Architecture of BI - Fonte: (SHARDA,DELEN,TURBAN ,2017)

DADOS - NATUREZA

Dados: uma coleção de fatos

Geralmente obtido como resultado de experiências ou observações;

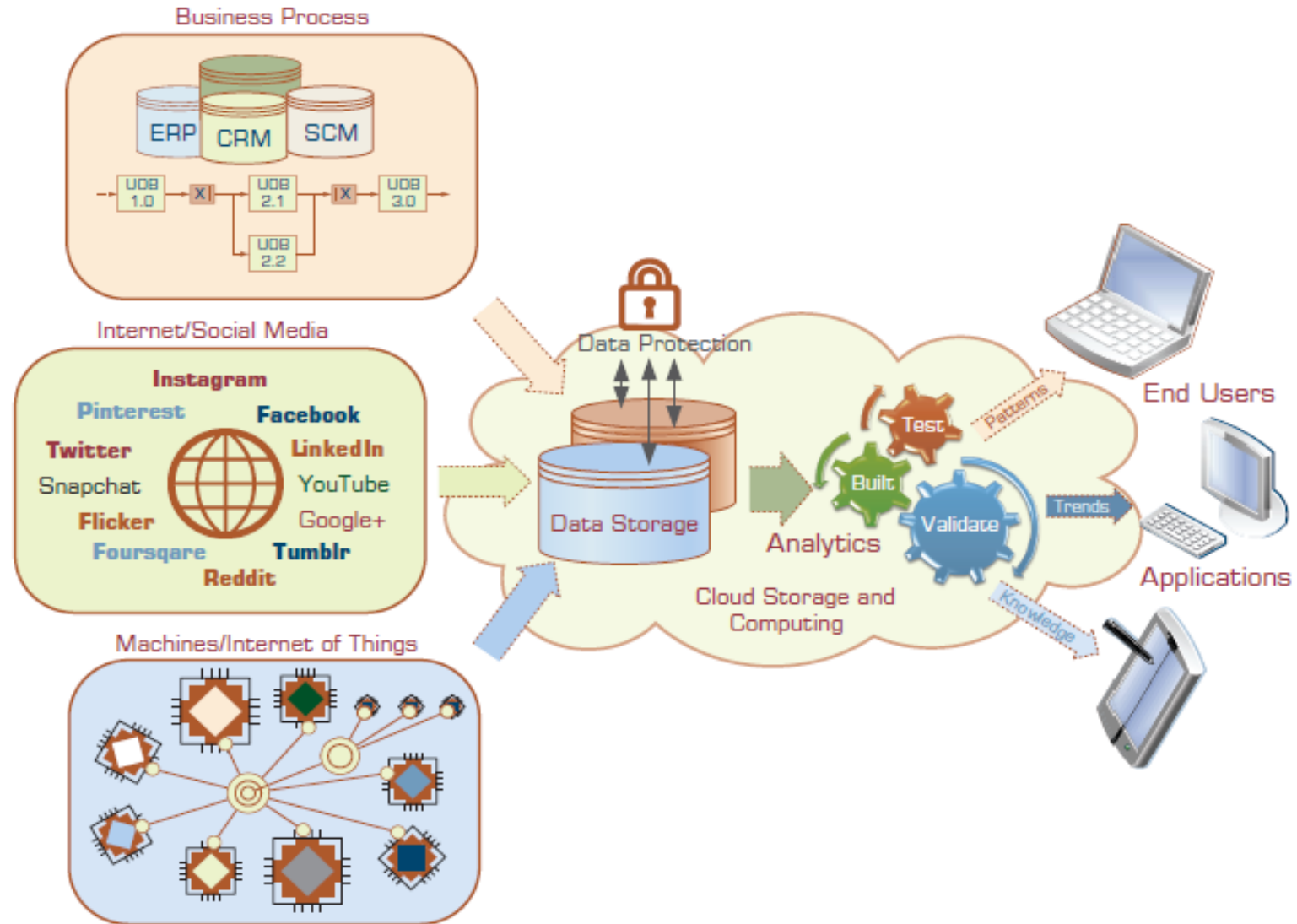
Os dados podem consistir em números, palavras, imagens, ...

Os dados são o menor nível de abstração (do qual são derivadas informações e conhecimento)

Os dados são fonte de informação e conhecimento.

Qualidade de dados e integridade de dados são críticas para análise

DADOS - NATUREZA



DADOS - TAXONOMIA

Dados = fatos

Dados estruturados

- Destinado a computadores para processar
- Numérico versus nominal

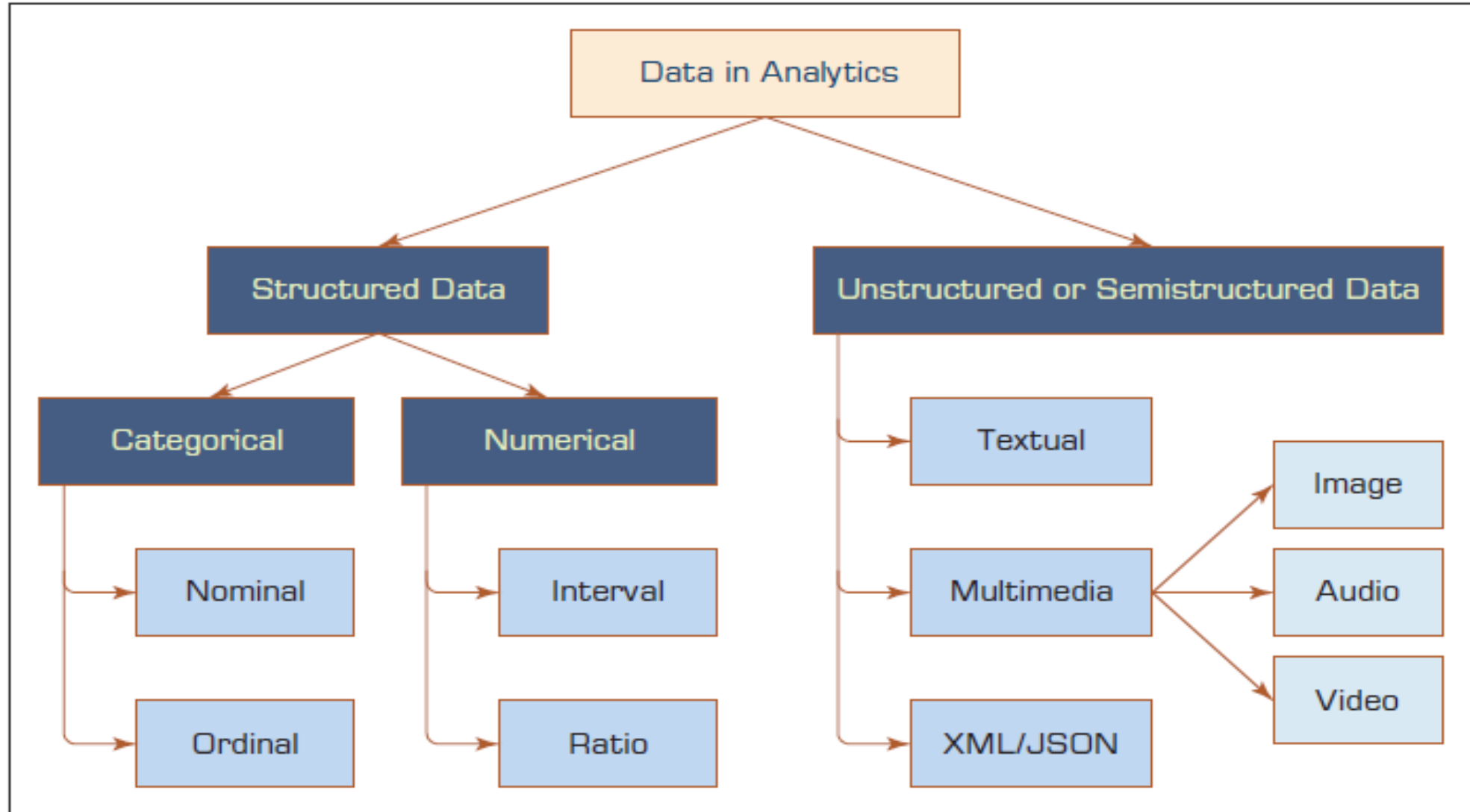
Dados não estruturados / textuais

- Destinado a humanos para processar / digerir

Dados semiestruturados?

- XML, HTML, arquivos de log, etc.

DADOS - TAXONOMIA



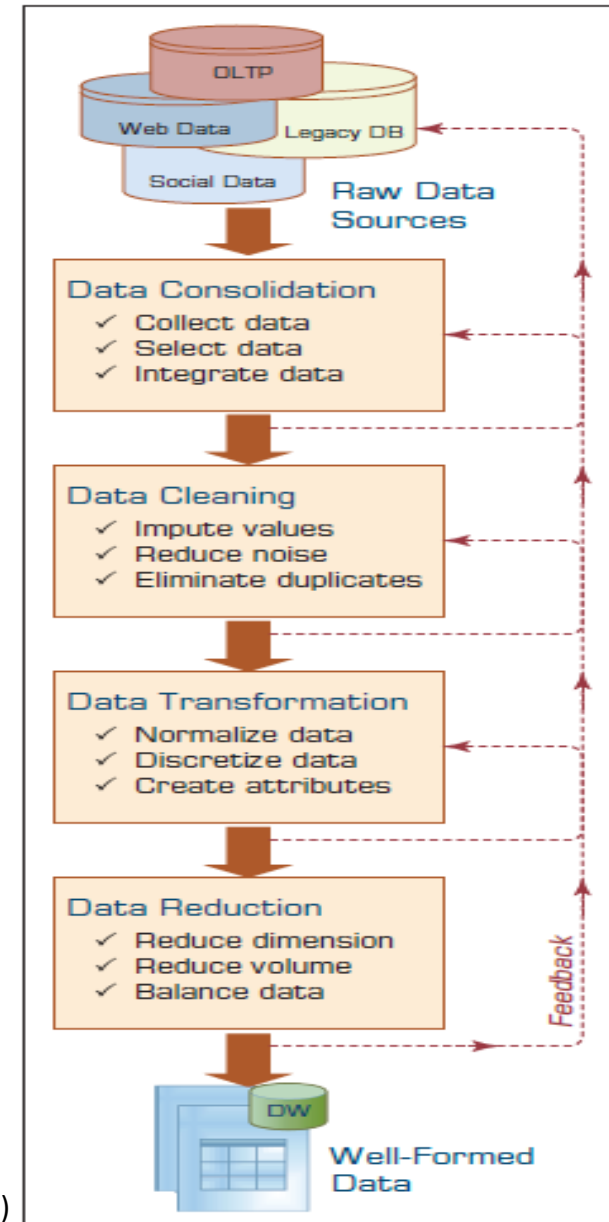
DADOS - PRÉ-PROCESSAMENTO

Os dados do mundo real estão sujos, desorganizados, excessivamente complexos e imprecisos

- Não estão prontos para análise!

Faz-se necessário preparar os dados para análise:

- Consolidação de dados
- Limpeza de dados
- Transformação de dados
- Redução de dados



DADOS - PRÉ-PROCESSAMENTO

Consolidação de dados

- Acessar e coletar dados
 - Consultas SQL;
 - Softwares agentes;
 - Web services;
- Selecionar e filtrar os dados
 - Experiências no domínio;
 - Consultas SQL;
 - Testes estatísticos.
- Integrar e unificar os dados
 - Experiências no domínio;
 - Consultas SQL;
 - Mapeamento de dados orientados à ontologia

DADOS - PRÉ-PROCESSAMENTO

Limpeza de dados

- Tratar dados nulos e vazios no conjunto de dados
 - Preencher campos vazios e nulos com valores mais apropriados (médias, medianas, mínimo, máximo, moda, etc);
 - Recodificar valores vazios e nulos com uma constante, como “NI”;
 - Remover registros com dados nulos e vazios;
 - Não fazer nada;
- Identificar e reduzir ruídos nos dados
 - Identificar os pontos extremos (outliers) nos dados com técnicas estatísticas simples (tais como médias e desvios padrões);
 - Identificar os pontos extremos (outliers) nos dados com análise de cluster e então removê-los ou suavizá-los por meio de regressão ou médias simples;
- Encontrar e eliminar dados errados
 - Identificar valores errados nos dados (diferente de outlier), e após identificá-los usar experiência de domínio para corrigi-los ou remover os registros com valores errados;

DADOS - PRÉ-PROCESSAMENTO

Transformação de dados

- Normalizar dados
 - Reduzir a amplitude dos dados em cada variável numérica para padronizar a amplitude (por exemplo 0 a 1, ou -1 a 1) por meio do uso de uma variedade de técnicas de normalização e scaling.
 - Padronizar tipos de dados (datas, escalas de medidas, etc)
- Discretizar os dados
 - Se for necessário converter variáveis numéricas em representações discretas usando amplitudes ou técnicas de fragmentação por frequência.
 - Para variáveis categóricas, reduzir o número de valores aplicando conceitos de hierarquias apropriados.
- Construir novos atributos
 - Derivar novas e mais informativas variáveis a partir de variáveis existentes usando uma ampla gama de funções matemáticas (adicionais, multiplicações entre outras)

DADOS - PRÉ-PROCESSAMENTO

Redução de dados

- Reduzir o número de atributos
 - Análise de componentes principais, análise de correlação.
- Reduzir o número de registros
 - Amostragem aleatória, amostragem estratificada, amostragem orientada por conhecimento especializado.
- Balanceamento de dados
 - Fazer oversample das classes menos representativas ou undersample da classe mais representativa.

O QUE É UM DATA WAREHOUSE?

Um repositório físico onde os dados relacionais são especialmente organizados para fornecer dados corporativos limpos em um formato padronizado

Um banco de dados relacional? (Então qual é a diferença?)

"O data warehouse é uma coleção de bases de dados **integradas**, **orientadas a assuntos**, projetadas para suportar funções DSS, onde cada unidade de dados **não é volátil** e **relevante** em algum momento no tempo

INTELIGÊNCIA DE NEGÓCIOS - COMPONENTES

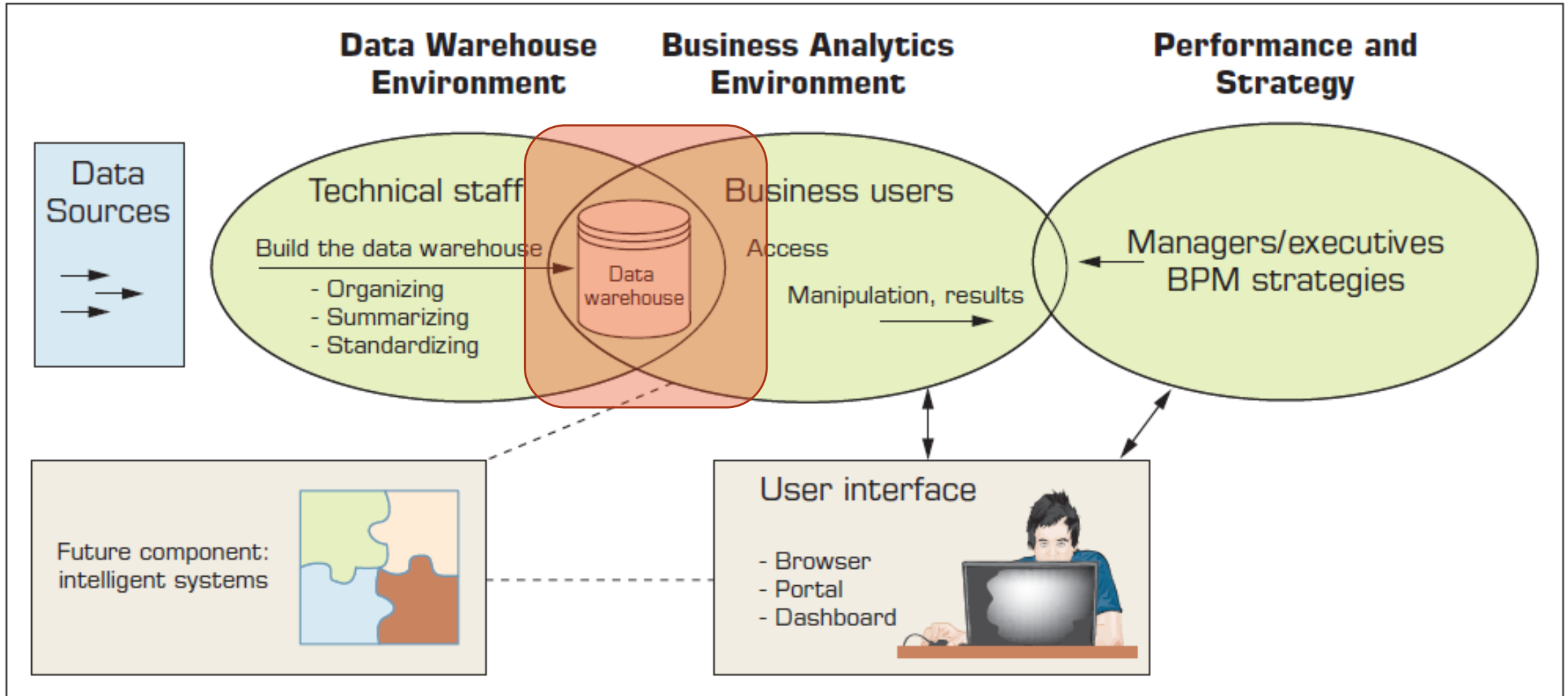
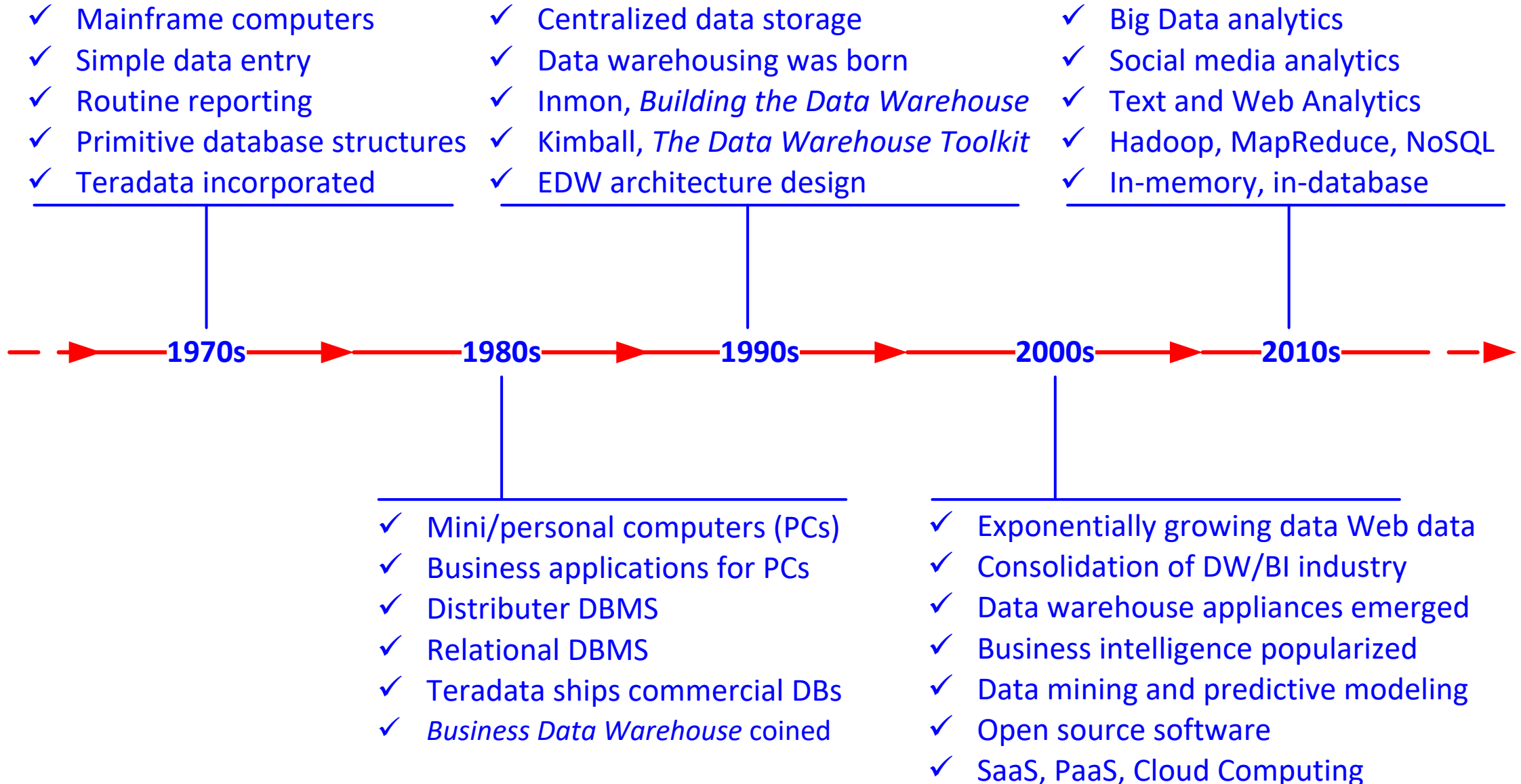


Figura 1.10 A High-Level Architecture of BI - Fonte: (SHARDA,DELEN,TURBAN ,2017)

PERSPECTIVA HISTÓRICA PARA DATA WAREHOUSING



CARACTERÍSTICAS DOS DWS

Orientado a assuntos

Integrado

Variante de tempo (séries temporais)

Não volátil

Resumido

Não normalizado

Metadados

Baseado na Web,

Relacional / multidimensional

Cliente / servidor

DATA MART

Um "DW" departamental de pequena escala que armazena apenas dados limitados / relevantes

- Data mart dependentes
 - Um subconjunto que é criado diretamente de um data warehouse
- Data mart independente
 - Um pequeno data warehouse projetado para uma unidade de negócios estratégica ou um departamento

OUTROS COMPONENTES DE DW

Repositório de dados operacionais (ODS)

- Um tipo de banco de dados frequentemente usado como uma área provisória para um data warehouse

Oper marts

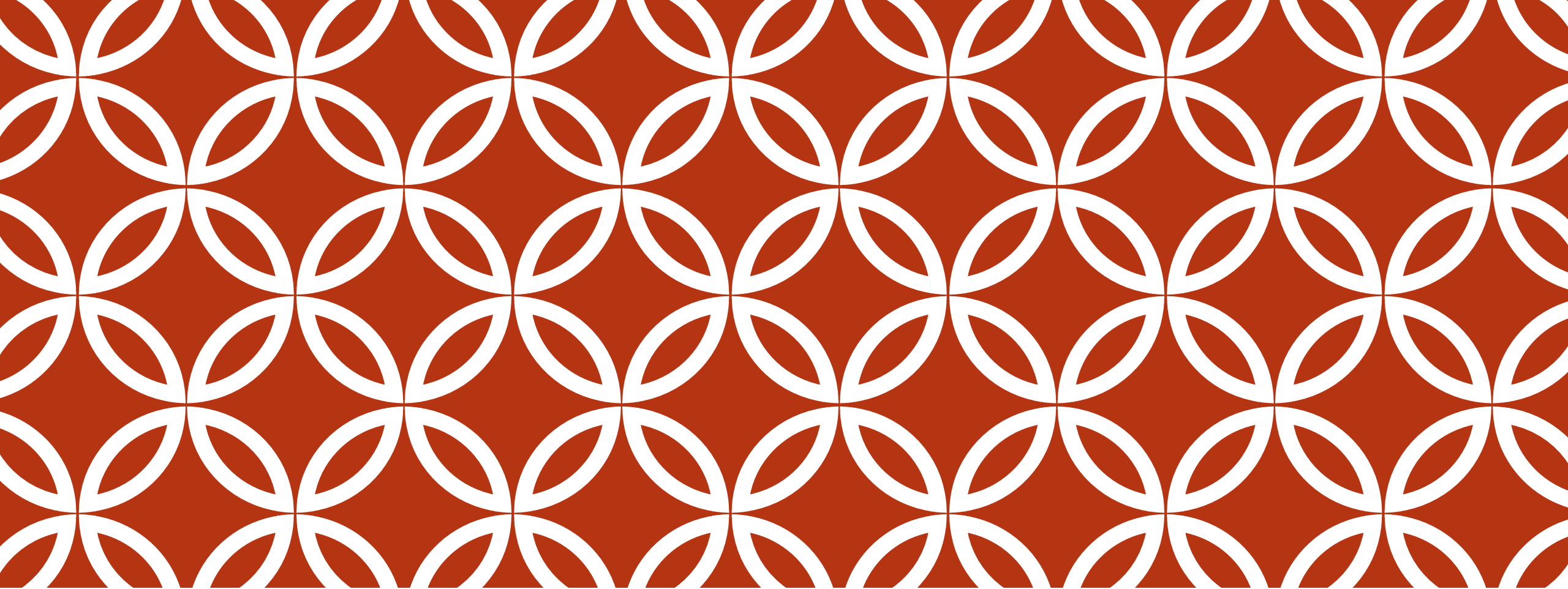
- Um data mart operacional

Data warehouse corporativos (EDW)

- Um data warehouse para toda a empresa

Metadados - "dados sobre dados"

- Os metadados descrevem o conteúdo de um data warehouse e sua coleta e utilização



BUSSINESS INTELIGENCE — BI E QLIKSENSE

AULA 04

AULA PRÁTICA

PRÉ-PROCESSAMENTO DOS DADOS

Apresentação do conjunto de dados a ser utilizado

Carga e pré-processamento dos dados selecionados utilizando a ferramenta *Qlik Sense Desktop*.



BUSSINESS INTELIGENCE — BI E QLIKSENSE

AULA 05

INTELIGÊNCIA DE NEGÓCIOS - COMPONENTES

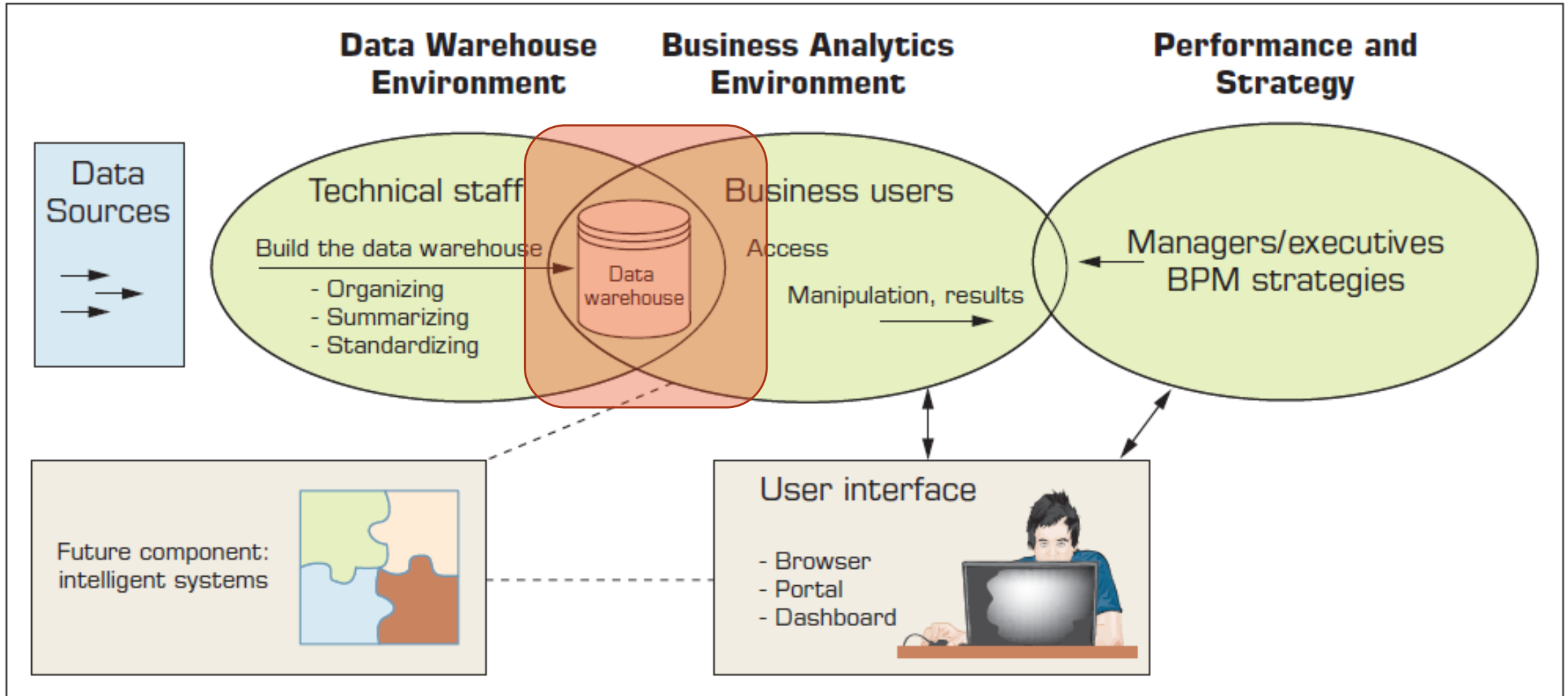
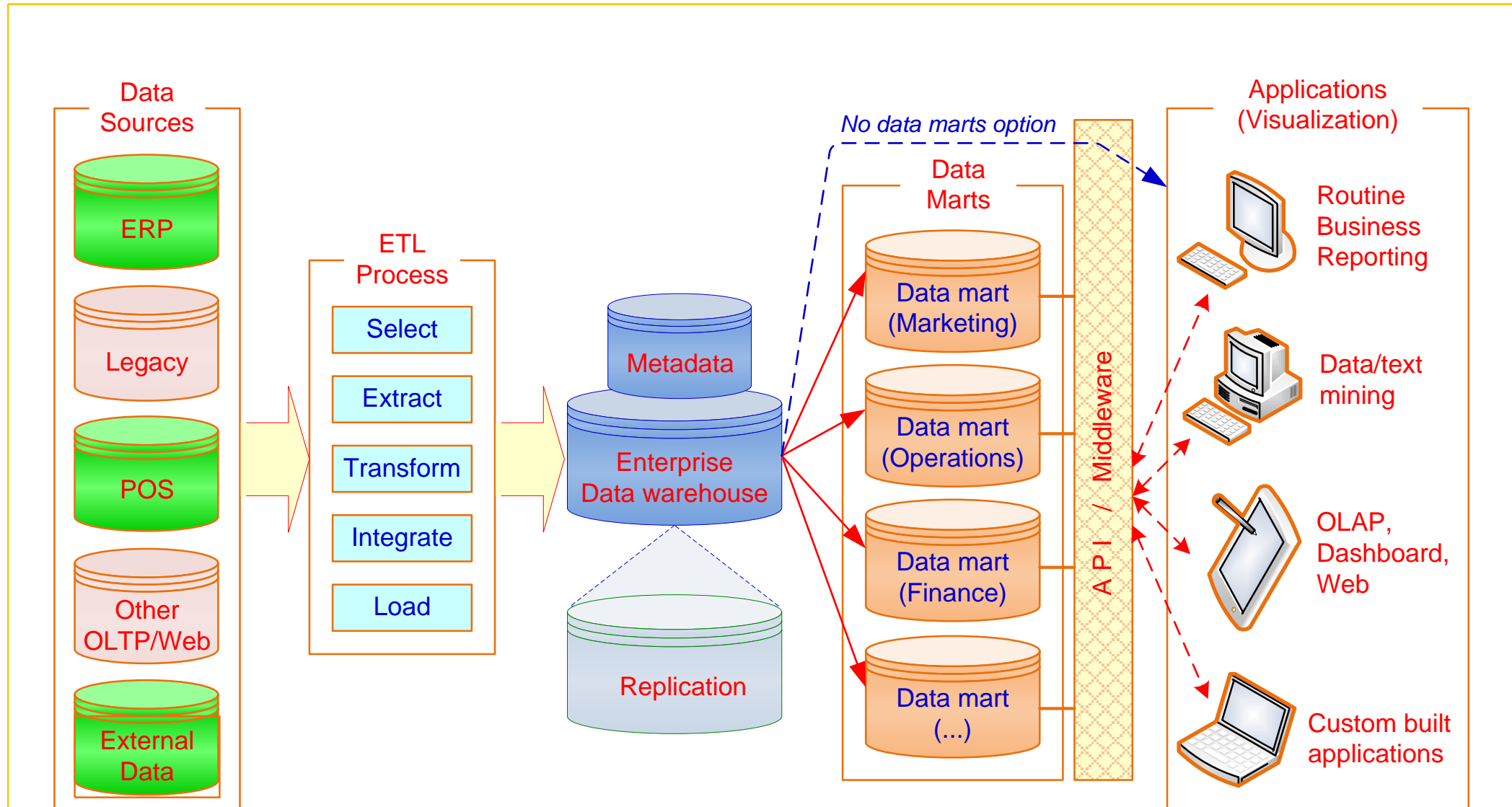


Figura 1.10 A High-Level Architecture of BI - Fonte: (SHARDA,DELEN,TURBAN ,2017)

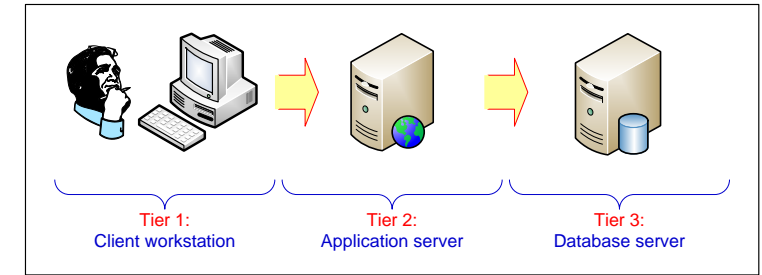
ARQUITETURA GENÉRICA DE DW



ARQUITETURAS DE DW

Arquitetura de três camadas

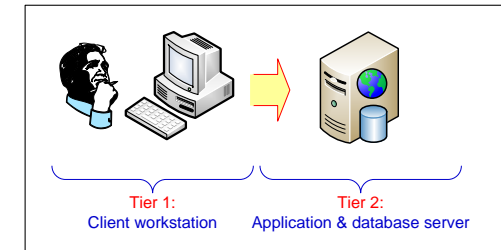
- Software de aquisição de dados (back-end)
- O data warehouse que contém os dados e software
- Software de cliente (front-end) que permite aos usuários acessar e analisar dados do repositório



Arquitetura de duas camadas

- As primeiras duas camadas na arquitetura de três camadas são combinadas em uma

... às vezes há apenas um nível?



ARQUITETURAS DE DW

Questões a serem consideradas ao decidir qual arquitetura usar:

- Qual sistema de gerenciamento de banco de dados (SGBD) deve ser usado?
- Processamento paralelo e / ou particionamento serão usados?
- Ferramentas de migração de dados serão usadas para carregar o data warehouse?
- Quais ferramentas serão usadas para suportar a recuperação e análise de dados?

ARQUITETURAS DE DW

Dez fatores que potencialmente afetam a decisão de seleção de arquitetura

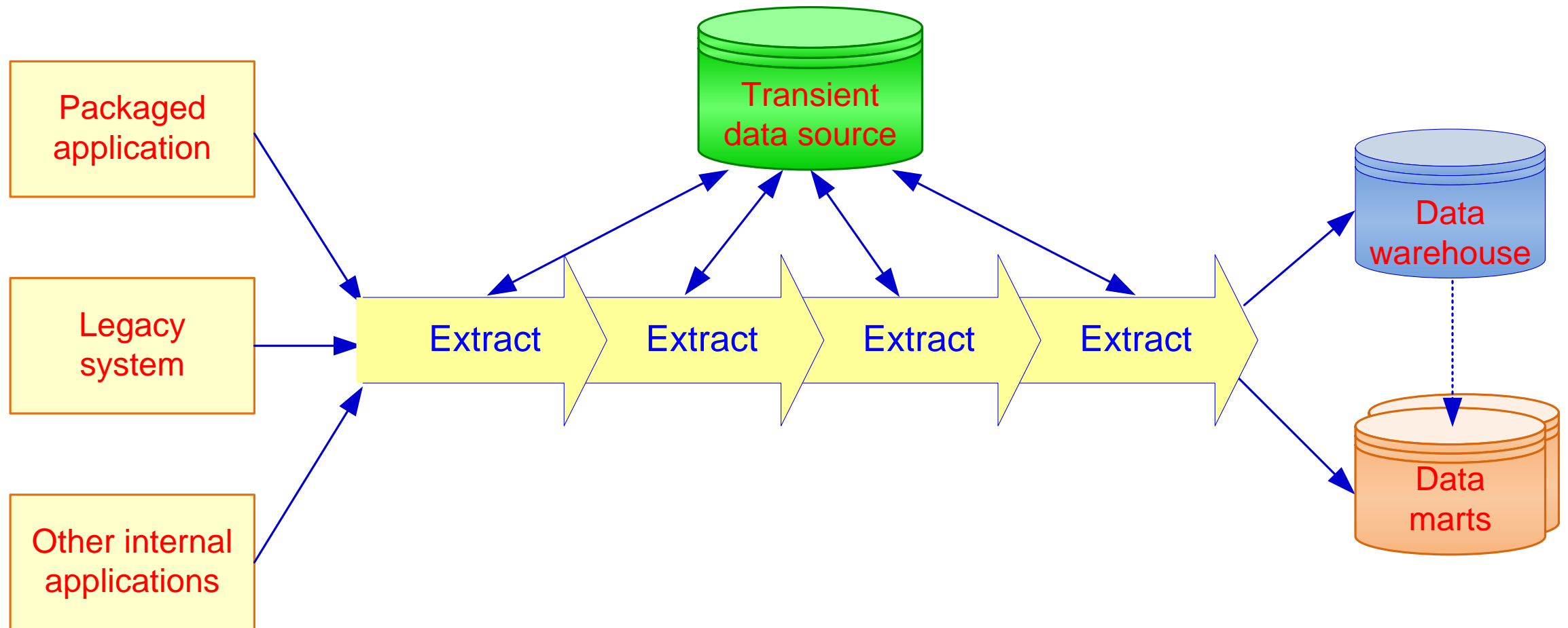
- Interdependência da informação entre unidades organizacionais
- Necessidades de informação da alta administração
- Urgência de necessidade de um data warehouse
- Natureza das tarefas do usuário final
- Restrições de recursos
- Visão estratégica do data warehouse antes da implementação
- Compatibilidade com sistemas existentes
- Habilidade perceptível da equipe de TI interna
- Problemas técnicos e Fatores sociais / políticos

INTEGRAÇÃO DE DADOS E EXTRAÇÃO, TRANSFORMAÇÃO E CARGA

ETL = Extract Transform Load (Extração, Transformação e Carga)

- Integração de dados
 - Integração que compreende três processos importantes: acesso a dados, federação de dados e captura de mudanças.
- Integração de aplicações corporativas (EAI)
 - Uma tecnologia que fornece um veículo para empurrar dados de sistemas de origem para um data warehouse
- Integração de informações empresariais (EII)
 - Um espaço de ferramentas em evolução que promete integração de dados em tempo real a partir de uma variedade de fontes, como bancos de dados relacionais ou multidimensionais, serviços da Web, etc.

INTEGRAÇÃO DE DADOS E EXTRAÇÃO, TRANSFORMAÇÃO E CARGA



INTEGRAÇÃO DE DADOS E EXTRAÇÃO, TRANSFORMAÇÃO E CARGA

The screenshot shows the Microsoft Visual Studio interface during the execution of an SSIS package. The main window displays a data flow diagram with the following tasks and connections:

- TRUNCAR AS TABELAS DE DESTINO** (Truncate destination tables) - Success
- IMPORTAR DADOS DO BANCO ACCESS** (Import data from Access database) - Success
 - IMPORTA PROJETOS** (Import projects) - Success
 - IMPORTA ATIVIDADES** (Import activities) - Success
- TRATAR OS DADOS PARA O PAINEL DE BI** (Process data for BI dashboard) - Success
 - EXTRAIR PROJETOS_ENTREGAS_HISTORICO_ANDAMENTO** (Extract project delivery history) - Success
 - EXTRAIR PROJETOS_ENTREGAS_HISTORICO_PERC_EXECUCAO** (Extract project delivery history percentage) - Success
 - EXTRAIR DEPARTAMENTOS_SECRETARIA** (Extract secretariat departments) - Success
 - EXTRAIR DEPARTAMENTOS_SECRETARIA_UNIDADES** (Extract secretariat units) - Success
 - EXTRAIR PROJETOS_ENTREGAS** (Extract project deliveries) - Success
 - EXTRAIR PROJETOS** (Extract projects) - Success

The status bar at the bottom indicates: **Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.**

The interface also shows the **Autos** window, **Call Stack**, and **Connection Managers** (10.209.40.112.seges_cginf_stage.userOrlando, seges-projetos).

FERRAMENTAS DE ETL — SELEÇÃO E AQUISIÇÃO

Problemas que afetam a compra de uma ferramenta ETL

- As ferramentas de transformação de dados são caras
- As ferramentas de transformação de dados podem ter uma longa curva de aprendizado

Critérios importantes na seleção de uma ferramenta ETL

- Capacidade de ler e escrever para um número ilimitado de fontes de dados / arquiteturas
- Captura e entrega automática de metadados
- Um histórico de conformidade com os padrões abertos
- Uma interface fácil de usar para o desenvolvedor e o usuário funcional

ABORDAGENS DE DESENVOLVIMENTO DE DATA WAREHOUSE

Modelo Inmon: abordagem EDW

- de cima para baixo

Modelo Kimball: abordagem Data Mart

- de baixo para cima

Qual modelo é o melhor?

REPRESENTAÇÃO DE DADOS EM UM DW

Modelagem dimensional

- Um sistema baseado em recuperação de dados que suporta alto volume de acesso de consulta de

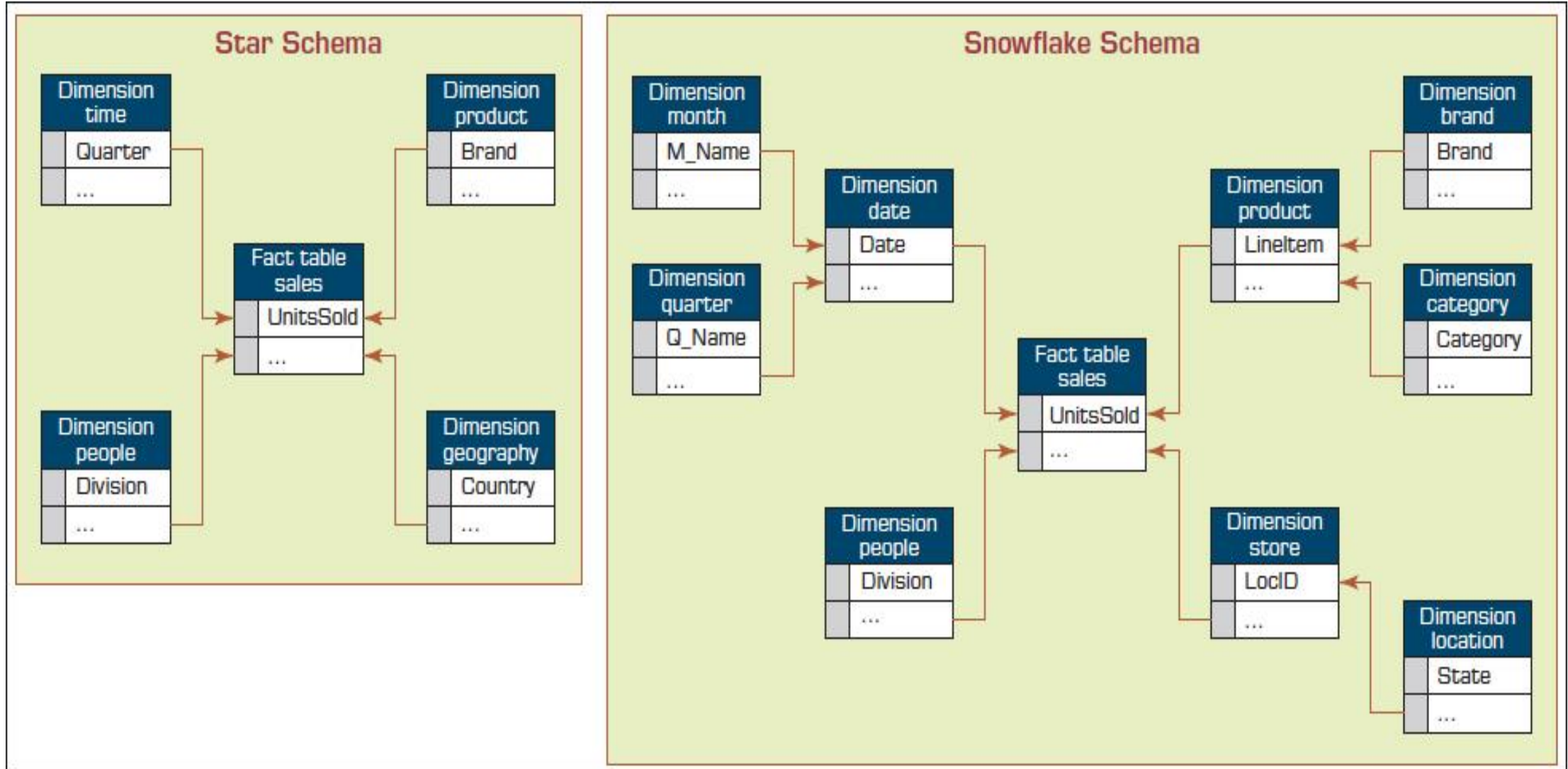
Esquema de estrela

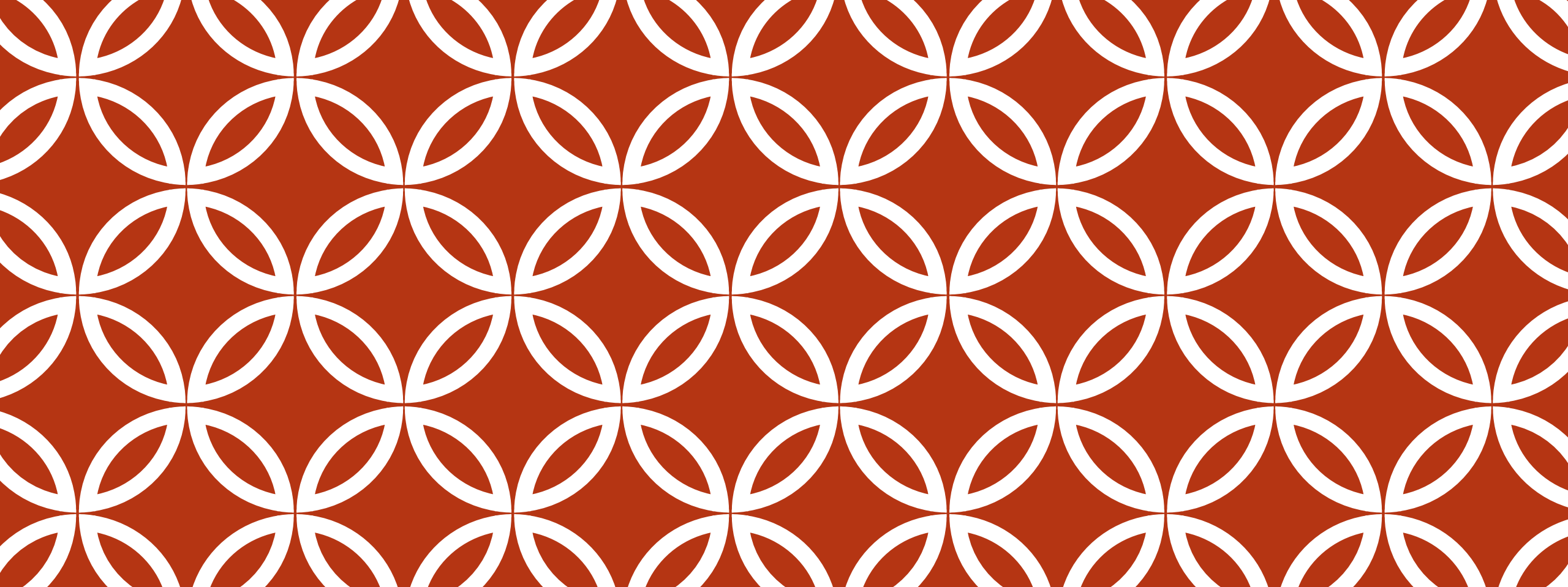
- O estilo mais comum e o estilo mais simples de modelagem dimensional
- Contém uma tabela de fatos cercada e conectada a várias tabelas de dimensão

Esquema de flocos de neve

- Uma extensão do esquema de estrelas onde o diagrama se assemelha, em forma, a um floco de neve

REPRESENTAÇÃO DE DADOS EM UM DW





BUSSINESS INTELIGENCE — BI E QLIKSENSE

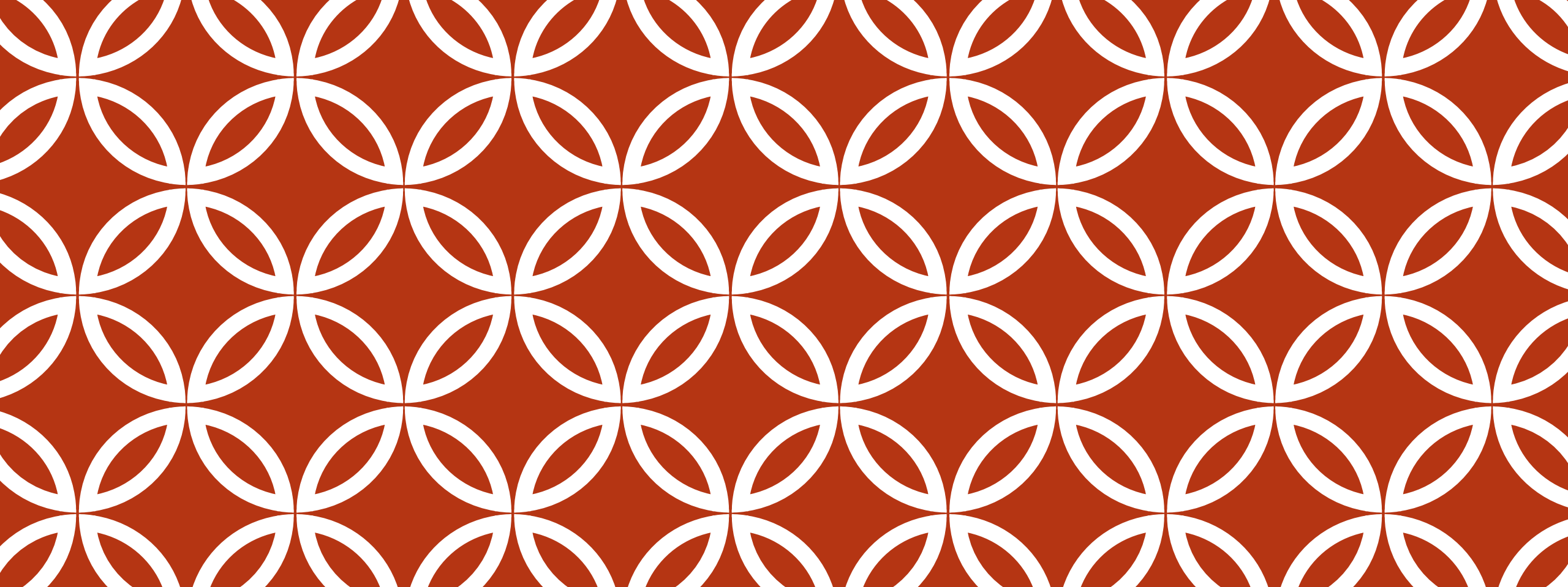
AULA 06

AULA PRÁTICA

METADADOS E VISUALIZAÇÕES

Manipulação de dados utilizando scripts QlikSense.

Demonstração de organização dos dados de orçamento em modelos multidimensionais no formato e estrela e floco de neve.



BUSSINESS INTELIGENCE — BI E QLIKSENSE

AULA 07

INTELIGÊNCIA DE NEGÓCIOS - COMPONENTES

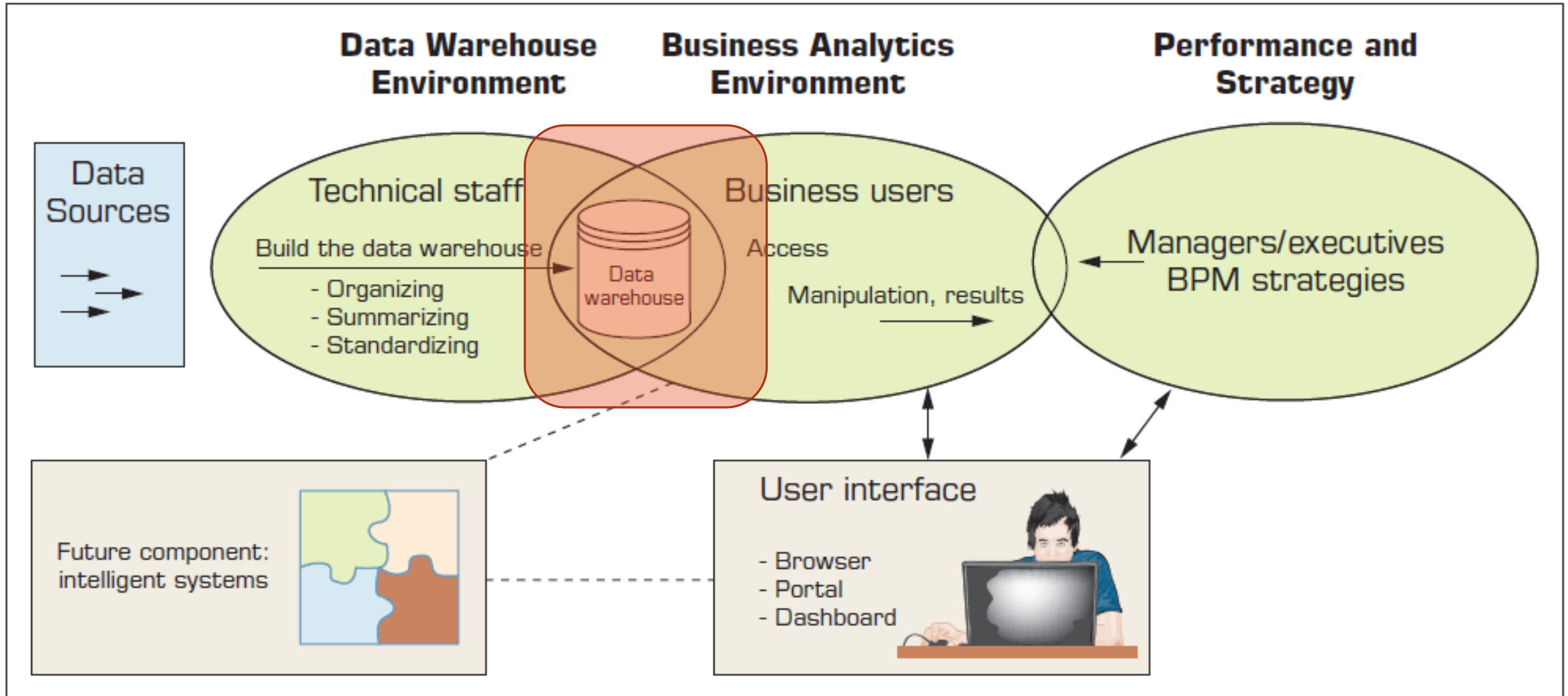


Figura 1.10 A High-Level Architecture of BI - Fonte: (SHARDA,DELEN,TURBAN ,2017)

MULTIDIMENSIONALIDADE

A capacidade de organizar, apresentar e analisar dados por várias dimensões, como vendas por região, por produto, por vendedor e por tempo (quatro dimensões)

Apresentação multidimensional

- Dimensões: produtos, vendedores, segmentos de mercado, unidades comerciais, locais geográficos, canais de distribuição, país ou indústria
- Medidas: dinheiro, volume de vendas, contagem de títulos, apuração de lucro
- Tempo: diária, semanal, mensal, trimestral ou anual

ANÁLISE DE DADOS EM DW

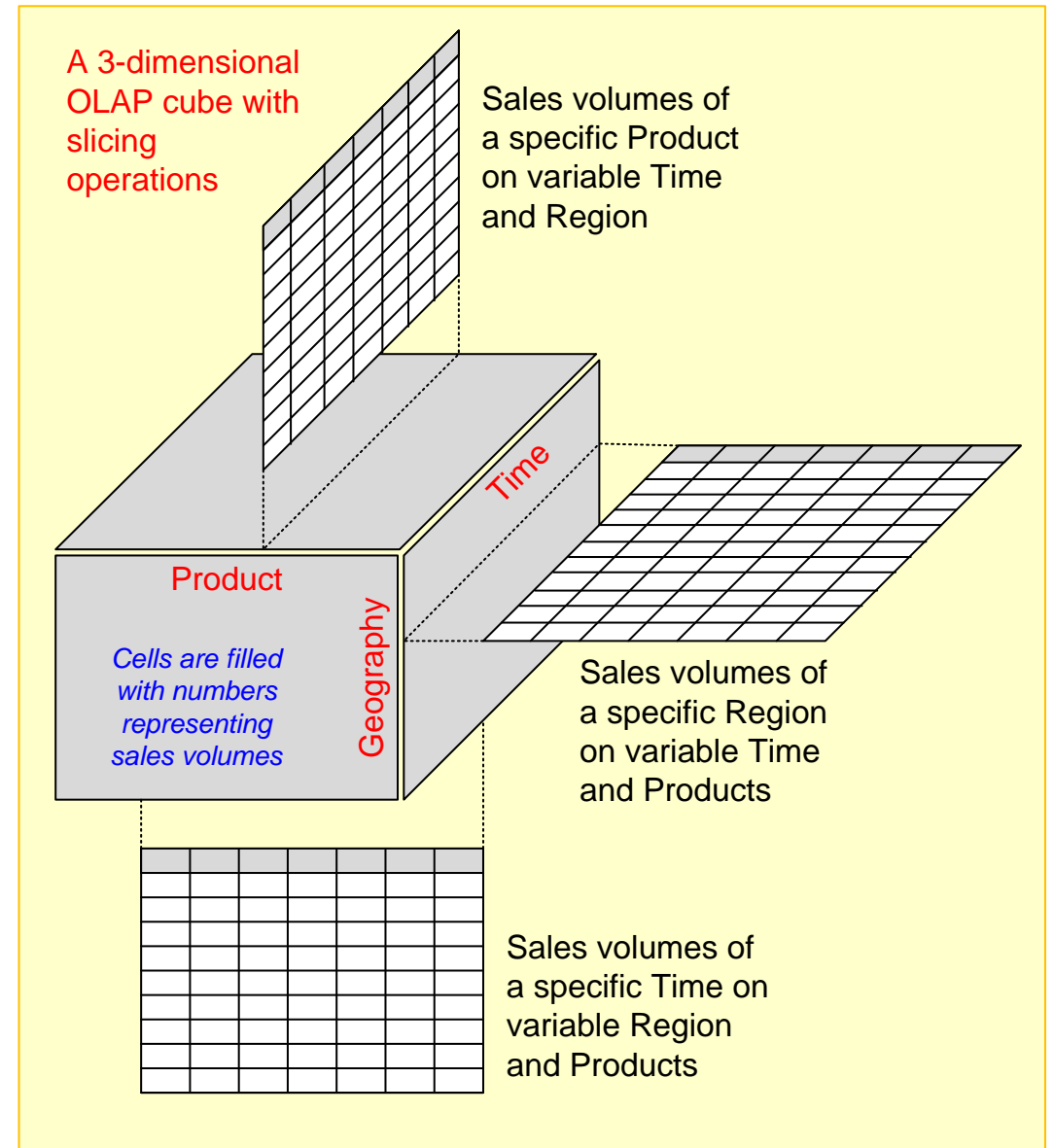
OLTP vs. OLAP ...

OLTP (Online Transaction Processing)

- Capturar e armazenar dados de ERP, CRM, POS, ...
- O foco principal é a eficiência das tarefas rotineiras

OLAP (Online Analytical Processing)

- Convertendo dados em informações para suporte de decisão
- Cubos de dados, drill-down / rollup, slice & dice.
- Gerar relatórios ad hoc
- Realização de análises estatísticas e outras análises



IMPLEMENTAÇÃO BEM SUCEDIDA DE DW: COISAS A EVITAR

Começar com o patrocínio errado

Definir expectativas que você não são viáveis

Carregar o data warehouse com informações apenas porque está disponível

Acreditar que o projeto de banco de dados de data warehousing é o mesmo que o projeto de banco de dados transacional

DATA WAREHOUSE E ESCALABILIDADE

Escalabilidade

As principais questões relativas à escalabilidade:

- A quantidade de dados no repositório
- Rapidez com que o repositório cresce
- O número de usuários simultâneos
- A complexidade das consultas dos usuários

Uma boa escalabilidade considera que consultas e outras funções de acesso a dados crescerão linearmente com o tamanho do repositório

DATA WAREHOUSE ADMINISTRAÇÃO E SEGURANÇA

Administrador do data warehouse (DWA) deve ...

- Ter conhecimento de software de alto desempenho, hardware e tecnologias de rede
- Possuir sólidos conhecimentos e percepção do negócio
- Estar familiarizado com os processos de tomada de decisão de modo a adequadamente projetar / manter a estrutura do data warehouse
- Possuir excelentes habilidades de comunicação

Segurança e privacidade é uma questão urgente em DW

- Salvaguardando os ativos mais valiosos
- Regulamentos governamentais
- Deve ser explicitamente planejada e executada

DATA LAKES (LAGOS DE DADOS)

Tecnologia de armazenamento de dados não estruturada para Big Data

Data Lake vs Data Warehouse

TABLE 3.6 A Simple Comparison between a Data Warehouse and a Data Lake

Dimension	Data Warehouse	Data Lake
The nature of data	Structured, processed	Any data in raw/native format
Processing	Schema-on-write (SQL)	Schema-on-read (NoSQL)
Retrieval speed	Very fast	Slow
Cost	Expensive for large data volumes	Designed for low-cost storage
Agility	Less agile, fixed configuration	Highly agile, flexible configuration
Novelty/newness	Not new/matured	Very new/maturing
Security	Well-secured	Not yet well-secured
Users	Business professionals	Data scientists

INTELIGÊNCIA DE NEGÓCIOS - COMPONENTES

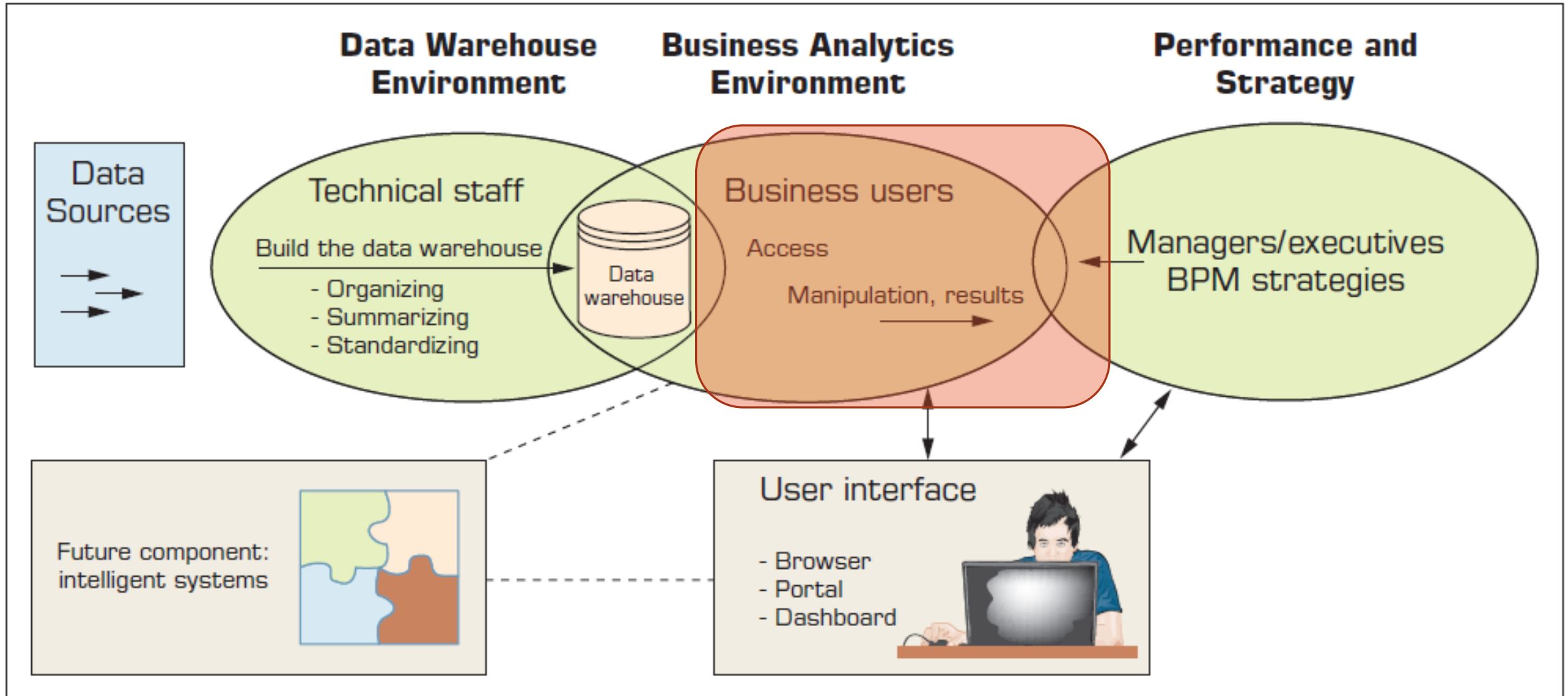
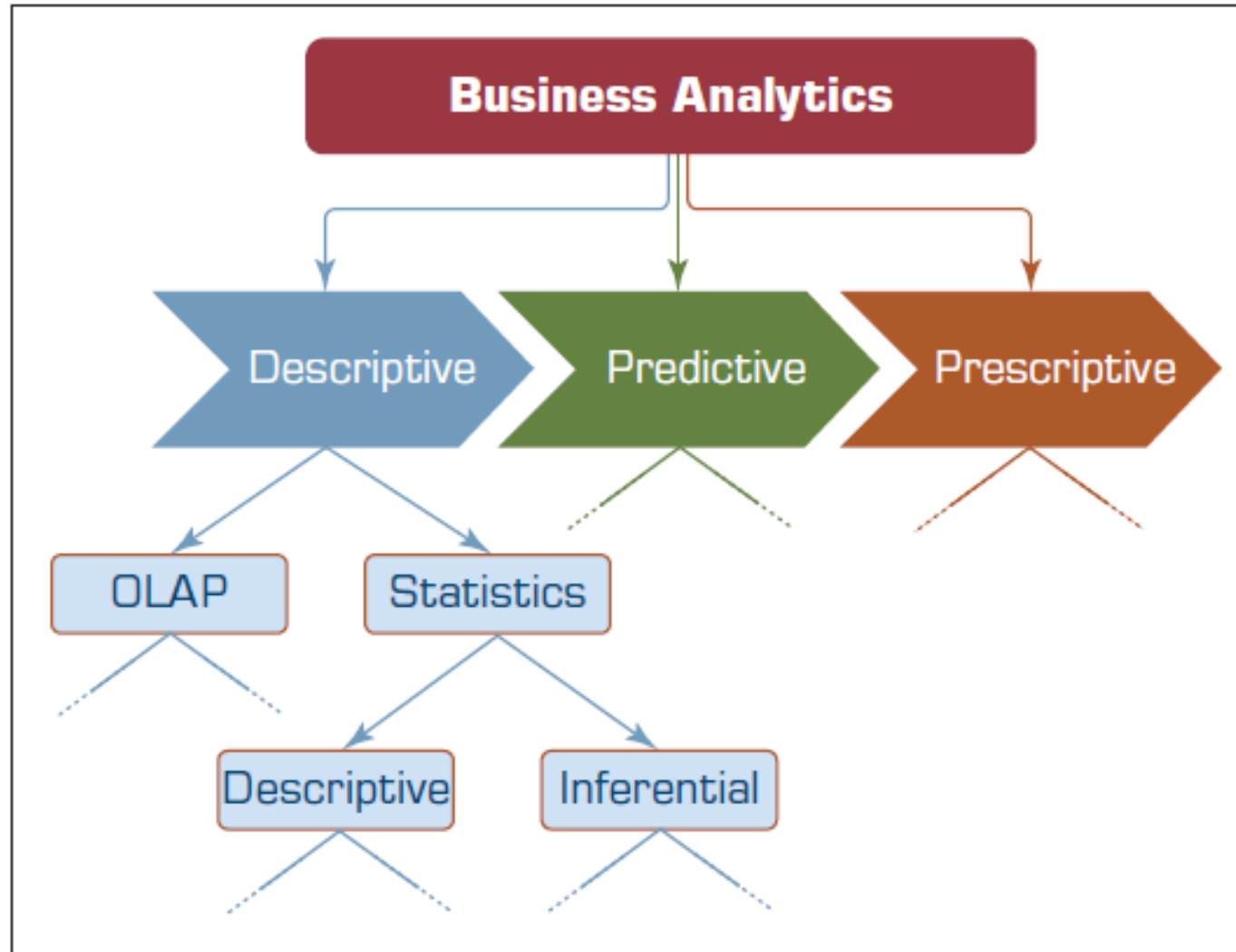


Figura 1.10 A High-Level Architecture of BI - Fonte: (SHARDA,DELEN,TURBAN ,2017)

MODELAGEM ESTATÍSTICA



DADOS — ESTATÍSTICA DESCRITIVA

Medidas de tendência de centralidade

- Média aritmética

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n}$$

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Mediana

- O número no meio

Moda

- A observação mais frequente

DADOS - ESTATÍSTICA DESCRITIVA

Medidas de Dispersão

Dispersão

- Grau de variação em uma determinada variável

Amplitude

- Max - Min

Variância

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$

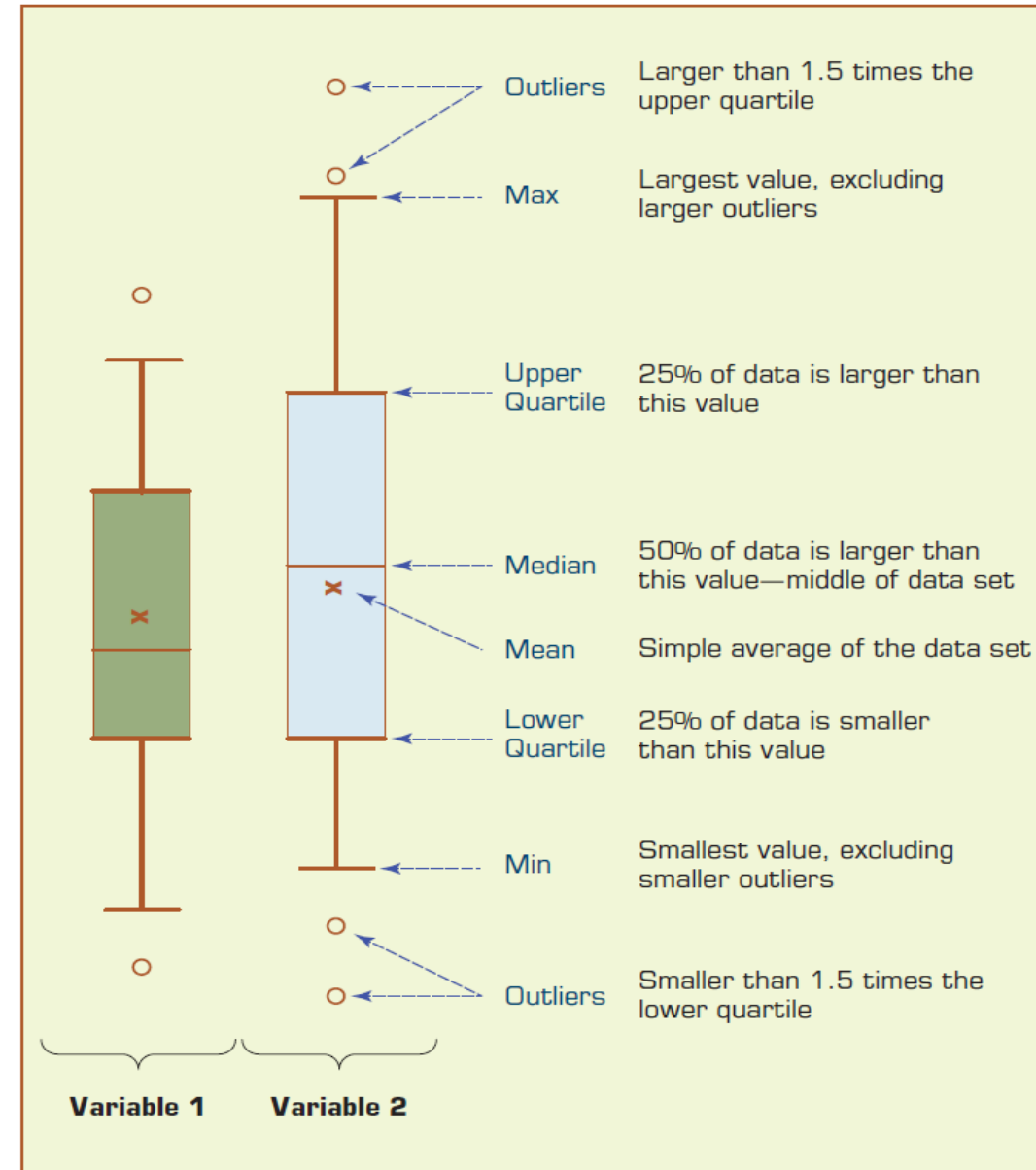
- Desvio Padrão

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

DADOS - ESTATÍSTICA DESCRITIVA

Medidas de Dispersão

- Quartis
- Box-and-Whiskers Plot
- Versátil e informativo

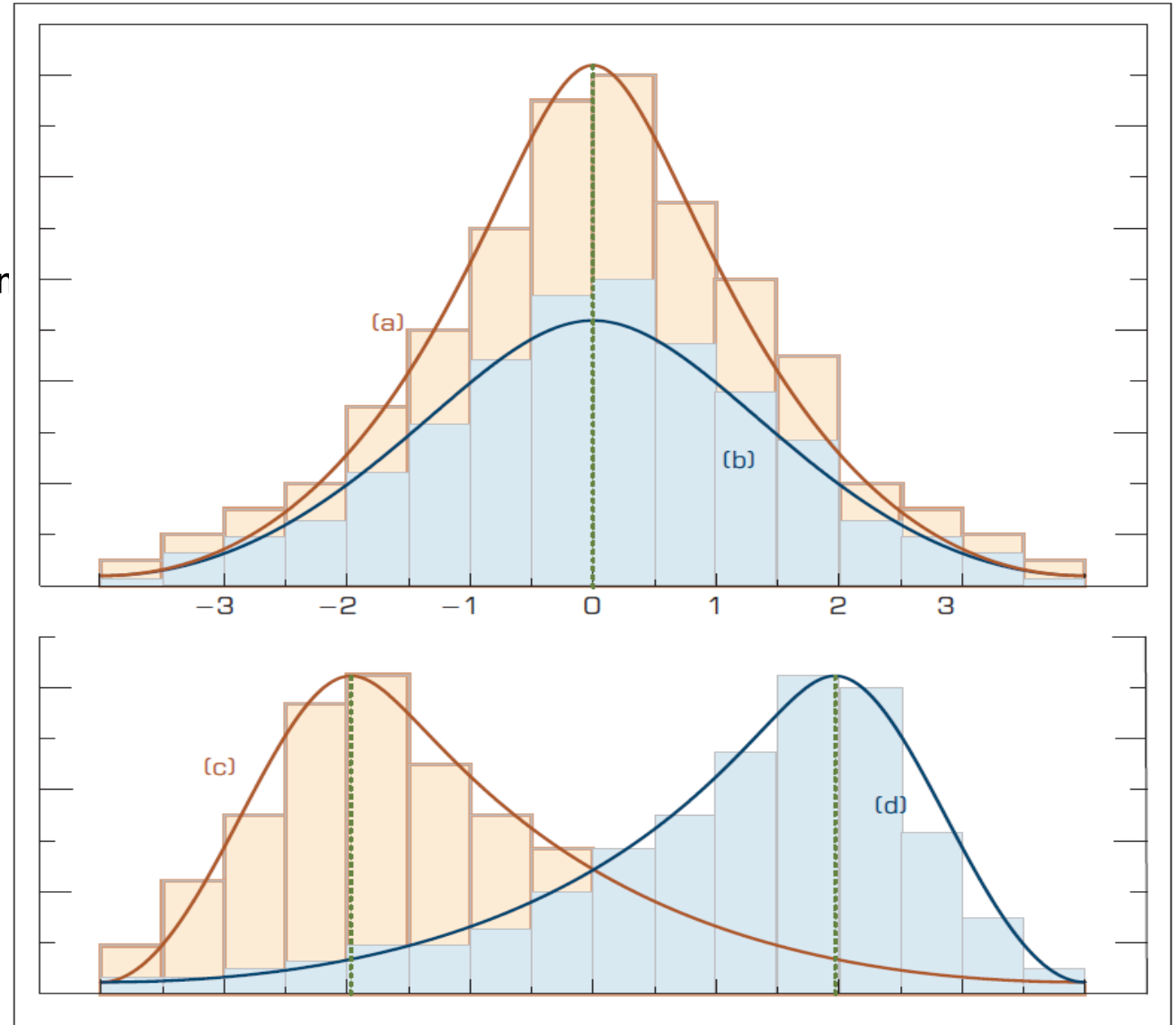


DADOS - ESTATÍSTICA DESCRITIVA

Forma da Distribuição

- Histograma - gráfico de frequência
- Skewness (Medida de assimetria)
 - Se $v > 0$, maior peso na cauda direita (valores acima da média)
 - Se $v < 0$, maior peso na cauda esquerda (valores abaixo da média)
 - Se $v = 0$, simétrica

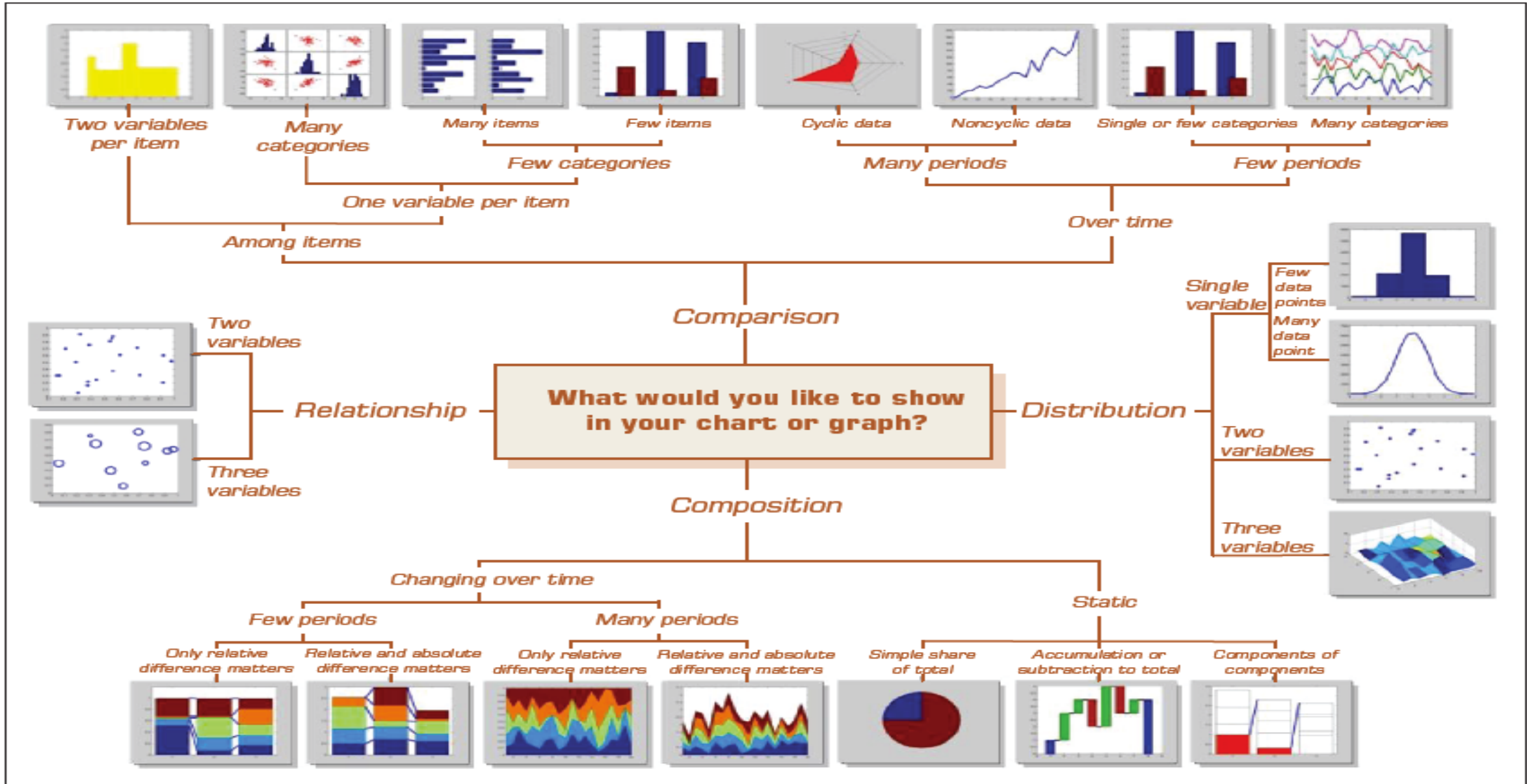
$$\text{Skewness} = S = \frac{\sum_{i=1}^n (x_i - \bar{x})^3}{(n - 1)s^3}$$

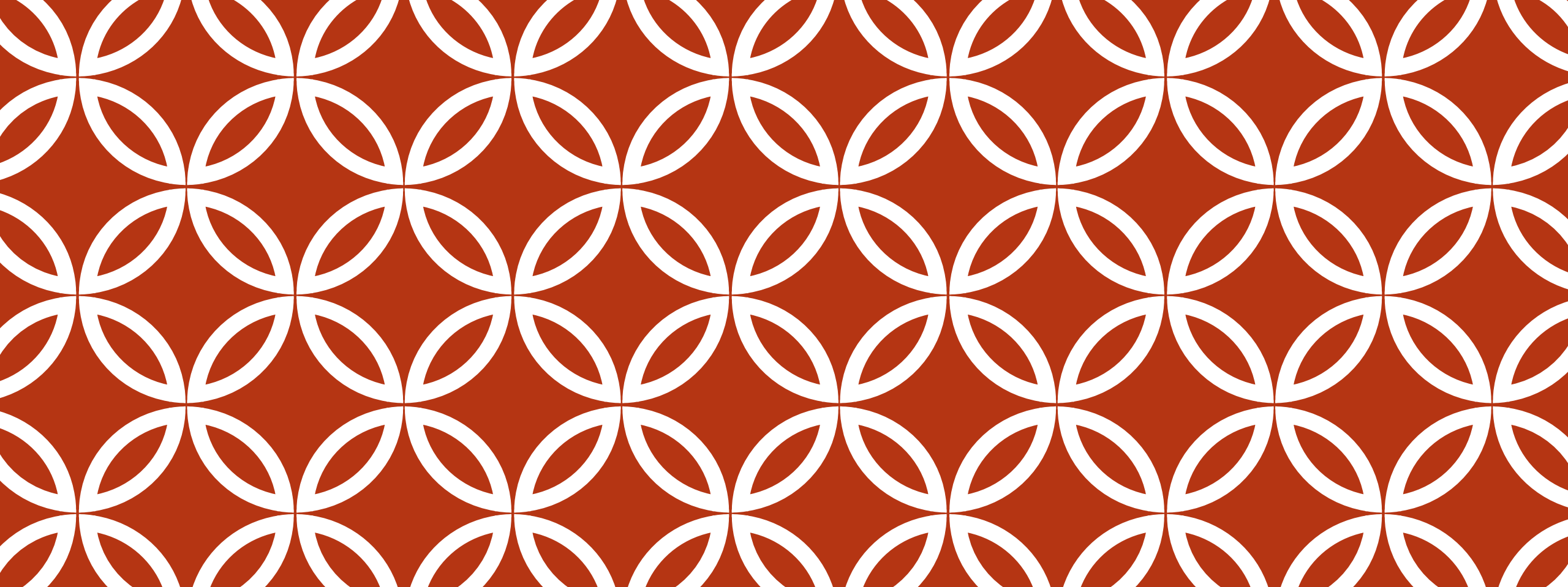


DADOS — VISUALIZAÇÃO DE DADOS

- "O uso de representações visuais para explorar, fazer sentido e comunicar dados".
- Visualização de dados vs. Visualização de informações
- Informação = agregação, resumo e contextualização de dados
- Relacionado a gráficos de informação, visualização científica e gráficos estatísticos
- Muitas vezes inclui quadros, gráficos, ilustrações
- Ferramentas Emergentes de Visualização de dados
 - Tableau, Spotfire, QlikSense, PowerBI

DADOS – VISUALIZAÇÃO DE DADOS





BUSSINESS INTELIGENCE — BI E QLIKSENSE

AULA 08

AULA PRÁTICA

METADADOS E VISUALIZAÇÕES

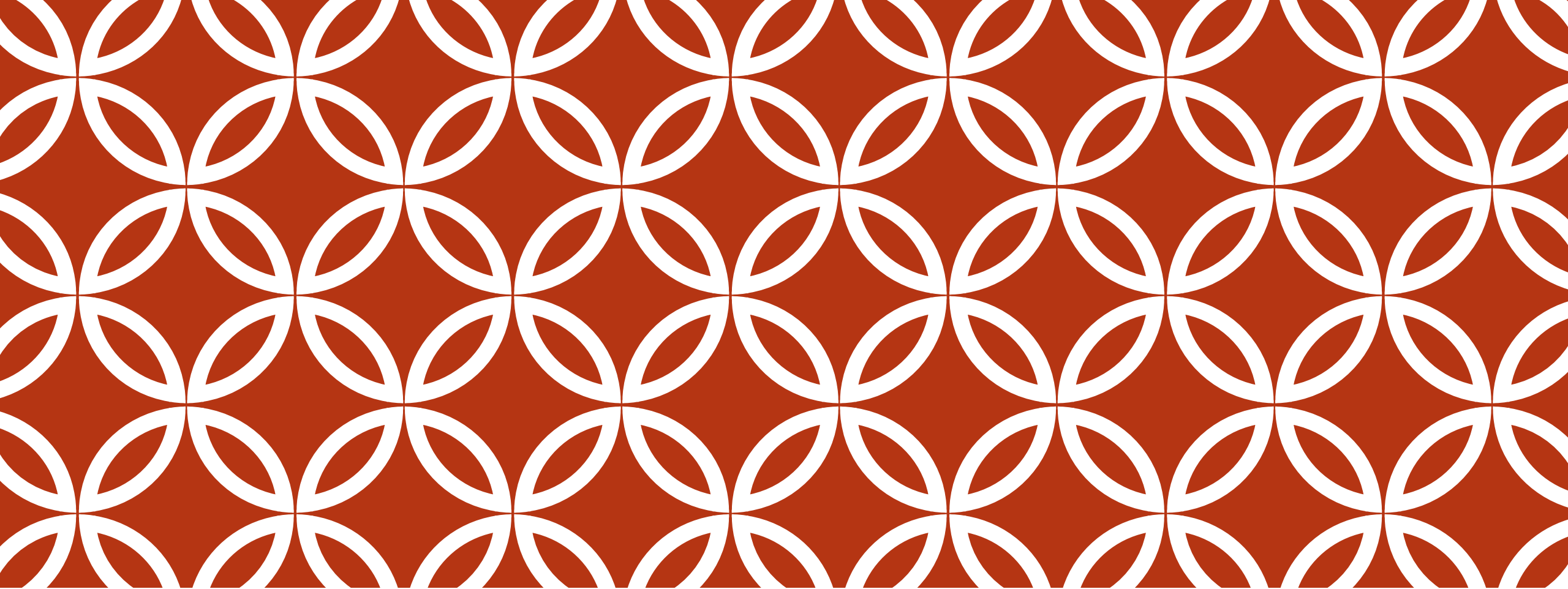
Manipulação de dados utilizando scripts Qlik Sense.

Definição de metadados (métricas e dimensões); e

Criação de visualizações de dados utilizando a ferramenta de *Qlik Sense Desktop*.

Link do Painel com várias abordagens de cargas

<https://drive.google.com/open?id=1BRugClHOfp6x0xzrcrgeLBuTRegw-5r3>



BUSSINESS INTELIGENCE — BI E QLIKSENSE

AULA 09

LINGUAGEM SQL

linguagem de pesquisa declarativa padrão para banco de dados relacional.

- Structured Query Language, ou
- Linguagem de Consulta Estruturada ou
- SQL

Pode sofrer alterações de sintaxe em diferentes fornecedores de Sistemas Gerenciadores de Bancos de Dados-SGDB

Composta de dois tipos de comandos:

- DDL - Data Definition Language ou Linguagem de Definição de Dados e
- DML - Data Manipulation Language, ou Linguagem de Manipulação de Dados

LINGUAGEM SQL

DDL - Data Definition Language
ou Linguagem de Definição de Dados e

LINGUAGEM SQL — DDL - CRIAR E APAGAR BANCO DE DADOS

Sintaxe:

```
CREATE DATABASE databasename;
```

```
DROP DATABASE databasename;
```

Exemplo:

```
CREATE DATABASE testDB;
```

```
DROP DATABASE testDB;
```

LINGUAGEM SQL — DDL - CRIAR E APAGAR TABELA

Sintaxe:

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

```
DROP TABLE table_name;
```

```
TRUNCATE TABLE table_name;
```

Exemplo:

```
CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);
```

```
DROP TABLE Persons;
```

```
TRUNCATE TABLE Persons;
```

LINGUAGEM SQL — DDL — ALTERAR TABELA

Sintaxe:

```
ALTER TABLE table_name  
ADD column_name datatype;
```

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

```
ALTER TABLE table_name  
ALTER COLUMN column_name  
datatype;
```

Exemplo:

```
ALTER TABLE Persons  
ADD Street varchar(255) ;
```

```
ALTER TABLE Persons  
DROP COLUMN Street;
```

```
ALTER TABLE Persons  
ALTER COLUMN Street varchar(100)
```


LINGUAGEM SQL — DDL - CONSTRAINT

Pode colocar restrições para limitar o tipo de dados a introduzir numa tabela.

Podem ser especificadas nas instruções:

- CREATE TABLE;
- ALTER TABLE.

Alguns tipos comuns de restrições incluem o seguinte:

- NOT NULL Constraint: Garante que uma coluna não pode ter o valor NULL.
- DEFAULT Constraint: Fornece um valor padrão para uma coluna quando nenhum é especificado.
- UNIQUE Constraint: Garante que todos os valores numa coluna são diferentes.
- CHECK Constraint: Garante que todos os valores numa coluna satisfazem um determinado critério.
- Primary Key Constraint: Utilizado para identificar de forma única uma linha na tabela.
- Foreign Key Constraint: Utilizado para garantir a integridade referencial dos dados.

LINGUAGEM SQL – DDL – RESTRIÇÕES

NOT NULL Constraint

Sintaxe:

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    Age int  
);
```

DEFAULT Constraint

Sintaxe:

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    City varchar(255) DEFAULT 'Sandnes'  
);
```

LINGUAGEM SQL — DDL — RESTRIÇÕES

UNIQUE Constraint

Sintaxe:

```
CREATE TABLE Persons (  
  ID int NOT NULL UNIQUE,  
  LastName varchar(255) NOT NULL,  
  FirstName varchar(255),  
  Age int  
);
```

CHECK Constraint

Sintaxe:

```
CREATE TABLE Persons (  
  ID int NOT NULL,  
  LastName varchar(255) NOT NULL,  
  FirstName varchar(255),  
  Age int CHECK (Age >= 18)  
);
```

LINGUAGEM SQL — DDL — RESTRIÇÕES

Chave Primária

Sintaxe:

```
CREATE TABLE Persons (  
  ID int NOT NULL PRIMARY KEY,  
  LastName varchar(255) NOT NULL,  
  FirstName varchar(255),  
  Age int  
);
```

Sintaxe:

```
ALTER TABLE Persons ADD PRIMARY KEY  
(SID);
```

LINGUAGEM SQL — DDL — RESTRIÇÕES

Chave Estrangeira / Externa

Sintaxe:

```
CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    FOREIGN KEY (PersonID) REFERENCES  
    Persons(PersonID)  
);
```

Sintaxe:

```
ALTER TABLE ORDERS  
ADD (CONSTRAINT fk_orders1) FOREIGN  
KEY (PersonID) REFERENCES  
Persons(PersonID);
```

LINGUAGEM SQL — DDL — CRIAR VISÃO

Sintaxe:

```
CREATE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

Exemplo:

```
CREATE VIEW [Products Above Average  
Price] AS  
SELECT ProductName, UnitPrice  
FROM Products  
WHERE UnitPrice > (SELECT AVG(UnitPrice)  
FROM Products);
```

```
SELECT * FROM [Products Above Average  
Price] ;
```

LINGUAGEM SQL

DML - Data Manipulation Language, ou Linguagem de Manipulação de Dados

LINGUAGEM SQL — DML — ALIASES

Sintaxe:

```
SELECT column_name AS alias_name  
FROM table_name;
```

Exemplo:

```
SELECT o.OrderID, o.OrderDate,  
c.CustomerName  
FROM Customers AS c, Orders AS o  
WHERE c.CustomerName="Around the  
Horn" AND c.CustomerID=o.CustomerID;
```

```
SELECT Orders.OrderID,  
Orders.OrderDate,  
Customers.CustomerName  
FROM Customers, Orders  
WHERE Customers.CustomerName="Aroun  
d the  
Horn" AND Customers.CustomerID=Orders.  
CustomerID;
```


LINGUAGEM SQL — DML — ALIASES

Sintaxe:

```
SELECT column_name AS alias_name  
FROM table_name;
```

```
SELECT column_name(s)  
FROM table_name AS alias_name;
```

Exemplo:

```
SELECT CustomerID as ID,  
CustomerName AS Customer  
FROM Customers;
```

```
SELECT CustomerName AS Customer,  
ContactName AS [Contact Person]  
FROM Customers;
```

```
SELECT CustomerName, Address + '  
' + PostalCode + ' ' + City + ', ' +  
Country AS Address  
FROM Customers;
```

LINGUAGEM SQL — DML — SELECT

Sintaxe:

```
SELECT column1, column2, ...  
FROM table_name;
```

```
SELECT * FROM table_name;
```

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```

Exemplo:

```
SELECT CustomerName,  
City FROM Customers;
```

```
SELECT * FROM Customers;
```

```
SELECT DISTINCT Country FROM Custome  
rs;
```

LINGUAGEM SQL – DML – WHERE

Sintaxe:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

Operadores

- = Igual
- <> Não é igual. (! =)
- > Maior do que
- < Menos do que
- >= Maior ou igual
- <= Menor ou igual
- BETWEEN Entre um intervalo inclusivo
- LIKE Procurar um padrão
- IN Para especificar vários valores possíveis para uma coluna

Exemplo:

```
SELECT * FROM Customers  
WHERE Country='Mexico';
```

```
SELECT * FROM Customers  
WHERE Country LIKE '%Mexico%';
```

```
SELECT * FROM Customers  
WHERE CustomerID=1;
```

LINGUAGEM SQL – DML – AND, OR AND NOT

Sintaxe:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE column LIKE pattern;
```

- WHERE CustomerName COMO 'a%'
 - Localiza valores que começam com "a"
- WHERE CustomerName COMO '% a'
 - Encontra valores que terminam com "a"
- WHERE CustomerName LIKE '%ou%'
 - Localiza quaisquer valores que tenham "ou" em qualquer posição
- WHERE ContactName LIKE 'a%o'
 - Localiza qualquer valor que comece com "a" e termina com "o"

Exemplo:

```
SELECT * FROM Customers  
WHERE CustomerName LIKE 'a%';
```

```
SELECT * FROM Customers  
WHERE City LIKE '[bsp]%';
```

```
SELECT * FROM Customers  
WHERE City LIKE 'L_n_on';
```

LINGUAGEM SQL – DML – IN

Sintaxe:

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1, value
2, ...);
```

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (SELECT STATE
MENT);
```

Exemplo:

```
SELECT * FROM Customers
WHERE Country IN ('Germany', 'France', 'UK
');
```

```
SELECT * FROM Customers
WHERE Country NOT IN ('Germany', 'France
', 'UK');
```

```
SELECT * FROM Customers
WHERE Country IN (SELECT Country FROM
Suppliers);
```

LINGUAGEM SQL — DML — BETWEEN

Sintaxe:

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name BETWEEN value1  
AND value2;
```

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name IN (SELECT STATE  
MENT);
```

Exemplo:

```
SELECT * FROM Products  
WHERE Price BETWEEN 10 AND 20;
```

```
SELECT * FROM Products  
WHERE Price NOT BETWEEN 10 AND 20;
```

```
SELECT * FROM Products  
WHERE ProductName BETWEEN 'Carnarvon  
Tigers' AND 'Mozzarella di Giovanni'  
ORDER BY ProductName;
```

```
SELECT * FROM Orders  
WHERE OrderDate BETWEEN '07/04/199  
6' AND '07/09/1996';
```

LINGUAGEM SQL – DML – AND, OR AND NOT

Sintaxe:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 AND condition2  
AND condition3 ...;
```

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 OR condition2 OR c  
ondition3 ...;
```

```
SELECT column1, column2, ...  
FROM table_name  
WHERE NOT condition;
```

Exemplo:

```
SELECT * FROM Customers  
WHERE Country='Germany' AND City='Berl  
in';
```

```
SELECT * FROM Customers  
WHERE City='Berlin' OR City='München';
```

```
SELECT * FROM Customers  
WHERE NOT Country='Germany';
```

```
SELECT * FROM Customers  
WHERE Country='Germany' AND (City='Ber  
lin' OR City='München');
```

LINGUAGEM SQL — DML — VALOR NULO

Sintaxe:

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NULL;
```

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NOT NULL;
```

Exemplo:

```
SELECT LastName, FirstName,  
Address FROM Persons  
WHERE Address IS NULL;
```

```
SELECT LastName, FirstName,  
Address FROM Persons  
WHERE Address IS NOT NULL;
```


LINGUAGEM SQL — DML — ORDER BY

Sintaxe:

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2,  
... ASC|DESC;
```

Exemplo:

```
SELECT * FROM Customers  
ORDER BY Country;
```

```
SELECT * FROM Customers  
ORDER BY Country DESC;
```

```
SELECT * FROM Customers  
ORDER BY Country, CustomerName;
```

```
SELECT * FROM Customers  
ORDER BY Country ASC,  
CustomerName DESC;
```

LINGUAGEM SQL — DML — SELECT TOP

Sintaxe:

```
SELECT TOP number | percent column_name(s)  
FROM table_name  
WHERE condition;
```

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
LIMIT number;
```

Exemplo:

```
SELECT TOP 3 * FROM Customers;
```

```
SELECT * FROM Customers  
LIMIT 3;
```

LINGUAGEM SQL – DML

COUNT(), AVG() , SUM()

Sintaxe:

```
SELECT COUNT(column_name)  
FROM table_name  
WHERE condition;
```

```
SELECT AVG(column_name)  
FROM table_name  
WHERE condition;
```

```
SELECT SUM(column_name)  
FROM table_name  
WHERE condition;
```

Exemplo:

```
SELECT COUNT(ProductID)  
FROM Products;
```

```
SELECT AVG(Price)  
FROM Products;
```

```
SELECT SUM(Quantity)  
FROM OrderDetails;
```

LINGUAGEM SQL – DML

COUNT(), AVG() , SUM()

Sintaxe:

```
SELECT COUNT(column_name)  
FROM table_name  
WHERE condition;
```

```
SELECT AVG(column_name)  
FROM table_name  
WHERE condition;
```

```
SELECT SUM(column_name)  
FROM table_name  
WHERE condition;
```

Exemplo:

```
SELECT COUNT(ProductID)  
FROM Products;
```

```
SELECT AVG(Price)  
FROM Products;
```

```
SELECT SUM(Quantity)  
FROM OrderDetails;
```

LINGUAGEM SQL – DML – JOIN

Sintaxe:

```
SELECT Orders.OrderID,  
Customers.CustomerName,  
Orders.OrderDate  
FROM Orders  
INNER JOIN Customers ON Orders.Cus  
tomerID=Customers.CustomerID;
```

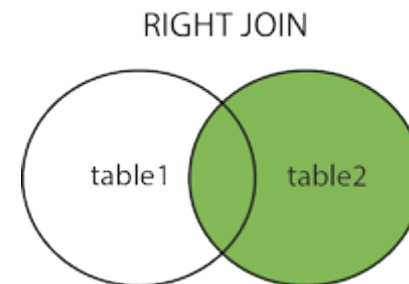
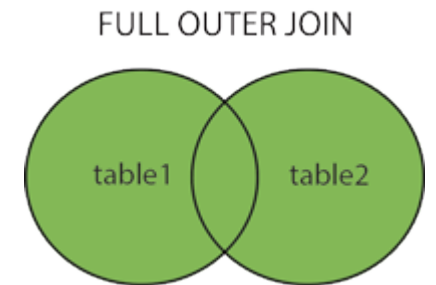
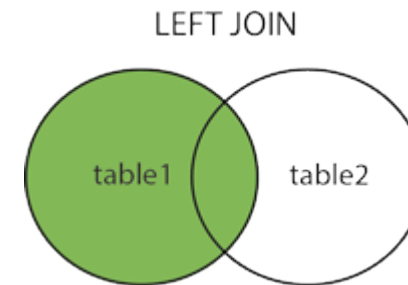
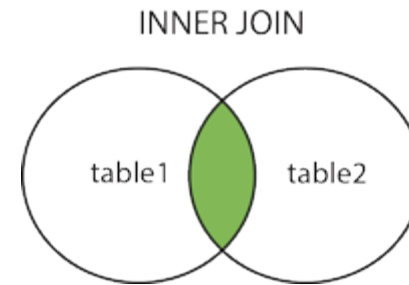
Exemplo:

```
SELECT Orders.OrderID,  
Customers.CustomerName,  
Orders.OrderDate  
FROM Orders  
INNER JOIN Customers ON Orders.Custome  
rID=Customers.CustomerID;
```

LINGUAGEM SQL – DML – JOIN

Tipos de Joins:

- (INNER) JOIN: Retorna registros que têm valores correspondentes em ambas as tabelas;
- LEFT (OUTER) JOIN: retorna todos os registros da tabela à esquerda e os registros correspondentes da tabela direita;
- RIGHT (OUTER) JOIN: retornar todos os registros da tabela direita e os registros correspondentes da tabela da esquerda;
- FULL (OUTER) JOIN: Retorna todos os registros quando há uma correspondência na tabela esquerda ou direita



LINGUAGEM SQL – DML – INNER JOIN

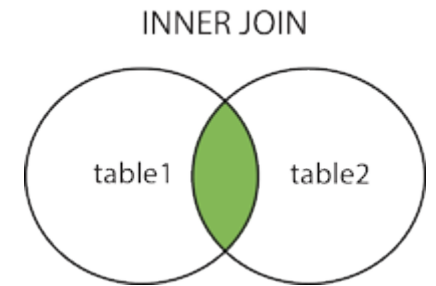
Sintaxe:

```
SELECT column_name(s)
FROM table1
INNER JOIN table2 ON table1.column_name = table2.column_name;
```

Exemplo

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```

```
SELECT Orders.OrderID, Customers.CustomerName, Shippers.ShipperName
FROM ((Orders
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID)
INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID);
```



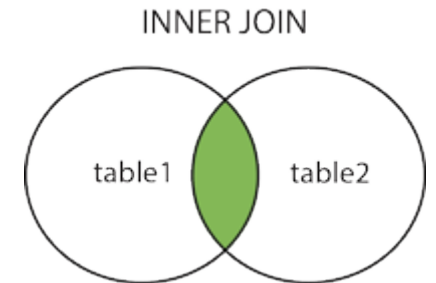
LINGUAGEM SQL — DML — SELF JOIN

Sintaxe:

```
SELECT column_name(s)  
FROM table1 T1, table1 T2  
WHERE condition;
```

Exemplo

```
SELECT A.CustomerName AS CustomerName1, B.CustomerName AS CustomerName2, A.City  
FROM Customers A, Customers B  
WHERE A.CustomerID <> B.CustomerID  
AND A.City = B.City  
ORDER BY A.City;
```



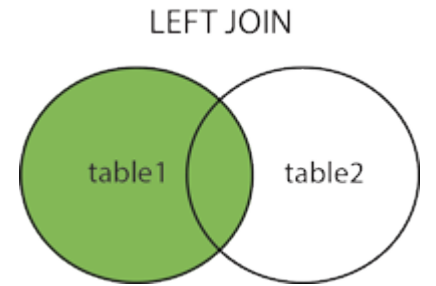
LINGUAGEM SQL – DML – LEFT JOIN

Sintaxe:

```
SELECT column_name(s)  
FROM table1  
LEFT JOIN table2 ON table1.column_name = table2.column_name;
```

Exemplo

```
SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers  
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID  
ORDER BY Customers.CustomerName;
```



LINGUAGEM SQL – DML – RIGHT JOIN

Sintaxe:

```
SELECT column_name(s)
```

```
FROM table1
```

```
RIGHT JOIN table2 ON table1.column_name = table2.column_name;
```

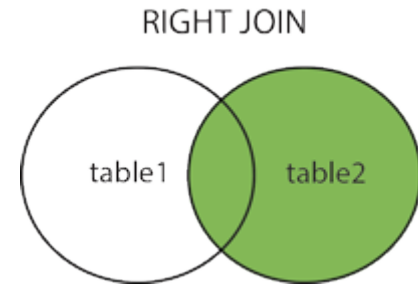
Exemplo

```
SELECT Orders.OrderID, Employees.LastName, Employees.FirstName
```

```
FROM Orders
```

```
RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID
```

```
ORDER BY Orders.OrderID;
```



LINGUAGEM SQL – DML – FULL JOIN

Sintaxe:

```
SELECT column_name(s)
```

```
FROM table1
```

```
FULL OUTER JOIN table2 ON table1.column_name = table2.column_name;
```

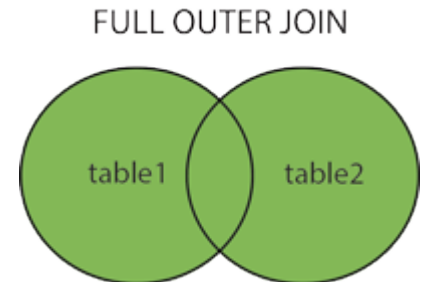
Exemplo

```
SELECT Customers.CustomerName, Orders.OrderID
```

```
FROM Customers
```

```
FULL OUTER JOIN Orders ON Customers.CustomerID=Orders.CustomerID
```

```
ORDER BY Customers.CustomerName;
```



LINGUAGEM SQL – DML – UNION

Restrições:

- Cada instrução SELECT no UNION deve ter o mesmo número de colunas
- As colunas também devem ter tipos de dados semelhantes
- As colunas em cada instrução SELECT também devem estar na mesma ordem

Sintaxe:

```
SELECT column_name(s) FROM table1  
UNION  
SELECT column_name(s) FROM table2;
```

Exemplo

```
SELECT column_name(s) FROM table1  
UNION ALL  
SELECT column_name(s) FROM table2;
```

Exemplo:

```
SELECT City FROM Customers  
UNION  
SELECT City FROM Suppliers  
ORDER BY City;
```

```
SELECT City FROM Customers  
UNION ALL  
SELECT City FROM Suppliers  
ORDER BY City;
```

```
SELECT City, Country FROM Customers  
WHERE Country='Germany'  
UNION  
SELECT City, Country FROM Suppliers  
WHERE Country='Germany'  
ORDER BY City;
```

```
SELECT 'Customer' As Type, ContactName, City,  
Country  
FROM Customers  
UNION  
SELECT 'Supplier', ContactName, City, Country  
FROM Suppliers;
```

LINGUAGEM SQL — DML — GROUP BY

- Usada com funções agregadas (COUNT, MAX, MIN, SUM, AVG) para agrupar o conjunto de resultados por uma ou mais colunas.

Sintaxe:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

Exemplo:

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;
```

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
ORDER BY COUNT(CustomerID) DESC;
```

```
SELECT Shippers.ShipperName, COUNT(Orders.
OrderID) AS NumberOfOrders FROM Orders
LEFT JOIN Shippers ON Orders.ShipperID =
Shippers.ShipperID
GROUP BY ShipperName;
```

LINGUAGEM SQL — DML — HAVING

Sintaxe:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

Exemplo:

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 5;
```

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 5
ORDER BY COUNT(CustomerID) DESC;
```

LINGUAGEM SQL — DML — EXISTS

Sintaxe:

```
SELECT column_name(s)
FROM table_name
WHERE EXISTS
(SELECT column_name FROM table_name WHERE
condition);
```

Exemplo:

```
SELECT SupplierName
FROM Suppliers
WHERE EXISTS (SELECT ProductName FROM
Products WHERE SupplierId =
Suppliers.supplierId AND Price < 20);
```

```
SELECT SupplierName
FROM Suppliers
WHERE EXISTS (SELECT ProductName FROM Pro
ducts WHERE SupplierId =
Suppliers.supplierId AND Price = 22);
```

LINGUAGEM SQL – DML – INSERT INTO

Sintaxe:

```
INSERT INTO table_name (column1,  
column2, column3, ...)
```

```
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO table_name
```

```
VALUES (value1, value2, value3, ...);
```

Exemplo:

```
INSERT INTO Customers (CustomerName,  
ContactName, Address, City, PostalCode,  
Country)
```

```
VALUES ('Cardinal', 'Tom B.  
Erichsen', 'Skagen  
21', 'Stavanger', '4006', 'Norway');
```


LINGUAGEM SQL — DML — SELECT INTO

Sintaxe:

```
SELECT *  
INTO newtable [IN externaldb]  
FROM oldtable  
WHERE condition;
```

```
SELECT column1, column2, column3, ...  
INTO newtable [IN externaldb]  
FROM oldtable  
WHERE condition;
```

Exemplo:

```
SELECT * INTO CustomersGermany  
FROM Customers  
WHERE Country = 'Germany';
```

```
SELECT CustomerName, ContactName INTO  
CustomersBackup2017  
FROM Customers;
```

LINGUAGEM SQL – DML – COMENTAR

Sintaxe:

Linha:

Texto depois de –

Exemplo:

--Select all:

```
SELECT * FROM Customers;
```

```
SELECT * FROM Customers -- WHERE City='Berlin';
```

```
--SELECT * FROM Customers;
```

```
SELECT * FROM Products;
```

Muitas linhas:

Texto entre /* */

```
/*Select all the columns  
of all the records  
in the Customers table:*/  
SELECT * FROM Customers;
```

LINGUAGEM SQL — DML — UPDATE

Sintaxe:

```
UPDATE table_name  
SET column1 = value1, column2 = value  
2, ...  
WHERE condition;
```

Exemplo:

```
UPDATE Customers  
SET ContactName='Juan'  
WHERE Country='Mexico';
```

```
UPDATE Customers  
SET ContactName='Juan';
```

LINGUAGEM SQL — DML — DELETE

Sintaxe:

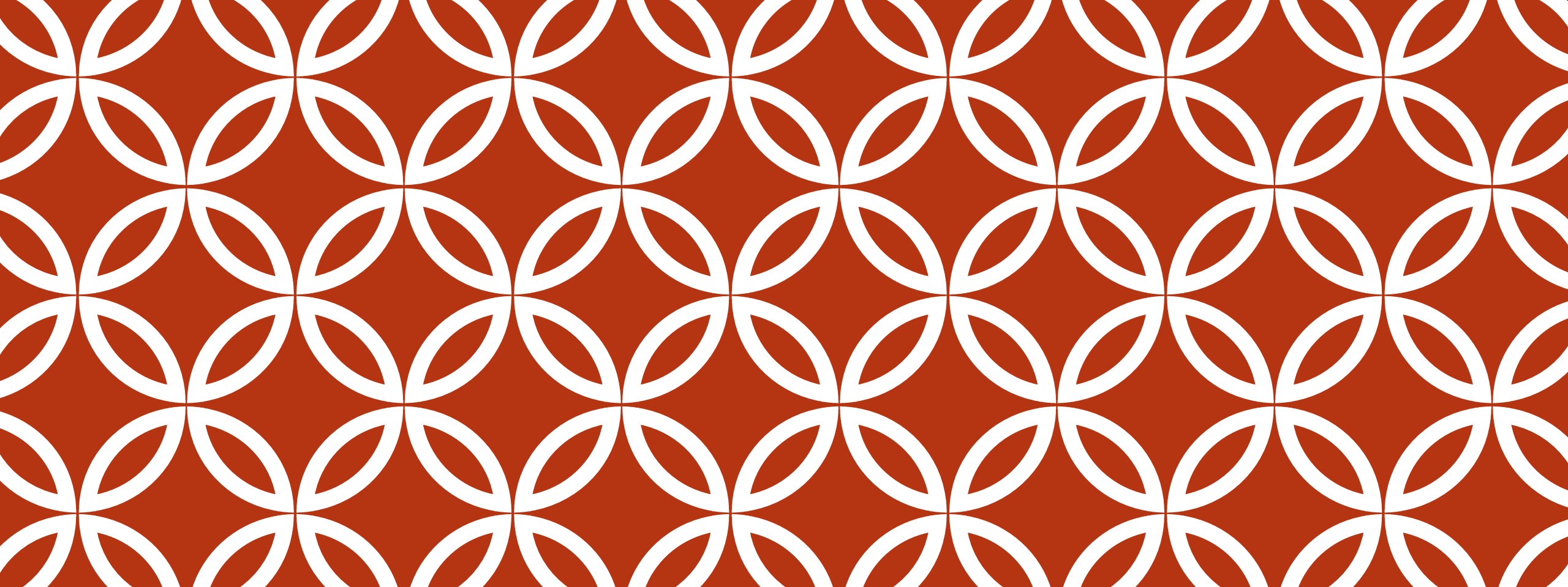
```
DELETE FROM table_name;
```

```
DELETE FROM table_name  
WHERE condition;
```

Exemplo:

```
DELETE FROM Customers  
WHERE CustomerName='Alfreds  
Futterkiste';
```

```
DELETE * FROM table_name;
```



BUSSINESS INTELIGENCE — BI E QLIKSENSE

AULA 10

AULA PRÁTICA

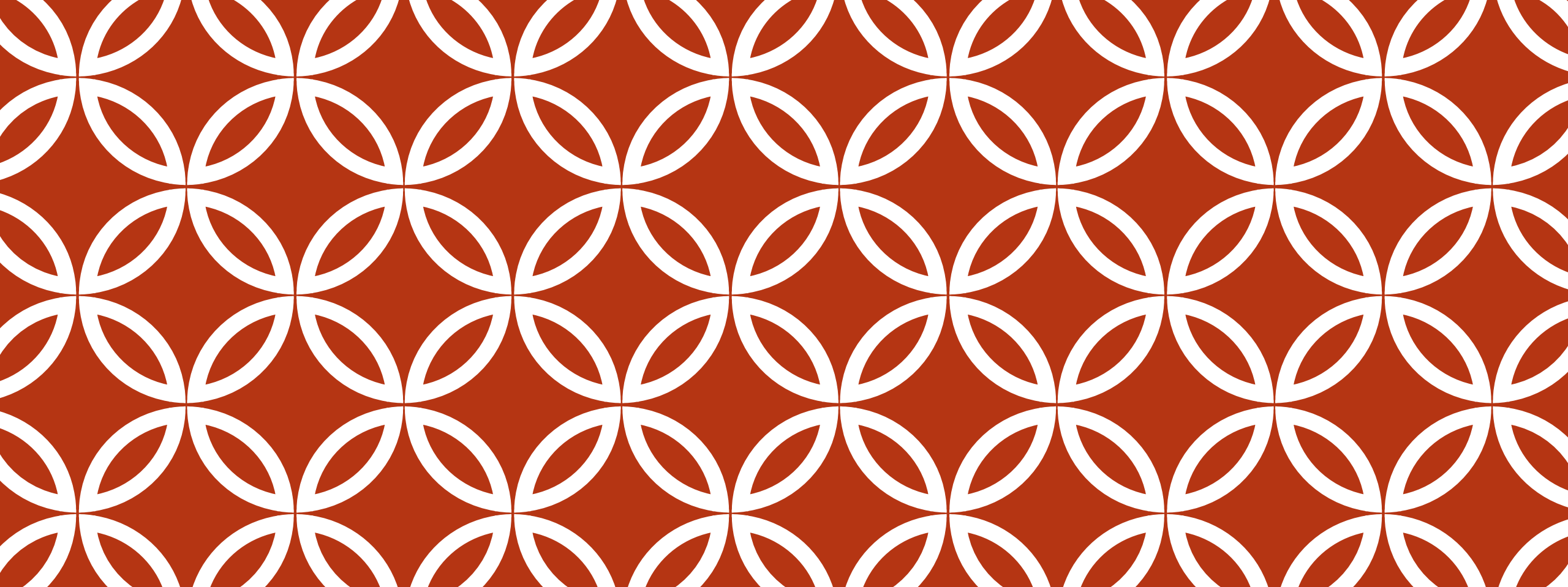
MANIPULAÇÃO DE DADOS

Realização de operações de manipulação de dados em bancos de dados utilizando linguagem SQL;

Manipulação de dados utilizando scripts QlikSense.

Definição de metadados (métricas e dimensões); e

Criação de visualizações de dados utilizando a ferramenta de *QlikSense Desktop*.



BUSSINESS INTELIGENCE — BI E QLIKSENSE

AULA 11

AULA PRÁTICA

ATIVIDADE DE FIXAÇÃO

Manipulação de dados utilizando scripts QlikSense.

Definição de metadados (métricas e dimensões); e

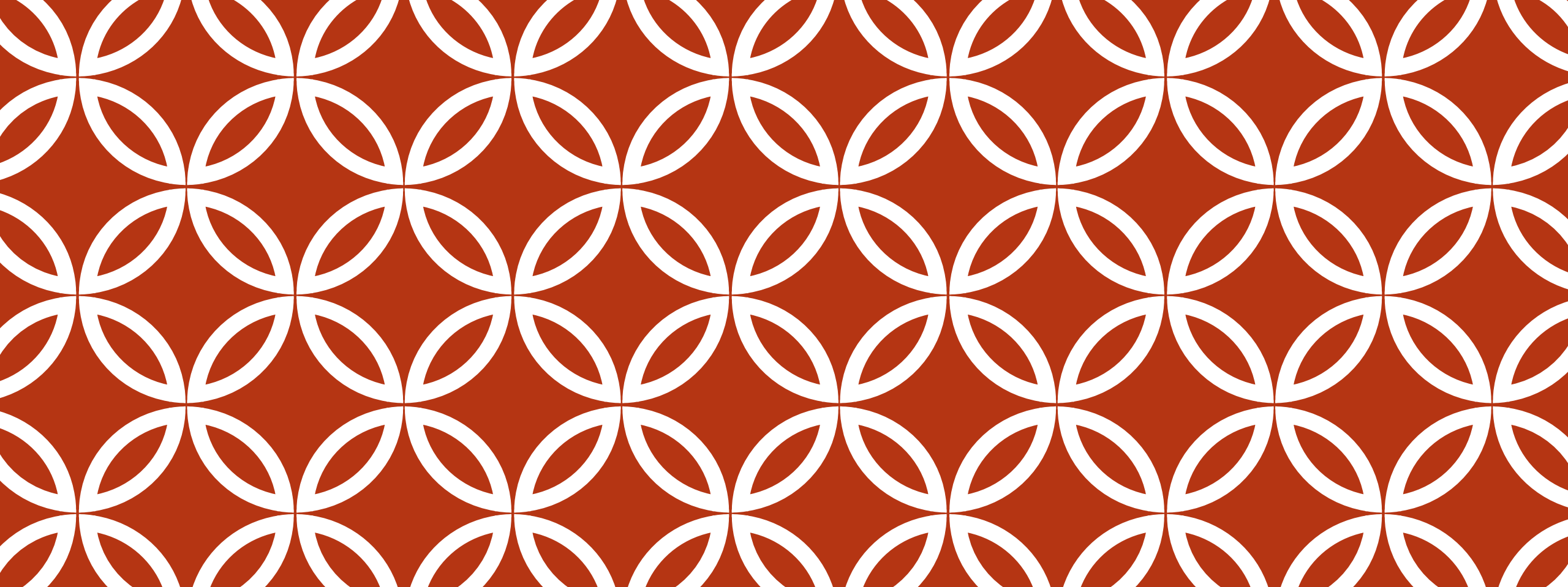
Criação de visualizações de dados utilizando a ferramenta de *QlikSense Desktop*.

Link do Painel com várias abordagens de cargas - Atualizado

<https://drive.google.com/file/d/1BRugClHOfp6x0xzrcrgeLBuTRegw-5r3/view?usp=sharing>

Link para arquivos de dados de municípios

<https://drive.google.com/file/d/1jebYW4RGxZJUMVfbYP4eF2B5t1jRJ5qk/view?usp=sharing>



BUSSINESS INTELIGENCE — BI E QLIKSENSE

AULA 12

AULA PRÁTICA

ATIVIDADE DE FIXAÇÃO

Atividade em dupla para desenvolvimento de um painel de BI com dados escolhidos pela dupla.