

# Introdução ao Data Warehouse

Geraldo Xexéo

14 de Novembro de 2019

Copyright © 2019 Geraldo Xexéo.

Todos os direitos reservados.

Esta é um pré-publicação com o texto parcial em trabalho.

Apenas os alunos da cadeira de Data Warehouse para Tomada de Decisão do DCC/IM/UFRJ estão autorizados a ter uma cópia digital e uma cópia impressa deste livro.

Contato com o autor pelo e-mail [xexeo@cos.ufrj.br](mailto:xexeo@cos.ufrj.br).

DRAFT

# Conteúdo

<b>Lista de Figuras</b>	vii
<b>Lista de Tabelas</b>	xiii
<b>Prefácio</b>	1
<b>I Assuntos Preparatórios</b>	3
<b>1 Sistemas de Informação</b>	5
1.1 Organizações . . . . .	5
1.2 Dados e informação . . . . .	6
1.3 Sistemas de Informação . . . . .	7
1.4 Características dos Sistemas de Informação . . . . .	9
1.5 Os SI e a Organização . . . . .	10
<b>2 Modelo de Entidades e Relacionamentos</b>	13
<b>3 Bancos de Dados Relacionais e SQL</b>	17
3.1 Definição Formal de um BD Relacional . . . . .	19
3.2 Sistemas Gerenciadores de Bancos de Dados . . . . .	21
3.3 SQL . . . . .	21
3.3.1 Data Definition Language . . . . .	22
3.3.2 Data Query Language . . . . .	24

3.3.3	Data Manipulation Language . . . . .	25
3.3.4	Data Transaction Language . . . . .	28
3.3.5	Data Control Language . . . . .	28
3.4	O que mais . . . . .	28
<b>4</b>	<b>OLAP</b>	<b>31</b>
4.1	O Cubo de Dados . . . . .	32
4.2	OLAP com Pivot Table no Excel <sup>TM</sup> . . . . .	33
4.3	Slice and Dice . . . . .	35
4.3.1	Filtros ( <i>Slice and Dice</i> ) no Pivot Table do Excel <sup>TM</sup> . .	38
4.4	Drill-down e Roll-up . . . . .	39
4.4.1	Drill down e Roll Up . . . . .	42
4.5	Pivot . . . . .	44
4.5.1	Pivot no Excel <sup>TM</sup> . . . . .	44
<b>5</b>	<b>Bases Exemplo</b>	<b>47</b>
5.1	O Bando de Dados Sakila . . . . .	47
5.2	O Banco de Dados Northwind . . . . .	52
5.3	A Cadeia de Valor da Northwind . . . . .	55
5.4	O Banco de Dados Adventure Works . . . . .	55
<b>II</b>	<b>Modelagem de Negócio</b>	<b>57</b>
<b>6</b>	<b>Modelagem de Negócio</b>	<b>59</b>
6.1	Processos de Negócio . . . . .	59
6.2	Analise de Exemplos de Processos . . . . .	61
6.3	Como são Representados Processos de Negócio . . . . .	62
<b>7</b>	<b>Organograma</b>	<b>63</b>
7.1	Relações em um organograma . . . . .	65

7.2	Outros formatos de organograma . . . . .	66
<b>8</b>	<b>Introdução ao Aris EPC</b>	<b>71</b>
8.1	Os softwares de desenho e as notações ARIS . . . . .	72
8.2	O Modelo Básico do EPC . . . . .	73
8.2.1	Atividades . . . . .	74
8.2.2	Eventos . . . . .	75
8.2.3	Regras . . . . .	77
8.2.4	Interface de Processo . . . . .	78
8.3	Exemplos Adicionais de EPC . . . . .	79
8.4	eEPC . . . . .	80
8.4.1	5W2H no EPC . . . . .	81
8.5	Passos para construir modelos EPC/EPCe[B23] . . . . .	83
8.5.1	EPC e Loops/Laços . . . . .	83
8.6	Deadlocks . . . . .	84
8.7	Algumas padrões em EPC . . . . .	85
8.8	Revisão das regras básicas do EPC . . . . .	88
<b>III</b>	<b>Data Warehouse</b>	<b>89</b>
<b>9</b>	<b>A Situação das Informações nas Organizações</b>	<b>91</b>
9.1	A necessidade de informação das organizações . . . . .	91
9.2	Processo de Informatização nas Organizações . . . . .	92
9.3	A necessidade centralização dos dados . . . . .	93
9.4	Momento Histórico da Proposta dos Data Warehouse . . . . .	96
9.5	O Data Warehouse . . . . .	96
9.6	Data Warehouse e Bancos de Dados . . . . .	98
9.7	Data Warehouses ainda são necessárias? . . . . .	99
9.8	Mais de um Data Warehouse? . . . . .	99

<b>10 O Que é um Data Warehouse</b>	<b>101</b>
10.1 O Debate Kimball x Inmon . . . . .	101
10.2 Definição de Inmon para Data Warehouse . . . . .	102
10.2.1 Orientação a Assunto . . . . .	102
10.3 Data Warehouse 2.0 . . . . .	104
10.4 Data Lake . . . . .	104
<b>11 O Método Beam*</b>	<b>105</b>
11.1 Eventos de Negócio OU Histórias de Dados . . . . .	108
11.2 Descobrindo as Histórias . . . . .	110
11.2.1 Prática para Levantamento e Seleção de Eventos . . . . .	111
11.3 Passo a Passo da Modelagem de Eventos de Negócio . . . . .	112
11.3.1 A sessão de criação da Tabela BEAM* . . . . .	114
11.3.2 Tabela BEAM* Inicial . . . . .	114
11.3.3 Modelando o quando . . . . .	115
11.3.4 Primeiros Exemplos . . . . .	116
11.3.5 Refazendo o ciclo “quem-o que -como” . . . . .	118
11.3.6 Definindo onde . . . . .	120
11.3.7 Encontrando quantos . . . . .	121
11.4 Definindo por que . . . . .	125
11.5 Definindo como . . . . .	126
11.5.1 Escolhendo a Granularidade . . . . .	126
11.5.2 Dando nome ao evento . . . . .	127
11.5.3 Resumo do passo a passo . . . . .	127
11.6 Modelando Dimensões . . . . .	128
11.6.1 Codificação Usada . . . . .	129
11.6.2 Início da tabela . . . . .	130
11.6.3 Granularidade e chaves . . . . .	131
11.6.4 Os Atributos de Dimensão . . . . .	131

11.6.5 Prevendo mudanças . . . . .	135
11.6.6 Construindo a hierarquia da dimensão . . . . .	135
11.7 Matriz de Evento . . . . .	137
<b>Índice</b>	<b>139</b>
<b>A Exercício Um</b>	<b>145</b>
A.1 Algumas questões . . . . .	147
A.2 Orientações . . . . .	148
A.3 Dicas de implementação . . . . .	148

DRAFT

# **Lista de Figuras**

1.1	Uma tela de um sistema de informações real . . . . .	8
1.2	Divisão do mercado de ERP mundial em 2013 (Columbus, 2014) . . . . .	11
2.1	Um diagrama de entidades e relacionamentos simples, sem mostrar atributos . . . . .	14
2.2	Um modelo de entidades e relacionamentos escrito na notação da Engenharia de Informação . . . . .	15
3.1	Exemplo de um banco de dados relacional simples . . . . .	18
3.2	Diagrama de entidades e eelacionamentos relativo ao banco de dados da Figura 3.1 . . . . .	18
4.1	Pequena fotografia de uma planilha com os micro-dados do ENEM 1998. Fonte (INEP, 2016). . . . .	32
4.2	Cubo OLAP simplificado para rede de lojas de calçado. . . . .	33
4.3	Diálogo do Pivot Table do Microsoft Excel <sup>TM</sup> 365. . . . .	34
4.4	Nesse quadro são controladas que colunas aparecem em que posição na pivot table . . . . .	34
4.5	Na planilha, a pivot table vai se alterando dinamicamente para atender a configuração dada no quadro de controle. . . . .	35
4.6	Escolhendo “Sum” como função de agregação de dados para a presença dos candidatos. . . . .	35
4.7	Efeito do uso de um filtro selecionando apenas os dados da cidade de São Paulo, realizando um <i>slice</i> . . . . .	36
4.8	Aplicando mais de um filtro, obtemos um <i>dice</i> . . . . .	36

4.9	Usando o filtro para cortar algumas idades . . . . .	38
4.10	Tabela de presenças no ENEM 1998 por estado para maiores de 17 anos. Fonte (INEP, 2016) . . . . .	39
4.11	Hierarquias de conceitos. No caso, na hierarquia de períodos de tempo, temos dois caminhos paralelos. . . . .	40
4.12	Fazendo um <i>drill down</i> se obtém um novo cubo mais detalhado. . . . .	41
4.13	Pequena fotografia de uma planilha com uma pivot table verificando as presenças por cidade e estado do ENEM 1998. Fonte (INEP, 2016). . . . .	42
4.14	Pivot table com algumas informações suprimidas por meio de um <i>roll up</i> nos estados de Acre, Alagoas, Amazonas, Amapá e Bahia. Fonte (INEP, 2016). . . . .	43
4.15	Operação de <i>pivot</i> no cubo OLAP troca a posição das dimensões. É necessário comparar com a Figura 4.2. . . . .	44
4.16	Posição dos dados antes da operação <i>pivot</i> . . . . .	45
4.17	Após o <i>pivot</i> . . . . .	45
5.1	O modelo de dados da base sakila . . . . .	48
5.2	O modelo de dados da base Sakila com os nomes de campo . .	50
5.3	O modelo de dados da base Sakila com nomes de campo e seus tipos . . . . .	51
5.4	O modelo de dados da base Northwind só com as entidades . .	53
5.5	O modelo de dados da base Northwind com nomes de campo .	53
5.6	O modelo de dados da base Northwind com nomes de campo e seus tipos . . . . .	54
5.7	O processo Pedir Produto da Northwind . . . . .	55
7.1	Um organograma simples, contendo apenas cargos, de uma empresa hipotética. . . . .	64
7.2	Um exemplo de um organograma de uma empresa hipotética escrito em ARIS. . . . .	65
7.3	A relação de subordinação, normalmente mantida na horizontal, entre o gerente geral e seus gerentes subordinados. . . . .	66

7.4	A relação de assessoria é normalmente feita com uma barra horizontal, como a Assessorial Internacional aparece no diagrama acima. . . . .	66
7.5	Um exemplo de organograma radial . . . . .	67
7.6	Organograma do Poder Judiciário, em forma inversa, criado pelo site Guia de Direitos, deixado em Copyleft. . . . .	68
8.1	Um processo de recepção e atendimento de pedido escrito em ARIS/EPC, desenhado no ARISExpress . . . . .	72
8.2	Formas de representação de uma atividade em diferentes versões do Aris. . . . .	75
8.3	Formas de representação de uma evento em diferentes versões do Aris. . . . .	76
8.4	Exemplos de eventos Segundo a versão original de EPCs, sempre deveria haver um evento entre dois processos. Atualmente é permitido que uma sequência de processos não tenha nenhum evento entre eles. Logo, os diagramas das figuras a seguir podem ser usados como equivalentes. . . . .	77
8.5	Um conector só deve ser utilizado na forma de divisor ou junção, e nunca das duas formas ao mesmo tempo. . . . .	78
8.6	Formas de representação de uma interface de processo em diferentes versões do Aris. . . . .	79
8.7	Indicando que um processo é continuação lógica de outro processo. No trecho final do “Processo 1” é indicado que ele é seguido pelo “Processo 2”, e nesse processo é indicado que o caminho é proveniente do “Processo 1”. . . . .	79
8.8	Processo simplificado de entrega de pizzas. . . . .	80
8.9	Elementos complementares dos diagramas eEPC. . . . .	81
8.10	Nova lista e imagens dos elementos complementares dos diagramas EPC no ARISExpress. . . . .	82
8.11	Exemplo de eEPC, a partir do EPC anterior da Figura 8.1. Desenho feito no ARISExpress. Os símbolos suplementares podem variar de acordo com a ferramenta case sendo usada. .	82

8.12 Um EPC válido, com um loop simples baseado em XOR. Também é dado um exemplo de como conectar esse EPC a um outro processo. . . . .	84
8.13 O modelo acima, demonstra um deadlock. Como comentário, os números identificam as setas e as letras identificam os conectores . No conector XOR (B), um dos caminhos 4 ou 5 será escolhido. Porém, se for escolhido o caminho 5, o sistema para de funcionar no conector E (D), pois esse E nunca será satis- feito (já que 4-8-12 não será percorrido). Caso seja escolhido o caminho 4, o conector E (C) nunca será satisfeito (pois 5-7-9 não será percorrido), e o sistema vai parar em (C), esperando 9, e (D), esperando 11. . . . .	85
8.14 Descrição EPC para “Se X então Fazer A, Se Y, então Fazer B”	86
8.15 Caminhos paralelos, na notação do ARISExpress . . . . .	87
8.16 Descrição EPC para “Se Não X então Fazer A” . . . . .	88
9.1 Diferentes cargas exigidas de sistemas OLTP e OLAP. Fonte: (Inmon, 2005) . . . . .	97
9.2 Número de data warehouses por empresa(Wells e Nahari, 2019). . . . . .	100
10.1 A cadeia de valor de um mercado, descrição em ARIS. . . . .	102
10.2 Diversas fontes de onde pode ser obtido o assunto cliente . . . . .	104
11.1 Descrição gráfica do <i>Framework 7W</i> . Fonte:(Corr e Stagnitto, 2012) . . . . .	106
11.2 Descrição gráfica do Modelstorming. Fonte:(Corr e Stagnitto, 2012) . . . . .	107
11.3 Descrição do Modelstorming em ARIS-EPC . . . . .	108
11.4 Primeiro passo da ordem de descoberta. Fonte:(Corr e Stag- nitto, 2012) . . . . .	110
11.5 O processo Modelar Detalhes de Evento, descrito em ARIS-EPC	113
11.6 Definir Quem-O Que-Quando, descrito em ARIS-EPC . . . . .	113

11.7 Configuração inicial da tabela BEAM* para o evento Cliente Encomenda Produto. É necessário deixar bastante espaço para continuar a atividade. Adaptado de:(Corr e Stagnitto, 2012) . . . . .	115
11.8 Ordem de descoberta das informações do modelo. Fonte:(Corr e Stagnitto, 2012) . . . . .	116
11.9 Configuração da tabela BEAM* com o o detalhe “quando”. Aparece a preposição utilizada (em+a) e um sustantivo. Adaptado de:(Corr e Stagnitto, 2012) . . . . .	117
11.10Configuração da tabela BEAM* com o primeiro exemplo, mostrando uma história típica. Adaptado de:(Corr e Stagnitto, 2012) . . . . .	118
11.11Configuração da tabela BEAM* com o segundo exemplo, mostrando uma outra história típica, diferente da primeira. Adaptado de:(Corr e Stagnitto, 2012) . . . . .	119
11.12Configuração da tabela BEAM* com um exemplo de repetição de um item. Adaptado de:(Corr e Stagnitto, 2012) . . . . .	120
11.13Configuração da tabela BEAM* com um dados faltando. Adaptado de:(Corr e Stagnitto, 2012) . . . . .	121
11.14Configuração da tabela BEAM* ao final deste passo. Adaptado de:(Corr e Stagnitto, 2012) . . . . .	122
11.15Os três primeiros passos do processo podem ser repetidos para exaurir os conceitos de “quem” e “quanto”. Adaptado de:(Corr e Stagnitto, 2012) . . . . .	122
11.16Uma tabela BEAM* com dois “quem” e dois “quando”. (Corr e Stagnitto, 2012) . . . . .	123
11.17Uma tabela BEAM* descreve uma história que acontece em dois lugares, um em cada ponto no tempo. Por espaço algumas colunas já encontradas foram suprimidas. Fonte:(Corr e Stagnitto, 2012) . . . . .	123
11.18Introduzindo na tabela BEAM* as informações quantitativas. Por espaço algumas colunas já encontradas foram suprimidas. Fonte:(Corr e Stagnitto, 2012) . . . . .	124
11.19Introduzindo na tabela BEAM* as informações sobre por que. Por espaço algumas colunas já encontradas foram suprimidas. Fonte:(Corr e Stagnitto, 2012) . . . . .	125

11.20	Introduzindo na tabela BEAM* as informações sobre como. Por espaço algumas colunas já encontradas foram suprimidas. Fonte:(Corr e Stagnitto, 2012) . . . . .	126
11.21	Finalizando a Tabela BEAM* com nome e tipo. Fonte:(Corr e Stagnitto, 2012) . . . . .	128
11.22	Modelar Dimensão descrito em ARIS-EPC. . . . .	129
11.23	Início da definição da dimensão Cliente. Fonte: (Corr e Stag- nitto, 2012) . . . . .	131
11.24	Definição da chave de negócios para a dimensão Cliente. Fonte: (Corr e Stagnitto, 2012) . . . . .	132
11.25	Definição de um tipo para a Dimensão Cliente. Fonte: (Corr e Stagnitto, 2012) . . . . .	133
11.26	Definição dos atributos exclusivos. Fonte: (Corr e Stagnitto, 2012) . . . . .	135
11.27	Tipos de hierarquias para dimensões. Fonte: (Corr e Stag- nitto, 2012) . . . . .	136
11.28	Hierarquias para dimensões de data, com duas versões, e pro- duto. Fonte: (Corr e Stagnitto, 2012) . . . . .	136
11.29	Notações para hierarquias com loops. Fonte: (Corr e Stag- nitto, 2012) . . . . .	137
11.30	Notações para consultas perpassando hierarquias. Fonte: (Corr e Stagnitto, 2012) . . . . .	137
11.31	Exemplo de Matriz de Evento. Fonte: (Corr e Stagnitto, 2012)	138
A.1	Modelo estrela para a Data Warehouse do exercício . . . . .	146

# **Lista de Tabelas**

3.1	Dois tipos de domínio para o mesmo conceito . . . . .	20
4.1	Presenças por moradores dos estados no ENEM de 1998, Fonte: (INEP, 2016). . . . .	37
4.2	Presenças por moradores dos estados no ENEM de 1998, Fonte: (INEP, 2016). . . . .	38
4.3	Presenças por moradores das cidades do Acre no ENEM de 1998. Fonte: (INEP, 2016) . . . . .	41
8.1	Objetos básicos do EPC, em várias versões do Aris. . . . .	73
10.1	Sistemas em um pequeno mercado. . . . .	102
11.1	Equivalência dos tipos de eventos do método BEAM* e os Tipos de Tabela Fato de Kimball. Fonte: Corr e Stagnitto, 2012110	
11.2	Tipos de eventos e seus códigos. Baseado em:(Corr e Stag- nitto, 2012) . . . . .	127
11.3	Códigos usados na modelagem dos campos da dimensão . . . . .	130
A.1	Relação entre as tabelas do DW e as tabelas da base Sakila. .	145
A.2	Campos a serem calculados . . . . .	147

DRAFT

# Prefácio

Neste livro, a teoria de Inmon é usada, na maior parte, para explicar o papel do data warehouse na empresa e características do processo de Data Warehousing, enquanto as práticas de Kimball, e outras nele baseadas, são usadas para explicar em mais detalhes como construí-los.

Sinceramente, a adoção de uma ou de outra obra em partes deste texto é claramente uma preferência do autor, baseada no entendimento que os métodos de Kimball, em especial a Modelagem Dimensional, e os deles derivados, geram resultados mais imediatos e que isso é um grande fator de sucesso de projeto.

Como comentário em benefício do leitor, o livro mais famoso de Inmon, *Building the Data Warehouse*, apresenta uma abordagem mais de comentar as características importantes do processo, o que torna difícil seu uso imediato para o desenvolvimento de um projeto, enquanto os livros de Ralph Kimball, *The Data Warehouse Toolkit*, e de Kimball et al., *The Data Warehouse Lifecycle Toolkit: Practical Techniques for Building Data Warehouse and Business Intelligence Systems* dão uma abordagem mais objetiva para desenvolver um Data Warehouse, melhor definindo o passo a passo a ser feito. Ambos os autores possuem outras publicações que podem auxiliar no processo.

Ambos os autores possuem sites bastante interessantes:

- <https://www.kimballgroup.com/>
- <https://www.forestrimtech.com/>

DRAFT

**Parte I**

**Assuntos Preparatórios**

DRAFT

# Capítulo 1

## Sistemas de Informação

O objetivo desse capítulo é contextualizar o Data Warehouse como um Sistema de Informação utilizado dentro da empresa e a importância do seu uso na tomada de decisão.

Para isso, ele apresenta uma breve descrição de como funcionam, para que servem e quem usa os sistemas de informação dentro de uma organização.

Ao longo deste livro a palavra **organização** representa empresas, órgãos públicos, entidades benéficas, associações e qualquer outra forma de instituição, ou parte de uma instituição, com objetivos definidos e que pode obter algum benefício com o uso de sistemas de informação. Nisso está incluído um enorme espectro de interesses e tamanhos, tanto um consultório dentário quanto uma multinacional de bebidas.

### 1.1 Organizações

Neste texto o termo organização é usado para representar, de forma geral, todas as pessoas jurídicas e suas subdivisões. Empresas, departamentos e seções, associações, órgãos governamentais, clubes, igrejas, etc. Uma forma melhor de dizer é que queremos discutir sobre todas as pessoas **menos** as pessoas naturais ou pessoas físicas.

Se mesmo uma pessoa física já precisa de sistemas de informações, na forma de agenda, caderno de telefone, uso de planilhas eletrônicas para calcular os gastos do mês, mais ainda se pode esperar de uma organização. Isso acontece por vários motivos, entre eles a necessidade de manter uma memória, a de guardar informações para seu funcionamento, obrigações impostas pelo

governo e outras.

Uma organização tem um propósito de ser. Para as mais complexas ou mais bem gerenciadas, este propósito é definido em um planejamento estratégico que conta com a definição de uma missão, uma visão, metas e objetivos. Muitas vezes são definidos *KPI - Key Performance Indicators*, que permitem medir o desempenho da organização em relação ao que foi planejado.

Quanto maior a organização, maior a quantidade de dados que ela gera. Estes dados vêm tanto da interação com o mundo externo, quanto da comunicação e gestão interna. Esses dados representam informações que explicam de várias formas o que é a organização e qual o seu desempenho. Para manipular esses dados são usados os **Sistemas de Informação**.

Assim, a organização precisa que os dados estejam disponíveis para se conhecer. Ao mesmo tempo, ela precisa que esses dados sejam de boa qualidade.

Um **Data Warehouse** é um tipo de sistema de informação, e é também uma, entre várias, ferramentas importante para garantir a qualidade desse dados.

## 1.2 Dados e informação

Antes de entender o que é um Sistema de Informação, é preciso entender melhor o que significa a palavra Informação.

Segundo o *American Heritage*, **informação** é o **dado** quando processado, guardado ou transmitido. Já no dicionário Aurélio, **informação**, entre outros significados, pode ser “Conhecimento amplo e bem fundamentado, resultante da análise e combinação de vários informes”, “Coleção de fatos ou de outros dados fornecidos à máquina, a fim de se objetivar um processamento”, ou ainda “Segundo a teoria da informação, medida da redução da incerteza, sobre um determinado estado de coisas, por intermédio de uma mensagem”. Esta última definição, apesar de originária da Teoria da Informação, dá uma característica importante, que tem impacto em processos de decisão.

É aconselhável, no contexto da Ciência da Computação, criar uma diferenciação entre dados e informação, mesmo que as palavras possam ser consideradas sinônimas em muitos contextos, sendo usualmente confundidas ou utilizadas de forma intercambiável no uso corrente. Dessa forma, elas podem ser mais bem entendidas e utilizadas.

**Dados** são apenas os símbolos usados para representar alguma informação, o registro de diferentes aspectos de um fato ou fenômeno. Os números ou letras que guardados em um banco de dados são, como diz o nome, “dados”. Dados não são interpretados, eles existem, são adquiridos de alguma forma, via coleta, pesquisa ou criação, guardados de outra forma e, possivelmente, apresentados em uma terceira. O **computador** é uma máquina programável de uso geral que manipula dados.

Por outro lado, **informação** é o dado com significado, normalmente processado de forma a ser útil. Uma informação deve permitir responder perguntas como “quando”, “quanto”, “quem”, “qual” e “onde” sobre alguma coisa.

$$\text{Informação} = \text{Dado} + \text{Significado} \quad (1.1)$$

É sempre necessário fazer um mapeamento entre dados e informação. Esse mapeamento pode ser simples ou complexo, dependendo de várias variáveis envolvidas, que vão desde decisões arbitrárias tomadas pelo desenvolvedor até padrões internacionais. Por exemplo, em muitos sistemas é preciso ter a informação do sexo de uma pessoa (masculino ou feminino). Para isso, são guardados um número (1 ou 0) ou uma letra (M ou F) que é o dado que faz a indicação da informação.

### 1.3 Sistemas de Informação

**Sistemas de Informação** são utilizados em organizações para planejamento, monitoração, comunicação e controle das suas atividades, por meio da manipulação e guarda de informações.

Segundo o Dicionário Aurélio, a palavra **sistema** significa, entre outras coisas, um “Conjunto particular de instrumentos e convenções adotados com o fim de dar uma informação”. Os instrumentos são as ferramentas, os mecanismos, concretos ou abstratos, que utilizamos para fazer funcionar os sistemas, tais como: programas de computador, relatórios, formulários, etc. As convenções são as suas regras de utilização.

Um exemplo típico de sistema de informação é um sistema de aluguel de automóveis. Entre suas várias finalidades, a principal é certamente controlar o aluguel dos veículos, informando quem está com qual veículo em um determinado momento (quando), e quanto deve pagar por isso. Além disso, o sistema permite outras atividades, como a gerência do veículos disponíveis,

a monitoração dos modelos mais e menos alugados, etc.

Outro exemplo de sistema de informações é um *Dashboard* estratégico que informa o valor dos *KPI*, *Key Performance Indicators*, da organização, permitindo um diagnóstico de seu funcionamento e a verificação de sua aderência ao planejamento estratégico.

A Figura 1.1 mostra uma tela de um sistema real (BDO) que gerencia uma frota de caminhões. Nesta tela são informados os dados sobre uma viagem do caminhão.

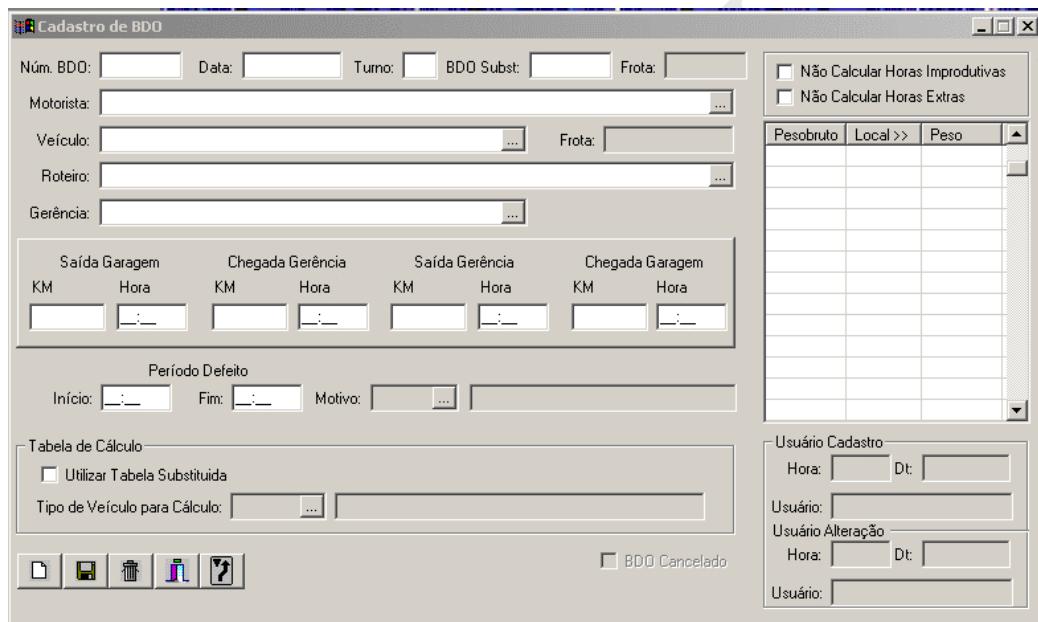


Figura 1.1: Uma tela de um sistema de informações real

Apesar de atualmente o termo ser mais usado em relação a **sistemas de informação automatizados**, implementados na forma de programas de computador, isso não é uma necessidade. Durante séculos as organizações usaram sistemas de informação apenas com o uso de pessoas, papel e tinta. Apenas bem mais tarde, apareceram máquinas como máquinas de escrever e de somar. Não seria exagerado dizer que a escrita e os números foram criados para suportar os primeiros sistemas de informação, que tratavam, por exemplo, de colheitas e comércio.

K. Laudon e J. Laudon (2011) fornecem uma definição formal para um **Sistema de Informação**: “Conjunto de componentes inter-relacionados, que coletam (ou recuperam), processam, armazenam e distribuem informações destinadas a apoiar a tomada de decisões, a coordenação e o controle de uma

organização”

Uma organização possui muitos sistemas de informação, integrados ou não. Mesmo uma pequena empresa caseira pode ser gerenciada com várias planilhas, enquanto uma grande multi-nacional vai certamente possuir de um grande sistema de *Enterprise Resource Planning* (ERP), que é o nome dado a sistemas de gestão empresarial, e mais uma miríade de outros sistemas, entre eles um **Data Warehouse** e vários **Data Marts**. Esses sistemas acabam por compor um ecossistema extremamente complexo, com várias interações e que apresenta diversos desafios ao seu uso e sua gerência.

Todos esses sistemas de informação, conectados ou isolados, dentro de uma organização, guardam uma quantidade enorme de dados que podem ser usados para monitoração, controle e tomada de decisão. Os dados, porém, podem conter erros, incertezas, repetições, e outros problemas que tornam necessário uma prática muito cuidadosa para seu uso.

## 1.4 Características dos Sistemas de Informação

É importante entender que sistemas de informação tradicionais são sistemas **interativos e reativos**.

**Interativo** significa que o sistema troca informações com o ambiente, em especial com os agentes externos que fazem parte desse ambiente, pessoas e outros sistemas de computador. O sistema só faz sentido se é capaz dessa interação.

**Reativo** significa que o sistema funciona reagindo a mudanças no ambiente, e em especial, a mudanças provocadas pelos agentes externos.

Sendo iterativo e reativo, um sistema de informações atende a comandos e pedidos que cruzam sua fronteira com o ambiente, processa dados que vêm desse ambiente, guarda esses dados e depois os envia, processados, ao ambiente.

Esses sistemas também são sistemas de **respostas planejadas**. Isso significa que as respostas são determinadas, na prática por algoritmos, e que é possível criar um programa que as produza. Também significa que todas as perguntas que podem ser feitas ao sistema podem, e são, identificadas previamente, sendo ele limitado em escopo.

Apesar de existir uma classe de sistema de informação que está ocupando cada vez mais espaço nas empresas, baseados em técnicas de Inteligência Ar-

tificial, que possuem algumas características além dos sistemas tradicionais, como aprender a partir de dados antigos e novos e responder a perguntas de formas não previstas, ou pelo menos envolvendo incerteza. A criação, manutenção e gerência desses sistemas trazem também novos desafios a organização. Porém, as respostas ainda são planejadas, pelo menos em largo espectro.

## 1.5 Os SI e a Organização

Atualmente vários sistemas de informações típicos de uma empresa são necessidades básicas que podem ser atendidas de uma só vez. Esses sistemas constroem o que é comumente chamado de ERP – de *Enterprise Resource Planning* – ou **Sistemas de Gestão Empresarial** em português – mas que na prática não são sistemas de planejamento (ou de recursos), mas sim de controle e administração de uma empresa.

Entre as características encontradas em ERPs podemos citar a integração das atividades da empresa e o uso de um banco de dados único. O líder mundial do mercado é a SAP AG, com o produto SAP R/3. O custo de implantação de um ERP de grande porte pode chegar até 300 milhões de dólares. No Brasil, existem produtos menos ambiciosos e mais baratos. A Figura 1.2 mostra a divisão do mercado de ERP em 2013 (Columbus, 2014).

Os sistemas de ERP atuais contêm módulos representando os mais típicos sistemas de informações necessários em uma empresa, tais como: Contabilidade Fiscal, Contabilidade Gerencial, Orçamento e Execução Orçamentária, Ativo Fixo, Caixa e bancos, Fluxo de Caixa, Aplicações e Empréstimos, Contas a Receber, Contas a Pagar, Controle de Viagens, Controle de Inadimplência, Administração dos preços de venda, Compras, Controle de fretes, Controle de contratos, Controle de investimentos, Cotações de vendas, Estoque, Exportação, Faturamento, Gerenciamento de armazéns, Importação, Obrigações fiscais, Pedidos, Previsão de vendas, Recebimento, Gestão de informação de RH, Pagadoria, Treinamento, RH scorecard, Planejamento de RH, Planejamento de produção, Planejamento da capacidade, Custos industriais, Controle de chão de fábrica, Controle da produção, Configurador de produtos, Planejamento de Manutenção, Acompanhamento de Manutenção e ainda muitos outros.

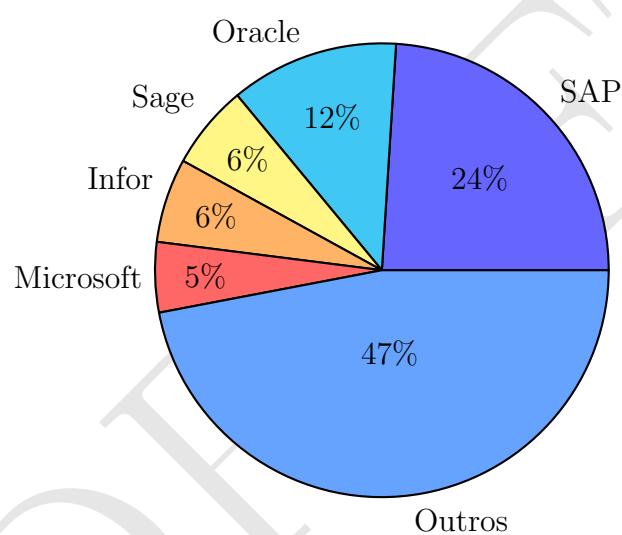


Figura 1.2: Divisão do mercado de ERP mundial em 2013 (Columbus, 2014)

DRAFT

## Capítulo 2

# Modelo de Entidades e Relacionamentos

O **Modelo de Entidades Relacionamentos** (MER), segundo Cougo (1999), descreve o mundo como: "...cheio de coisas que possuem características próprias e que se relacionam entre si". Essas descrições são feitas sobre mundos restritos, algumas vezes chamados domínios, e se referem a informação necessária para que um sistema de informações cumpra suas funções. Normalmente a descrição é feita na forma de um diagrama, o diagrama de entidades e relacionamentos.

A Figura 2.1 mostra um diagrama de entidades e relacionamentos simples escrito. Esse diagrama, que descreve um mini-mundo sobre novelas, mostra as entidades Novela, Capítulo, Ator, Diretor, Ator Horista e Horas. Ele também apresenta os relacionamentos Dirige, Compõe, Atua, Pode ser e Trabalha, que ligam, cada um, duas dessas entidades.

As coisas ao qual o MER se referem podem ser pessoas, objetos, conceitos, eventos, etc., existentes ou imaginárias. Elas são classificadas em **entidades**. Alguns autores preferem o termo **tipo de entidade** ao termo entidade. Este texto usa os termos entidade e tipo de entidade indiferentemente, para representar a classe.

Uma entidade representa uma classe de objetos do universo de discurso do modelo . Por exemplo, em uma universidade podemos encontrar um funcionário chamado funcionário José e uma aluna chamada Maria. José é uma instância da entidade funcionário, enquanto Maria é uma instância da entidade aluna. Funcionário e aluno são os tipos de entidade. Cada instância, então, deve poder ser identificada unicamente.

## 14 CAPÍTULO 2. MODELO DE ENTIDADES E RELACIONAMENTOS

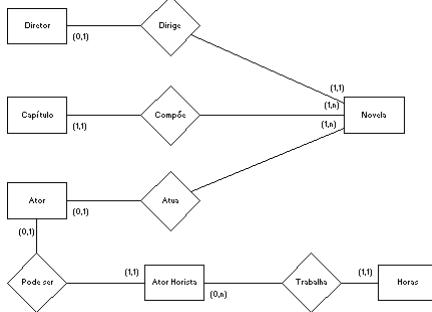


Figura 2.1: Um diagrama de entidades e relacionamentos simples, sem mostrar atributos

Classificar, um tipo de abstração, consiste em resumir uma quantidade de características comuns de objetos que tem identidade própria por meio da criação de uma classe, ou tipo, que os descreva de alguma forma. Assim, é possível saber que todos os funcionários, por serem instâncias de um mesmo tipo, possuem características comuns (como trabalhar na universidade, ter um salário, etc.).

Para representar os objetos específicos, os membros da classe, é adotado o termo **instância**, ou **instância de entidade**. Porém, no discurso normal, a palavra entidade também é muitas vezes usada no lugar de instância.

A priori, só é exigido de uma entidade que cada um dos seus membros possa ser identificado distintamente, isso é, tenha identidade própria. Cada coisa distintamente identificada é uma instância.

Existem várias propostas de representação do MER. Em quase todas elas, no diagrama de entidade e relacionamentos cada tipo de entidade é representado por um retângulo, identificado pelo nome da entidade. Apenas algumas entidades do mundo real (ou imaginário) são de interesse para o sistema. Durante a modelagem conceitual nos preocupamos com as “coisas” que o sistema deve lembrar e colocamos essas “coisas” no modelo de entidade e relacionamentos. Uma entidade deve ser relevante para o objetivo do negócio e necessária para a sua operação.

Cada entidade tem dois tipos de características importantes: seus atributos e seus relacionamentos.

Os atributos são características que toda a instância de um tipo possui, mas que podem variar entre as instâncias. Uma instância do tipo “aluno” tem os atributos “nome” e “ano de matrícula”, por exemplo.

Atributos caracterizam a informação que deve ser guardada sobre uma

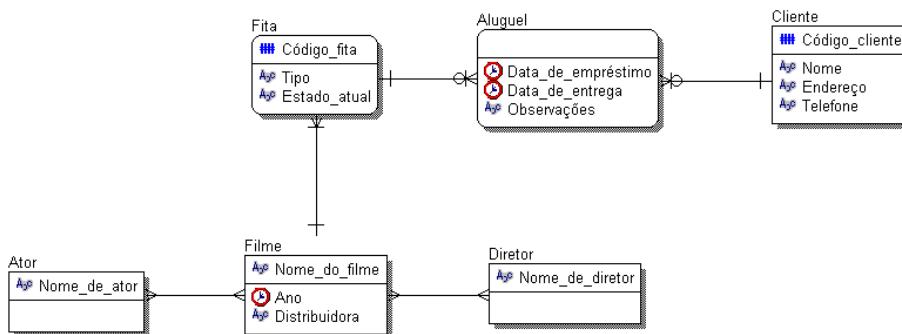


Figura 2.2: Um modelo de entidades e relacionamentos escrito na notação da Engenharia de Informação

entidade. Só devemos colocar como atributos aquelas informações que o sistema precisa lembrar em algum momento. Assim, uma instância de “aluno” não precisa ter o atributo “nome do animal de estimação” em um sistema acadêmico, pois apesar de ser algo importante para o “aluno” propriamente dito, não tem importância alguma para o sistema.

A Figura 2.2 mostra um diagrama de entidades e relacionamentos sobre uma locadora de fitas de vídeo.

Cada característica deve possuir um domínio. O domínio indica o conjunto de valores válidos para a característica. No caso de “nome”, geralmente aceitamos qualquer seqüência de caracteres, enquanto no caso de “altura” podemos aceitar apenas valores reais positivos menores que 2,5.

Atributos eram originalmente descritos por círculos no modelo E-R. As notações mais modernas anotam os atributos dentro dos retângulos da entidade a que pertencem.

Finalmente, como indica o nome do modelo, entidades podem se relacionar entre si. Essa característica é a principal força do modelo de entidades e relacionamentos, pois permite que, de certa forma, “naveguemos” no modelo.

Podemos indicar relacionamentos apenas pelas entidades envolvidas, como “cliente-pedido”, ou usar um termo que descreva o relacionamento “cliente solicita pedido”.

Modelos de Entidades e Relacionamentos para serem completos exigem também um conjunto de restrições. Algumas dessas restrições, como a cardinalidade dos relacionamentos que veremos a seguir, podem ser descritas em algumas (ou todas) notações. Porém, a maioria das descrições é muito

## 16 CAPÍTULO 2. MODELO DE ENTIDADES E RELACIONAMENTOS

complexa para ser descrita em um diagrama. Nesse caso são necessárias anotações ao diagrama descrevendo as descrições. Isso pode ser feito em linguagem natural ou em alguma notação formal específica, dependendo de escolhas da equipe de projeto ou do método utilizado.

DRAFT

# Capítulo 3

## Bancos de Dados Relacionais e SQL

O objetivo deste capítulo é servir de uma pequena introdução ou revisão aos conceitos básicos de bancos de dados relacionais e SQL, que permita o leitor entender o restante do material apresentado no texto. O leitor encontrará cobertura bem mais completa em livros dedicados aos assuntos, como (Elmasri e Navathe, 2016).

Um **Banco de Dados Relacional** é um repositório de dados composto unicamente por tabelas, destinado a registrar as informações que representam o estado de uma sistemas de informação.

Cada tabela é formada por linhas e colunas. As linhas representam fatos de um mesmo tipo sobre o qual se deseja guardar informação, enquanto as colunas representam atributos a serem registrados sobre esses fatos, ou seja, as informações específicas que são guardadas sobre eleElmasri e Navathe (2016). Esses fatos se referem a objetos do mundo real, eventos, momentos, contratos, etc., e são diretamente ligados as instâncias de entidades ou relacionamentos.

Uma célula  $a_{i,j}$  da tabela  $A$  indica o valor do atributo  $j$  para um fato  $i$ . Essa maneira de representar o mundo é conhecida como **Modelo Relacional**.

A Figura 3.1 mostra informalmente uma pequena parte de um banco de dados relacional com três tabelas, criado a partir do modelo ER da 3.2, tratando de avaliações de especialistas sobre o custo de recuperação de terrenos com problemas de contaminação. Cada especialista é descrito por seu nome, em que empresa trabalha e qual sua especialidade. Cada local é descrito por um código, o tipo de terreno e o desenvolvimento da região. Uma terceira

**Especialista**

nome	empresa	especialidade
João	ServAmb	Qualidade de Água
Carlos	EcoConsult	Produtos Químicos
Ana	ServAmb	Vazamentos
...	...	...

**Local**

site_id	terreno	desenvolvimento
L171	Pantanoso	Baixo
M123	Subúrbio	Moderado
...	...	...

**Avaliação**

nome	site_id	custos
João	L171	100K
Paulo	M123	20K
Ana	L171	80K
...	...	...

Figura 3.1: Exemplo de um banco de dados relacional simples

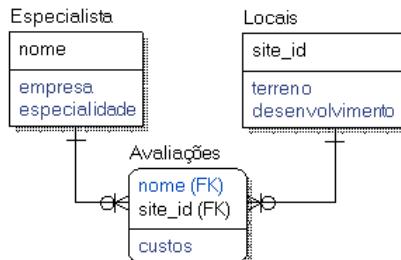


Figura 3.2: Diagrama de entidades e relacionamentos relativo ao banco de dados da Figura 3.1

tabela indica qual o custo de recuperação de um local de acordo com um especialista. Pela forma como foi construída, cada especialista só pode dar uma avaliação por local, porém todas as combinações de local e especialistas são possíveis, porém apenas algumas são verdadeiras e são registradas na tabela.

Essa é uma forma de explicar o que é um Banco de Dados Relacional em linguagem corrente. Os termos tabelas, linhas e colunas são muito fácil de entender e levam ao usuário a pensar em coisas com que tem o hábito de trabalhar.

Bancos de dados relacionais se assemelham de muita formas com o modelo de entidades e relacionamentos, sendo que a transformação de um MER em um MR é razoavelmente direta. Em linhas gerais, relacionamentos  $N \times M$  devem ser eliminados, sendo transformados em novas entidades e relacio-

namentos  $1 \times N$ , e então todas as entidades são convertidas em tabelas, enquanto os relacionamentos são indicados por um campo adicional em uma das tabelas. Heuser (2001) apresenta uma descrição de alta qualidade de como fazer essa transformação.

### 3.1 Definição Formal de um BD Relacional

É interessante também possuir uma noção da definição formal de um banco de dados relacional, como a apresentada por Elmasri e Navathe (2016).

O Modelo Relacional usa o conceito de relação matemática como base de sua construção. Ele foi proposto por Codd (1970).

Um **domínio**  $D$  é um conjunto de valores atômicos. Normalmente um domínio é especificado de acordo com um **tipo de dados**, que define todos os valores possíveis desse domínio.

Exemplos de domínios são: datas, números de telefone, nomes de pessoa, quantidade em dinheiro, etc.

Também é comum que seja especificado um formato para cada domínio. Por exemplo, um formato válido para número de telefone pode ser “+999 999 9999-9999”, indicando três números para o código do país, três para o código de cidade e nove para o número do telefone. Já um domínio relacionado a dinheiro pode ter seu valor máximo limitado e uma indicação de como será representado.

Sendo assim, para definir um domínio são necessários:

- um nome,
- um tipo de dados,
- um formato e
- outras informações adicionais que auxiliem a interpretação do domínio

A definição do domínio é arbitrária, mas segue a lógica de um conceito desejado em um contexto específico. A Tabela 3.1 apresenta a definição de dois domínios diferentes para o mesmo conceito “sexo”, para serem usados em contextos diferentes.

No primeiro caso, o sexo é definido de acordo com o padrão ISO/IEC 5218 (ISO/IEC, 2004). No segundo, segundo as necessidades de um consultório médico.

Tabela 3.1: Dois tipos de domínio para o mesmo conceito

	ISO/IEC 5218	Médico
Nome	Sexo	Sexo
Tipo de Dados	$\{0, 1, 2, 9\}$	F ou M
Formato	o número em Unicode	a letra em Unicode
Significado	0 - não informado 1 - masculino 2 - feminino 9 - não se aplica	F - Feminino M - masculino

Um **esquema de relação** é denotado por

$$R(A_1, A_2, \dots, A_n) \quad (3.1)$$

e contém um nome de relação  $R$  e uma lista finita de **atributos**  $A_1, A_2, \dots, A_n$  onde cada atributo  $A_i$  é um papel que algum domínio assume na relação  $R$ .

$D$  é chamado o domínio de  $A_i$ ,  $D = \text{dom}(A_i)$ . O **grau da relação** é o número de atributos  $n$ . O esquema da relação descreve uma **relação** chamada  $R$ .

Os esquemas de relação da Figura 3.1 são:

- Especialista(nome, empresa, especialidade)
- Local(site\_id, terreno, desenvolvimento)
- Avaliação(nome, site\_id, custos).

Uma **relação** do esquema de relação  $R(A_1, A_2, \dots, A_n)$ , denotada por  $r(R)$  é um conjunto de  $n$ -tuplas  $r = \{t_1, t_2, \dots, t_k\}$ , onde cada tupla é uma lista ordenada de valores  $t = < v_1, v_2, \dots, v_n >$  onde cada  $v_j$ ,  $1 \leq j \leq n \Rightarrow v_j \in \text{dom}(A_j) \vee v_j = \text{nulo}$ .

Ou seja, na definição formal, um esquema de relação explica como é organizada a relação, que é a tabela, já preenchida com os dados, da definição informal.

Para um mesmo esquema de relação podem existir várias, possivelmente infinitas, relações possíveis. Em um certo instante do tempo, porém, provavelmente apenas uma relação representa corretamente a informação necessária para um sistema de informação.

Como uma relação não é ordenada, então as tuplas de uma relação não são ordenadas.

É importante notar que dentro do Modelo Relacional só existem tabelas. Assim, para consultar a base de dados são feitas operações com tabelas que geram outras tabelas, com o resultado desejado da consulta imaginada. Dentro da teoria foram desenvolvidos tanto uma álgebra quanto um cálculo relacional, que definem operações que sempre geram tabelas como resultado Elmasri e Navathe (2016). A linguagem SQL é uma tentativa de implementar a álgebra relacional por meio de uma linguagem declarativa.

## 3.2 Sistemas Gerenciadores de Bancos de Dados

Bancos de dados relacionais se tornaram ubíquos em sistemas computacionais porque criam uma camada de abstração entre vários sistemas e um conjunto comum de dados para a organização.

Os dados são armazenados, gerenciados e acessados por meio de **Sistemas Gerenciadores de Banco de Dados, SGDB**, que permitem o controle e acesso aos dados por meio de uma linguagem padronizada conhecida como SQL.

Assim, o que a grande maioria das aplicações faz é garantir a visualização adequada e a lógica correta de negócio e consultar e guardar os dados em um SGDB Relacional, por meio de comandos nessa linguagem.

Existem muitos SGDB no mercado, boa parte deles seguindo o modelo relacional. Entre os mais conhecidos e usados estão: MySQL, PostgreSQL, Microsoft SQL Server, e Oracle Database. Os dois primeiros são open-source e possuem licenças para uso gratuito. O Microsoft SQL Server e o Oracle são proprietários, mas tem uma licenças gratuitas para desenvolvimento e uma versão Express para pequenas aplicações, também grátis<sup>1</sup>.

## 3.3 SQL

**SQL**<sup>2</sup>, sigla que significa **Structured Query Language**, é uma linguagem de consulta e manipulação de banco de dados onde o programador explicita o que ele deseja e não como ele deseja que a operação seja feita. Dessa forma,

<sup>1</sup>Informações válidas em novembro de 2019

<sup>2</sup>Deve ser pronunciada S-Q-L, letra a letra, em português ou como a palavra *sequel* em inglês.

SQL é uma linguagem declarativa e padronizada, apesar de cada vendedor possuir algumas extensões.

A linguagem pode ser dividida em Várias partes. A maioria dos autores registra duas partes principais: **Data Manipulation Language ( DML)**, **Data Definition Language(DDL)**.

Outros autores ou manuais de sistemas específicos apresentam ainda variações com os seguintes subconjuntos: **Data Query Language (DQL)**, **Data Transaction Language (DTL)** e **Data Control Language(DCL)**. Este texto usa a divisão mais detalhada para melhor explicar o espectro total de SQL.

A seguir será feita uma breve introdução a algumas instruções da linguagem SQL. Para maiores informações devem ser procurados livros textos, como (Elmasri e Navathe, 2016) e sites e manuais das ferramentas específicas. Nesta introdução é usada a linguagem do sistema MySQL(Oracle Corporation, 2019b), devido a seu largo uso na criação de web sites e na academia.

### 3.3.1 Data Definition Language

A função da **Data Definition Language** é permitir definir e alterar a estrutura do banco de dados, isto é, criar, alterar e modificar tabelas e outros objetos do banco de dados. Deste modo, a DDL trata dos esquemas de relação, ou seja, das estruturas das tabelas.

Esse subconjunto de SQL é formado pelos comandos CREATE, ALTER, DROP, RENAME, TRUNCATE.

Neste texto serão tratados as instruções CREATE, DROP que criam e apagam tabela e outros elementos de um banco de dados.

#### A instrução CREATE TABLE

A instrução **CREATE TABLE** é usada para criar uma tabela. Sua sintaxe permite definir o nome da tabela, seus campos, os tipos de dados dos campos, chaves e índices.

O formato mais simples da instrução é o seguinte:

```
CREATE TABLE nome_da_tabela (
    nome_da_coluna_1 tipo_de_dados PRIMARY KEY,
    nome_da_coluna_2 tipo_de_dados,
```

```

    nome_da_coluna_3 tipo_de_dados,
    ...
);

```

Para criar as tabelas da Figura 3.1 seriam necessários os comandos:

```

CREATE TABLE Especialista
(
    nome          CHAR(255) PRIMARY KEY,
    empresa       CHAR(255),
    especialidade CHAR(255)
);

CREATE TABLE Locais
(
    site_id      VARCHAR(20) PRIMARY KEY,
    terreno       VARCHAR(20),
    desenvolvimento VARCHAR(20)
);

CREATE TABLE Avaliacoes
(
    nome          CHAR(255) PRIMARY KEY,
    site_id       VARCHAR(20) PRIMARY KEY,
    custos        DECIMAL(19,4),
);

```

A instrução CREATE TABLE fornece variações poderosas que normalmente podem ser encontradas no manual de cada ferramenta. Além disso, para otimizar a base seria necessário criar restrições adicionais e índices. Sua sintaxe descrita no manual do MySQL(Oracle Corporation, 2019b) contém 147 linhas e ainda se refere a outras listagens.

Outras instruções do tipo CREATE existem em SQL, mas não serão tratadas neste texto.

### A instrução DROP TABLE

A instrução **DROP TABLE** simplesmente apaga uma tabela do sistema, tanto os dados como sua estrutura. Sua forma mais comum é: “DROP

TABLE NOME\_DA\_TABELA”. Um exemplo que destruiria a tabela “Local” é:

```
| DROP TABLE Local;
```

### 3.3.2 Data Query Language

A **Data Query Language** tem como finalidade fornecer funções de consulta a base de dados.

Esse subconjunto de SQL é formado pelos comandos SHOW, SELECT, que é o comando de consulta as informações no SGDB.

O comando SELECT é um dos mais poderosos de toda a linguagem. Sua forma mais simples retorna uma tabela inteira. Por exemplo, todos as linhas da tabela “Local” seriam retornadas com a instrução:

```
| SELECT * FROM Local;
```

Se apenas alguns campos forem necessários, então a consulta poderia ser:

```
| SELECT site_id, terreno FROM Local;
```

Porém, o maior poder da instrução vem de algumas capacidades adicionais. A primeira é a seleção de linha por meio de valores, como em:

```
| SELECT site_id, terreno FROM Local WHERE desenvolvimento = "baixo";
```

Mais ainda, é possível listar novas tabelas a partir de outras, realizando operações conhecidas como **junção**, como em:

```
| SELECT Local.site_id, terreno FROM Local , Avaliacao WHERE desenvolvimento =
```

Para deixar clara a complexidade da instrução SELECT, o texto a seguir apresenta a sintaxe do mesmo em MySQL(Oracle Corporation, 2019b).

```
| SELECT
  [ALL | DISTINCT | DISTINCTROW ]
  [HIGH_PRIORITY]
```

```

[STRAIGHT_JOIN]
[SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
[SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
select_expr [, select_expr ...]
[FROM table_references
    [PARTITION partition_list]
[WHERE where_condition]
[GROUP BY {col_name | expr | position}, ... [WITH ROLLUP]]
[HAVING where_condition]
[WINDOW window_name AS (window_spec)
    [, window_name AS (window_spec)] ...]
[ORDER BY {col_name | expr | position}
    [ASC | DESC], ... [WITH ROLLUP]]
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
[INTO OUTFILE 'file_name'
    [CHARACTER SET charset_name]
    export_options
    | INTO DUMPFILE 'file_name'
    | INTO var_name [, var_name]]
[FOR {UPDATE | SHARE} [OF tbl_name [, tbl_name] ...] [NOWAIT | SKIP LOCKED]
    | LOCK IN SHARE MODE]]

```

Essa sintaxe esconde muita complexidade, principalmente nos termos `table_references` e `where_condition`.

O exemplo a seguir mostra uma sentença SELECT contendo outra cláusula SELECT dentro dela, que responde com o nome de todos os especialistas que não fizeram nenhuma avaliação de mais de R\$ 100.000,00 reais.

```

SELECT nome
    FROM Especialista
 WHERE nore NOT IN (SELECT nome
                    FROM Avaliacoes
                   WHERE custos > 100000);

```

### 3.3.3 Data Manipulation Language

Esse subconjunto de SQL é formado pelos comandos INSERT, UPDATE, DELETE, MERGE, CALL, EXPLAIN PLAN e LOCK TABLE.

Neste texto serão tratados os comandos INSERT, UPDATE e DELETE são responsáveis por colocar, alterar e apagar informações das tabelas de um SGDB.

## DELETE

A instrução delete é a mais simples de definir. Para isso, basta normalmente usar algo como:

```
| DELETE FROM Local WHERE terreno = "Arenoso";
```

Sua sintaxe completa no MySQL é(Oracle Corporation, 2019b):

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM tbl_name [[AS] tbl_alias]
| [PARTITION (partition_name [, partition_name] ...)]
| [WHERE where_condition]
| [ORDER BY ...]
| [LIMIT row_count]
```

## INSERT

O objetivo dessa instrução é inserir dados em uma tabela. Sua forma mais usada é simplesmente inserir uma linha em uma tabela listando os valores de todos os seus campos na ordem correta(Oracle Corporation, 2019b), como em:

```
| INSERT INTO Local VALUES("s123-a", "pantanoso", "alto");
```

Sua sintaxe, em MySQL(Oracle Corporation, 2019b), permite variações:

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
| [INTO] tbl_name
| [PARTITION (partition_name [, partition_name] ...)]
| [(col_name [, col_name] ...)]
| {VALUES | VALUE} (value_list) [, (value_list)] ...
| [ON DUPLICATE KEY UPDATE assignment_list]

INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
```

```
[INTO] tbl_name
[PARTITION (partition_name [, partition_name] ...)]
SET assignment_list
[ON DUPLICATE KEY UPDATE assignment_list]

INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]
[INTO] tbl_name
[PARTITION (partition_name [, partition_name] ...)]
[(col_name [, col_name] ...)]
SELECT ...
[ON DUPLICATE KEY UPDATE assignment_list]

value:
{expr | DEFAULT}

value_list:
value [, value] ...

assignment:
col_name = value

assignment_list:
assignment [, assignment] ...
```

## UPDATE

A instrução update permite alterar o valor de células específicas de acordo com uma condição. Por exemplo para alterar o valor do custo de descontaminação do site L171 em mais 10%, o seguinte comando seria válido.

```
| UPDATE Avaliacoes SET custos=custos*1.1 WHERE site_id="L171";
```

A sintaxe completa da instrução UPDATE para MySQL(Oracle Corporation, 2019b) é:

```
UPDATE [LOW_PRIORITY] [IGNORE] table_reference
SET assignment_list
[WHERE where_condition]
[ORDER BY ...]
[LIMIT row_count]
```

```

value:
  {expr | DEFAULT}

assignment:
  col_name = value

assignment_list:
  assignment [, assignment]

UPDATE [LOW_PRIORITY] [IGNORE] table_references
  SET assignment_list
  [WHERE where_condition]

```

### 3.3.4 Data Transaction Language

Esse subconjunto de SQL é formado pelos comandos BEGIN TRANSACTION, COMMIT E ROLLBACK. Esses comandos permitem que o início, o fim e ainda abortar uma transação composta de várias operações no banco de dados.

### 3.3.5 Data Control Language

Esse subconjunto de SQL é formado pelos comandos GRANT, DENY e REVOKE. Esses comandos controlam os privilégios de acesso às tabelas do SGDB e não são tratados neste texto.

## 3.4 O que mais

Existe muito mais a falar sobre bancos de dados relacionais e SQL. Milhares de textos já foram escritos sobre o assunto. Este capítulo tem como finalidade apenas ser uma visão rápida, em especial para o leitor que já viu o tema e pode ter se esquecido de algumas coisas.

Havendo mais interesse, os livros *Fundamentals of Database Systems* de Elmasri e Navathe (2016), *An introduction to database systems* de Date (2004) e *Joe Celko's SQL for smarties: advanced SQL programming* Celko (2005) são boas referências. Uma boa introdução a SQL pode ser encontrada

também no site W3School (<https://www.w3schools.com/sql>). Há muito mais material de boa qualidade disponível em livros, artigos, tutoriais e sites.

DRAFT

DRAFT

# Capítulo 4

## OLAP

**OLAP** é a sigla para **Online Analytical Processing**. O termo é usado para diferenciar o tipo de sistema, ou de operações, que se usa quando o objetivo é analisar os dados de uma organização, normalmente para a tomada de decisão, e o uso normal dos dados dos sistemas que suportam o funcionamento da organização, conhecidos como transacionais ou **Online Transaction Processing, OLTP**, ou ainda **processamento de transações em tempo real**.

Assim, sistemas OLTP e sistemas OLAP, dentro de uma organização, são normalmente usados em momentos diferentes, por usuários diferentes e com objetivos diferentes. Além disso, guardam tipos de dados diferentes.

Sistemas OLAP são tipicamente construídos sobre data warehouses, mas não necessariamente. Além disso, eles admitem uma série de operações, conhecidas como **operações OLAP**, que permitem navegar e analisar os dados, normalmente por meio de um programa de computador com interface de usuário bastante dinâmica.

As principais operações OLAP são o **slice and dice** e o **drill down and roll up**. Além disso são também conhecidas as operações **pivot** e **drill across**.

Para ilustrar algumas operações, neste capítulo serão usados como dados exemplos uma planilha com os micro-dados do ENEM 1998<sup>1</sup>(INEP, 2016) dentro do software Microsoft Excel™(Microsoft, 2019).

---

<sup>1</sup>Esta planilha tem poucos candidatos e poucas colunas

	A	B	C	D	E	F	G	H	I	J	K
1	NU_INSCRICAO	NU_ANO	NU_IDADE	TP_SEXO	CO_MUNICIPIO_RESIDE	NO_MUNICIPIO_RESI	CO_UF_RESIDEN	SG_UF_I	TP_PRES	CO_PRC	VL_PE
2	1	1998	19	F	2408003	MOSSORO	24	RN	1B	43.3	
3	2	1998	17	F	2401453	BARAUNA	24	RN	1B	23.3	
4	3	1998	18	M	2408003	MOSSORO	24	RN	1Z	46.7	
5	4	1998	18	M	2408003	MOSSORO	24	RN	1G		3
6	5	1998	21	M	2413102	SENADOR REOLIO DE SO	24	RN	1A	43.3	
7	6	1998	17	F	2408102	NATAL	24	RN	1G	56.7	
8	7	1998	18	M	2408102	NATAL	24	RN	1A	36.7	
9	8	1998	17	M	2408102	NATAL	24	RN	0		
10	9	1998	30	M	2408102	NATAL	24	RN	0		
11	10	1998	19	M	2408102	NATAL	24	RN	0		
12	11	1998	18	M	2412104	SAO JOAO DO SABU	24	RN	1B	4	
13	12	1998	18	F	2408102	NATAL	24	RN	0		
14	13	1998	25	F	2408102	NATAL	24	RN	0		
15	14	1998	19	F	2408102	NATAL	24	RN	1B	3	
16	15	1998	38	F	2408102	NATAL	24	RN	0		
17	16	1998	21	M	2408102	NATAL	24	RN	0		
18	17	1998	21	M	2408102	NATAL	24	RN	1G	46.7	
19	18	1998	23	F	2402006	CAICO	24	RN	1G	3	
20	19	1998	18	F	2408102	NATAL	24	RN	1G	23.3	
21	20	1998	17	F	2408102	NATAL	24	RN	1B	66.7	
22	21	1998	22	F	2408102	NATAL	24	RN	0		
23	22	1998	17	M	2408102	NATAL	24	RN	1Z	6	
24	23	1998	23	F	2402006	CAICO	24	RN	1A	2	
25	24	1998	20	M	2408102	NATAL	24	RN	0		
26	25	1998	17	F	2408102	NATAL	24	RN	0		
27	26	1998	18	F	2408102	NATAL	24	RN	1Z	4	
28	27	1998	18	F	2408102	NATAL	24	RN	1Z	33.3	
29	28	1998	17	F	2408102	NATAL	24	RN	1B	36.7	
30	29	1998	22	F	2408102	NATAL	24	RN	1A	36.7	
31	30	1998	19	F	2403251	PARANAMirim	24	RN	0		
32	31	1998	19	F	2408003	MOSSORO	24	RN	0		
33	32	1998	18	F	2408003	MOSSORO	24	RN	1G	3	
34	33	1998	31	F	2408003	MOSSORO	24	RN	1Z	23.3	
35	34	1998	21	M	2408003	MOSSORO	24	RN	0		2
36	35	1998	21	F	2408003	MOSSORO	24	RN	0		
37	36	1998	21	F	2408003	MOSSORO	24	RN	1G	26.7	
38	37	1998	42	F	2408003	MOSSORO	24	RN	0		

Figura 4.1: Pequena fotografia de uma planilha com os micro-dados do ENEM 1998. Fonte (INEP, 2016).

## 4.1 O Cubo de Dados

Um metáfora usada para entender melhor as aplicações OLAP é a do Cubo de Dados. Usa-se um cubo porque é uma imagem 3D que pode ser entendida quando desenhada, e pode ser usada por algumas aplicações.

A ideia do cubo é estender para uma dimensão a mais o conceito de uma matriz de dados, para permitir visualizar melhor as operações OLAP.

Um cubo de dado mostra um dado de acordo com algumas dimensões. Por exemplo o cubo de dados OLAP da Figura 4.2 mostra a visão de dados sobre as vendas de uma rede de lojas de calçado. Em cada célula desse cubo

está a informação desejada.

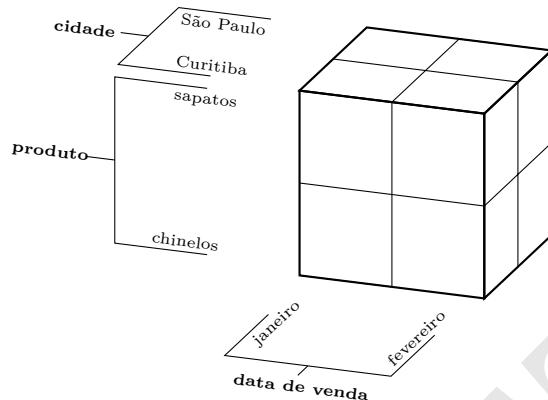


Figura 4.2: Cubo OLAP simplificado para rede de lojas de calçado.

Uma representação simples do cubo de dados OLAP é a de uma tabela com várias páginas. Assim, as linhas da tabela indicam a primeira dimensão, as colunas a segunda, e a terceira dimensão é indicada pela página.

## 4.2 OLAP com Pivot Table no Excel<sup>TM</sup>

As planilhas eletrônicas normalmente possuem uma funcionalidade que permite fazer facilmente uma análise de dados OLAP. Essa funcionalidade de análise é conhecida como **pivot table**.

Por exemplo, a Figura 4.1 mostra uma imagem com um pedaço de uma planilha Excel<sup>TM</sup> contendo uma única tabela com os micro-dados do ENEM de 1998(INEP, 2016).

A partir dessa tabela foi criada outra aba, contendo uma *pivot table*, que foi configurada para contar as presenças. Para isso foi usado o comando Insert → Table → Pivot Table, o que faz aparecer o diálogo da Figura 4.3. Normalmente basta apertar Ok.

A *pivot table* é controlada em duas áreas. Em uma, Pivot Table Field, detalhada na Figura 4.4 , é possível selecionar e configurar que campos serão mostrados de que forma, inclusive controlando fórmulas de cálculo.

Na outra, que parece uma planilha comum, Figura 4.5 , acontecem visualizações dinâmicas do que é selecionado e ainda podem ser editados alguns nomes e usados filtros.

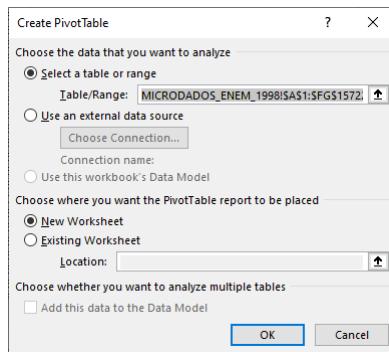


Figura 4.3: Diálogo do Pivot Table do Microsoft Excel<sup>TM</sup>365.

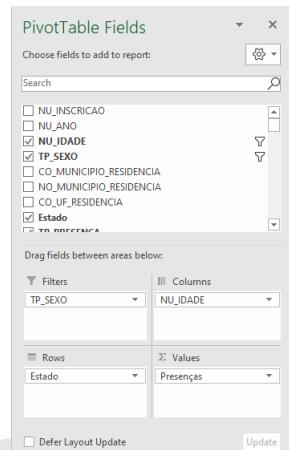


Figura 4.4: Nesse quadro são controladas que colunas aparecem em que posição na pivot table

Analizando as Figuras 4.4 e 4.5 é possível ver que foram selecionadas algumas colunas para aparecer na planilha. É possível definir três dimensões para o Cubo, *Filters*, que indica uma espécie de página, *Columns*, que indica as colunas da tabela e *Rows*, que indicam as linhas da tabela. Mais tarde no capítulo serão usadas hierarquias dentro das dimensões.

No quadro *Filters* foi escolhido o campo “TP\_SEXO”, que indica o sexo do candidato e só tem duas opções “F” ou “M”. Para as linhas foi escolhido o campo “SG\_UF\_RESIDENCIA”, que foi renomeado “para Estado”. Para coluna foi escolhido o campo “NU\_IDADE”, que indica a idade do candidato. E para valor foi escolhido o campo “TP\_PRESENÇA”, também renomeado para “Presenças”.

Clicando em “Presenças” é possível escolher como vai ser feito o cálculo

The screenshot shows a Microsoft Excel spreadsheet with a pivot table. The pivot table has 'Estado' as the column label and 'Presenças' as the row label. The data includes columns for '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', and '20'. The data rows represent different states: AC, AL, AM, AP, BA, CE, DF, ES, GO, MA, MG, MS, MT, PA, PB, PE, PI, PR, RJ, RN, RO, and RR. The values in the cells represent the count of presences, with '1' indicating presence and '0' indicating absence.

	A	B	C	D	E	F	G	H	I	J	K	L
1	TP_SEXO	F	Y									
2												
3	Presenças	Idade	Y									
4	Estado		10	11	12	13	14	15	16	17	18	19
5	AC		1						1	38	28	7
6	AL								5	48	22	5
7	AM			1	5	31	82	60	12	11	1	1
8	AP						1	6	12	4		
9	BA						7	34	42	7		
10	CE						10	197	131	24	16	
11	DF						1	3	47	45	8	
12	ES						6	61	665	558	208	70
13	GO						12	146	74	20	16	
14	MA						1	19	17	5		
15	MG		0				0	19	1704	2500	1219	73
16	MS						4	6	348	151	55	21
17	MT						1	23	286	173	107	51
18	PA						1	1	9	63	29	13
19	PB						1	8	83	41	12	
20	PE		1	0	1	1	1	65	788	781	484	37
21	PI						1	1	3	13	11	2
22	PR		2	6	1	3	1	32	5451	5291	3222	193
23	RJ		2	1	0	1	33	336	3069	3494	2136	122
24	RN						0	1	22	272	388	248
25	RO							2	21	27	8	
26	RR						1	2	16	24	16	1

Figura 4.5: Na planilha, a pivot table vai se alterando dinamicamente para atender a configuração dada no quadro de controle.

desse campo e foi escolhido a forma *Sum*, já que a presença é marcado com o número 1 e a ausência com o 0. Isso é mostrado na Figura 4.6.

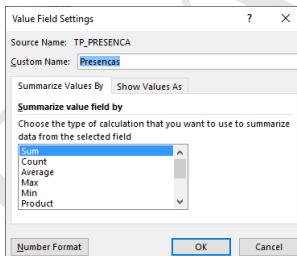


Figura 4.6: Escolhendo “Sum” como função de agregação de dados para a presença dos candidatos.

### 4.3 Slice and Dice

A primeira operação OLAP, **slice and dice** é bem fácil de entender e é bastante conhecida em outras áreas pelo nome de **filtro**.

Nessa operação simplesmente o usuário escolhe, dentro de uma visão que mostra muitos dados, uma coleção menor de dados para ver. O termo **slice** indica que ele fez essa escolha em apenas uma dimensão da consulta, cortando o cubo OLAP em uma fatia, e o termo **dice** indica que fez em várias, cortando essa fatia mais vezes e obtendo um cubo menor.

Usando o cubo da Figura 4.2, uma operação de *slice* poderia requisitar apenas os dados de São Paulo, resultando na Figura 4.7.

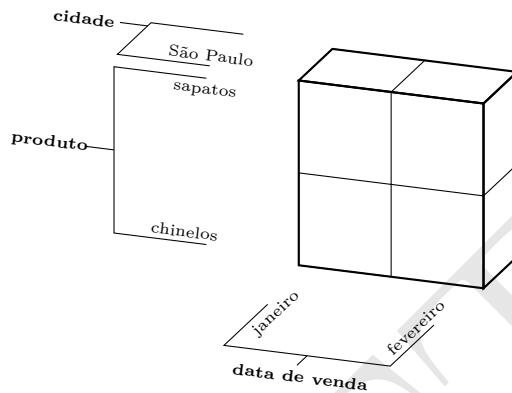


Figura 4.7: Efeito do uso de um filtro selecionando apenas os dados da cidade de São Paulo, realizando um *slice*.

E usando mais uma vez esse cubo, é possível selecionar apenas as vendas de janeiro, o que acaba resultando em um *dice*, como na Figura 4.8.

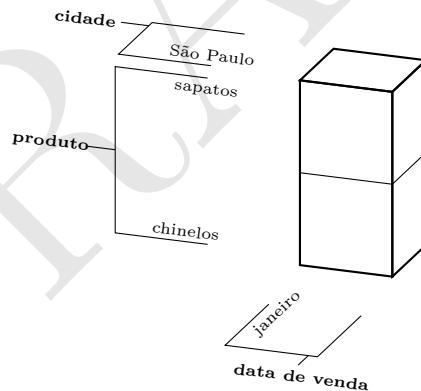


Figura 4.8: Aplicando mais de um filtro, obtemos um *dice*.

É possível observar que, em relação ao cubo original, a Figura 4.7 faz um corte, portanto é um *slice*, ou seja, uma fatia do cubo. Já na figura 4.8 mais um corte é aplicado e ficamos com um pedaço menor que uma fatia, um *dice*.

Por exemplo, a Tabela 4.1 apresenta os dados de todos os estados do Brasil, tornando difícil a comparação dos dados dos estados do Sul do Brasil, RS, PR e SC, entre eles. Uma operação de *slice* então corta essa fatia, por meio de filtros, e mostra os dados desejados, como na Tabela 4.2.

Tabela 4.1: Presenças por moradores dos estados no ENEM de 1998, Fonte: (INEP, 2016).

Estado	Presenças
AC	368
AL	152
AM	355
AP	46
BA	170
CE	622
DF	199
ES	2650
GO	443
MA	80
MG	14679
MS	1070
MT	1192
PA	248
PB	269
PE	5665
PI	68
PR	47650
RJ	22296
RN	2584
RO	99
RR	546
RS	800
SC	1001
SE	289
SP	7433
TO	44
(blank)	4557
Total	115575

Tabela 4.2: Presenças por moradores dos estados no ENEM de 1998, Fonte: (INEP, 2016).

Estado	Presenças
PR	47650
RS	800
SC	1001
Total	115575

### 4.3.1 Filtros (*Slice and Dice*) no Pivot Table do Excel™

Para usar os filtros do pivot table do Excel™ é preciso apertar nas pequenas caixas com setas, ou filtros, que são vistas ao lado das palavras “Estado”, “Idade” e “TP\_SEXO” na 4.5. No caso do aparecimento de um filtro, significa que há um filtro ativo. Nessa imagem já está sendo feito um filtro pelo sexo e um pela idade, logo caracterizando uma operação de *dice*.

Alterando os filtros, é possível obter outros resultados. Por exemplo, na Figura 4.9, são eliminadas as idades menores que 17 anos, resultando na tabela da Figura 4.10.

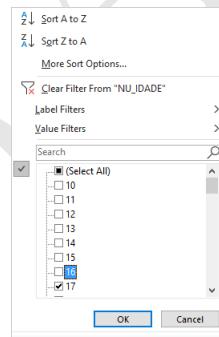


Figura 4.9: Usando o filtro para cortar algumas idades

	A	B	C	D	E	F	G	H	I	J	
1	TP_SEXO	F									
2											
3	Presenças	Idade	.	Y							
4	Estado		17	18	19	20	21	22	23	24	25
5	AC		38	28	7	9	7	2	1	1	1
6	AL		48	22	5	3	0	1	0	0	
7	AM		82	60	12	18	7	4	6	3	2
8	AP		6	12	4	1	4				
9	BA		34	42	7	9	3	1			
10	CE		197	131	24	10	2	2	1	3	1
11	DF		47	45	8	4	2	1	1	0	1
12	ES		655	558	208	70	40	14	12	5	5
13	GO		146	74	20	10	6	5	5	2	
14	MA		19	17	5	4	2	1		1	
15	MG		1704	2500	1219	730	456	302	208	124	136
16	MS		348	151	55	28	12	8	4	4	3
17	MT		286	173	107	52	34	17	9	12	3
18	PA		63	29	13	10	5	2	1		
19	PB		83	41	12	5	2	1			
20	PE		788	781	484	372	250	192	142	115	70
21	PI		13	11	2	2	1	2	1	2	
22	PR		5451	5291	3222	1930	1173	778	553	450	333
23	RJ		3069	3494	2136	1227	657	443	280	199	136
24	RN		272	388	248	177	143	80	64	52	28
25	RO		21	27	8	2	1		2		
26	RR		16	24	16	13	10	8	6	4	4

Figura 4.10: Tabela de presenças no ENEM 1998 por estado para maiores de 17 anos. Fonte (INEP, 2016)

## 4.4 Drill-down e Roll-up

Quando estamos consultando uma base de dados é comum que uma pergunta possa ser feita em vários níveis de uma hierarquia que cobre o mesmo conceito.

Por exemplo, todas as perguntas a seguir querem obter informações sobre o valor total vendido, que é a informação importante, em um período de tempo, que é o conceito:

- Qual o valor total vendido por dia?
- Qual o valor total vendido por semana?
- Qual o valor total vendido por mês?
- Qual o valor total vendido por trimestre?
- Qual o valor total vendido por ano?

O mesmo pode ser visto em uma consulta sobre a quantidade de litros de combustível vendida por localidade:

- Qual a quantidade de litros de combustível vendida por posto de gasolina?
- Qual a quantidade de litros de combustível vendida por bairro?
- Qual a quantidade de litros de combustível vendida por cidade?
- Qual a quantidade de litros de combustível vendida por estado?
- Qual a quantidade de litros de combustível vendida por país?

Hierarquias como [dia - semana - mês - trimestre - ano] e [posto - bairro -

cidade - estado - país], são muito comuns. Essas duas, a de intervalo de tempo e a de localidade ou endereço, inclusive, são as mais comuns e aparecem em quase todos data warehouses. Outras hierarquias comuns são [produto - subcategoria - categoria], ou [setor - departamento - diretoria].

Essas hierarquias são muitas vezes desenhadas, como na Figura 4.11. Em especial, o desenho mostra que mês e semana devem ser vistos em paralelo, isto é, não é possível navegar de semana para mês nesse modelo.



Figura 4.11: Hierarquias de conceitos. No caso, na hierarquia de períodos de tempo, temos dois caminhos paralelos.

Muitas vezes quando fazemos uma pergunta como as listadas acima, como “Qual o valor total vendido por trimestre?” ficamos insatisfeitos com o resultado e queremos mais detalhe. Por exemplo, queremos saber o valor total vendido por mês, em um determinado trimestre. Essa operação, onde pedimos mais detalhes sobre uma dimensão da pergunta, é conhecida como **drill down**, pois é como se estivéssemos “escavando mais profundamente” para entender o que está acontecendo.

Normalmente o *drill down* é feito em uma interface gráfica, onde o usuário clica em um dado para ver o detalhe. Por exemplo, clicando em um dado sobre um mês ele poderia ver os dados sobre cada dia do mês.

A ideia básica é demonstrada na Figura 4.12, escolhendo uma célula do cubo de OLAP e fazendo o *drill down*, chegamos a um cubo mais detalhado.

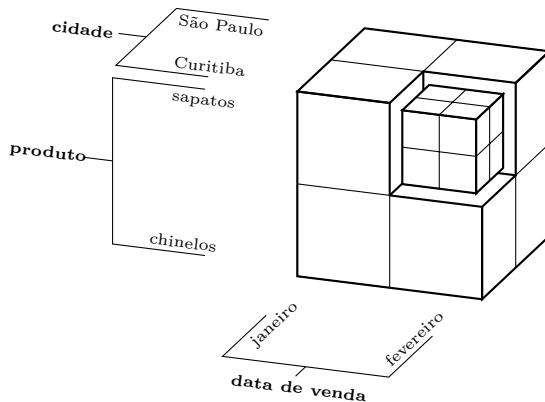


Figura 4.12: Fazendo um *drill down* se obtém um novo cubo mais detalhado.

Não é necessário todo um cubo para entender um *drill down*, basta olhar para uma tabela com dados agregados e procurar abrir esses dados de alguma forma prevista na interface do software.

A Tabela 4.1, por exemplo, mostra o número de presenças no ENEM de 1998, por estado de residência do candidato. Supondo que um usuário, vendo esse dado, queira saber de onde vieram os candidatos do Acre; em um sistema que suportasse o *drill down*, ele poderia clicar no Acre e receber uma visão da Tabela 4.3.

Tabela 4.3: Presenças por moradores das cidades do Acre no ENEM de 1998. Fonte: (INEP, 2016)

Estado	Presenças
ACRELANDIA	2
MANCIO LIMA	1
RIO BRANCO	365
Total AC	368

A operação inversa, que nos permite ter uma visão mais ampla, foi chamada inicialmente de **roll up**, e com o tempo ganhou também o nome de **drill up**<sup>2</sup>.

Na operação inversa é importante notar que a visão passa a um nível superior da hierarquia. Então, pedindo para fazer um roll-up quando vê a tabela de cidade do Acre (4.3, o usuário passaria a ver a tabela que inclui todos os estados, e o Acre como um deles (4.1).

<sup>2</sup>Que, convenhamos, não faz muito sentido

#### 4.4.1 Drill down e Roll Up

Nas linhas foi definida uma hierarquia [município - estado], como mostrado na Figura 4.13. A imagem ainda mostra, à direita, o painel de configuração da pivot table.

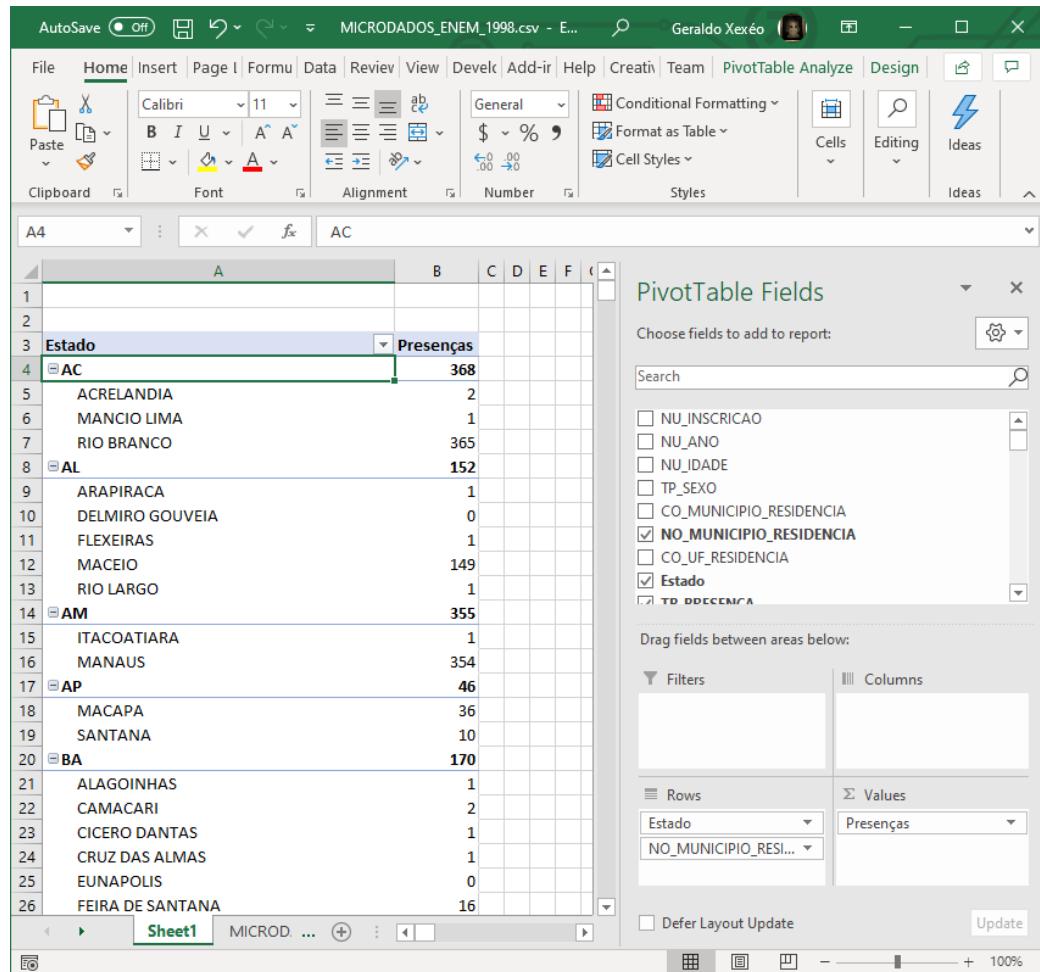


Figura 4.13: Pequena fotografia de uma planilha com uma pivot table verificando as presenças por cidade e estado do ENEM 1998. Fonte (INEP, 2016).

Clicando nos estados é possível abrir ou fechar o detalhe dos municípios, o equivalente à função de *drill down*. A Figura 4.14 mostra uma imagem da mesma pivot table com alguns estados fechados, ou seja, tendo sofrido um *roll up*.

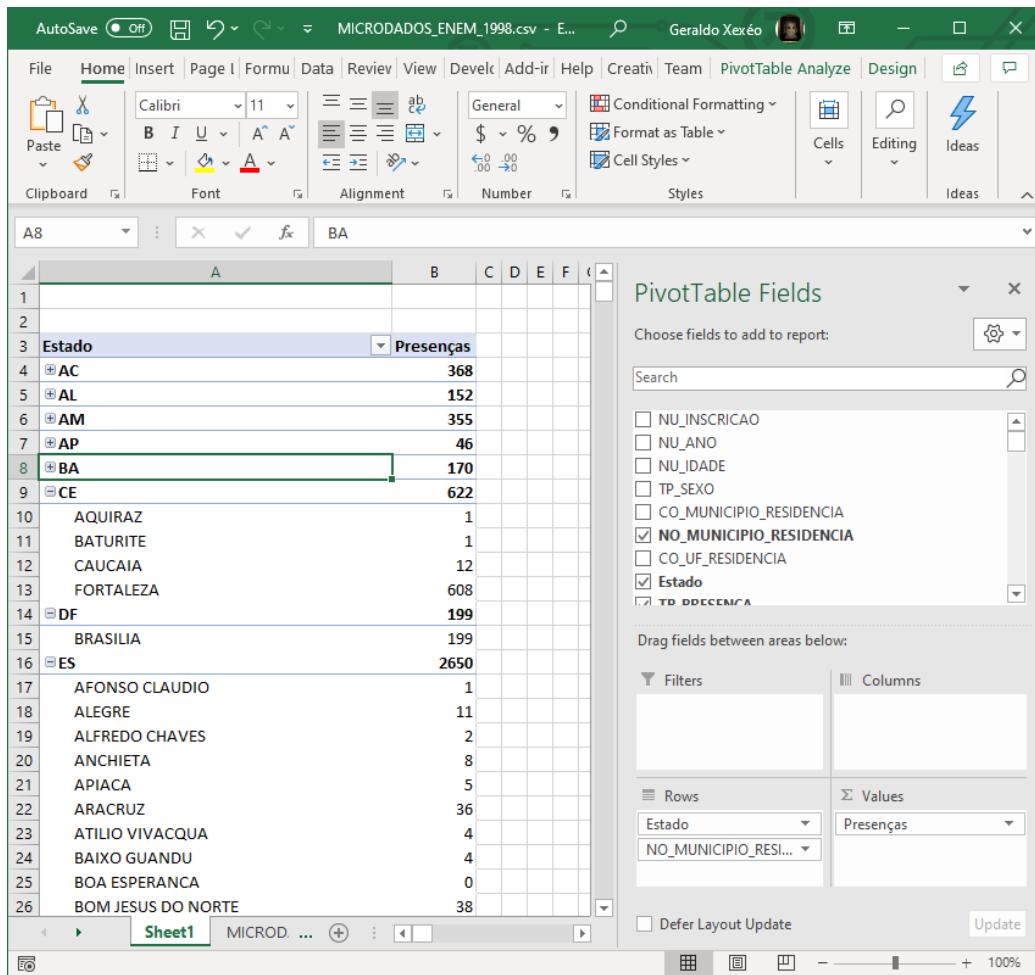


Figura 4.14: Pivot table com algumas informações suprimidas por meio de um *roll up* nos estados de Acre, Alagoas, Amazonas, Amapá e Bahia. Fonte (INEP, 2016).

## 4.5 Pivot

A operação **pivot** é basicamente uma mudança de visualização, que reorienta o cubo. Seu efeito típico é trocar linhas por colunas, ou páginas por linhas.

Uma operação de *pivot* na Figura 4.2 poderia resultar na 4.15.

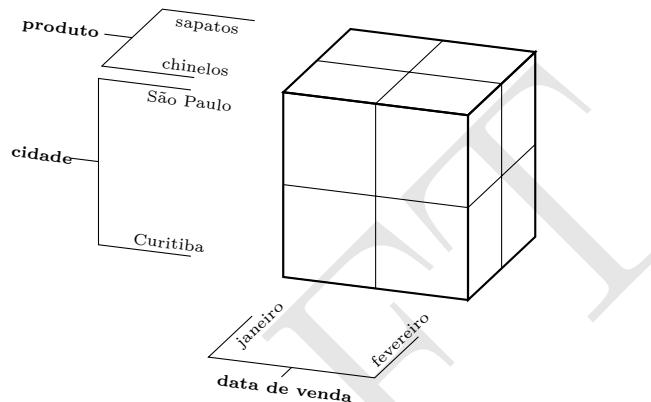


Figura 4.15: Operação de *pivot* no cubo OLAP troca a posição das dimensões. É necessário comparar com a Figura 4.2.

### 4.5.1 Pivot no Excel<sup>TM</sup>

Já diz o nome que as *pivot table* no Excel<sup>TM</sup> permitem facilmente fazer o *pivot*. Para isso basta um dado de linha para coluna e vice versa, com é demonstrado nas Figuras 4.16 e 4.17.

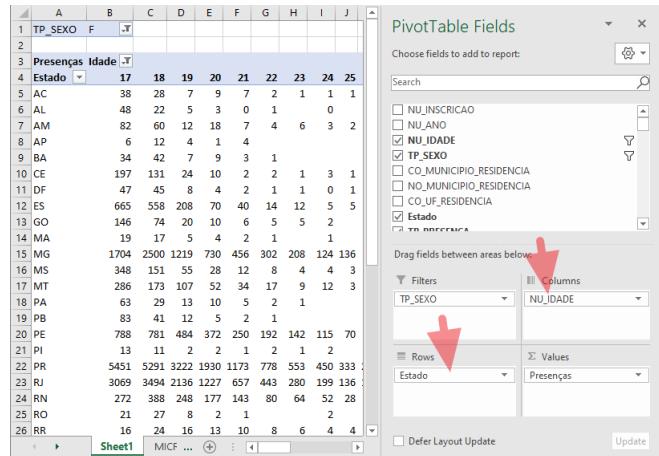


Figura 4.16: Posição dos dados antes da operação pivot.

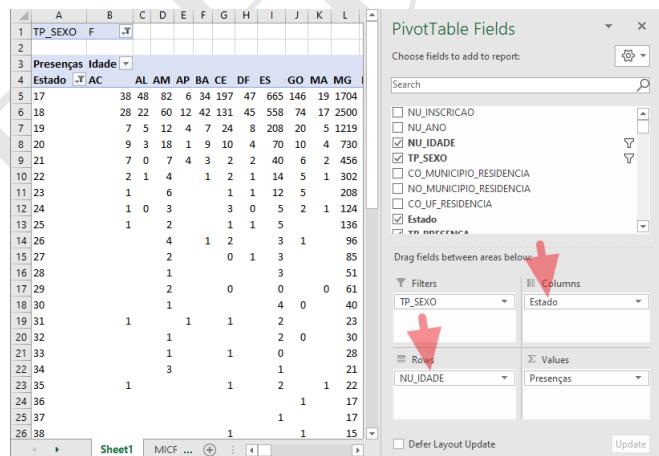


Figura 4.17: Após o pivot.

DRAFT

# Capítulo 5

## Bases Exemplo

Neste capítulo vamos mostrar dois bancos de dados de exemplo. A primeira é a base Sakila e a segunda a NorthWind.

### 5.1 O Banco de Dados Sakila

O modelo de dados da Figura 5.1, escrito na notação de Engenharia de Informação, descreve o banco de dados conhecido como **Sakila**(Oracle Corporation, 2019c). Esse banco é disponibilizado pela Oracle como um banco de dados exemplo de MySQL, licenciado com uma licença New BSD(Oracle Corporation, 2019a).

O modelo descreve uma empresa que possui várias lojas que alugam filmes, sem discutir o formato dos mesmos (DVDs?). Apesar de ser um tipo de negócio em decadência, ainda serve para o nosso propósito.

A informação central do modelo, que aparece na Figura 5.2 está na tabela que registra os aluguéis (*rental*) e na que registra os pagamentos (*payment*).

Essas tabelas se referem a basicamente três áreas: os clientes (*customer*), que alugam (*rental*) itens de estoque (*inventory*), sendo atendidos por um funcionário (*staff*). Esse itens são cópias de filmes (*film*) e os funcionários trabalham em lojas (*store*).

Os pagamentos são feitos pelos clientes aos funcionários para pagar um aluguel.

Clientes, funcionários e lojas possuem endereço (*address*), que é uma tabela única para todos. Endereços estão em cidades (*city*) que estão, por sua

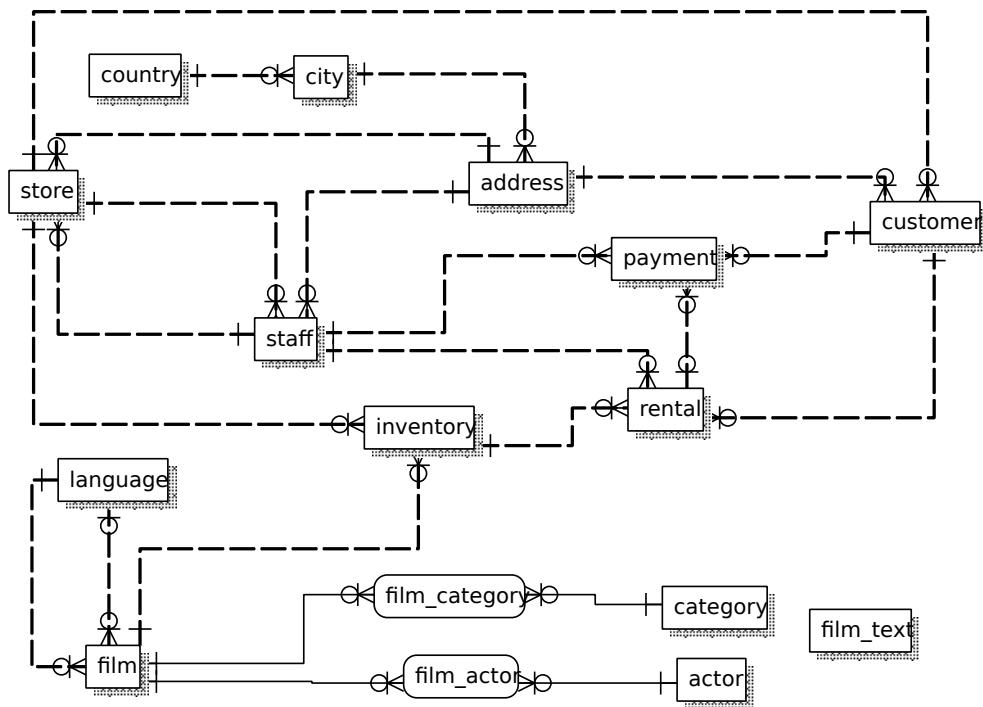


Figura 5.1: O modelo de dados da base sakila

vez, em países (*country*).

Todos os funcionários estão alocados em alguma loja, e alguns funcionários são gerentes de uma ou mais lojas.

Os filmes são classificados (*film\_category*) em categorias (*category*), e também organizados (*film\_actor*) pelos atores (*actor*) que trabalham nele. Todo filme também possui uma pela linguagem (*language*) original e linguagem disponível o que implica em dois relacionamentos distintos.

A Figura 5.2 apresenta o modelo mais detalhado, com os nomes dos campos, onde as chaves estrangeiras permitem entender como os relacionamentos foram implementados.

Com os nomes dos campos, e mesmo sem os tipos, podemos ver que a principal informação de faturamento do negócio está na tabela *payment* e se refere ao valor (*amount*) do aluguel.

É importante ver que nessa base, toda criação ou atualização de uma linha de tabela é anotada no campo *last\_update*, que aparece em todas as tabelas.

Também existe uma tabela extra que repete alguns dados dos filmes (*film\_text*).

A base Sakila contém apenas duas lojas e dois funcionários. Também possui 599 clientes, 16005 aluguéis, 16086 pagamentos, 1000 filmes, 4581 itens de inventário, 10 categorias, 603 endereços e 200 atores cadastrados.

Finalmente a Figura 5.3 mostra o modelo de dados com seus campos e os tipos dos campos. Vemos que os valores financeiros são guardados com o tipo *decimal*. As datas usam o tipo *datetime*, mas algumas informações, como *last\_update*, que aparece em todas as tabelas usam o tipo *timestamp*.

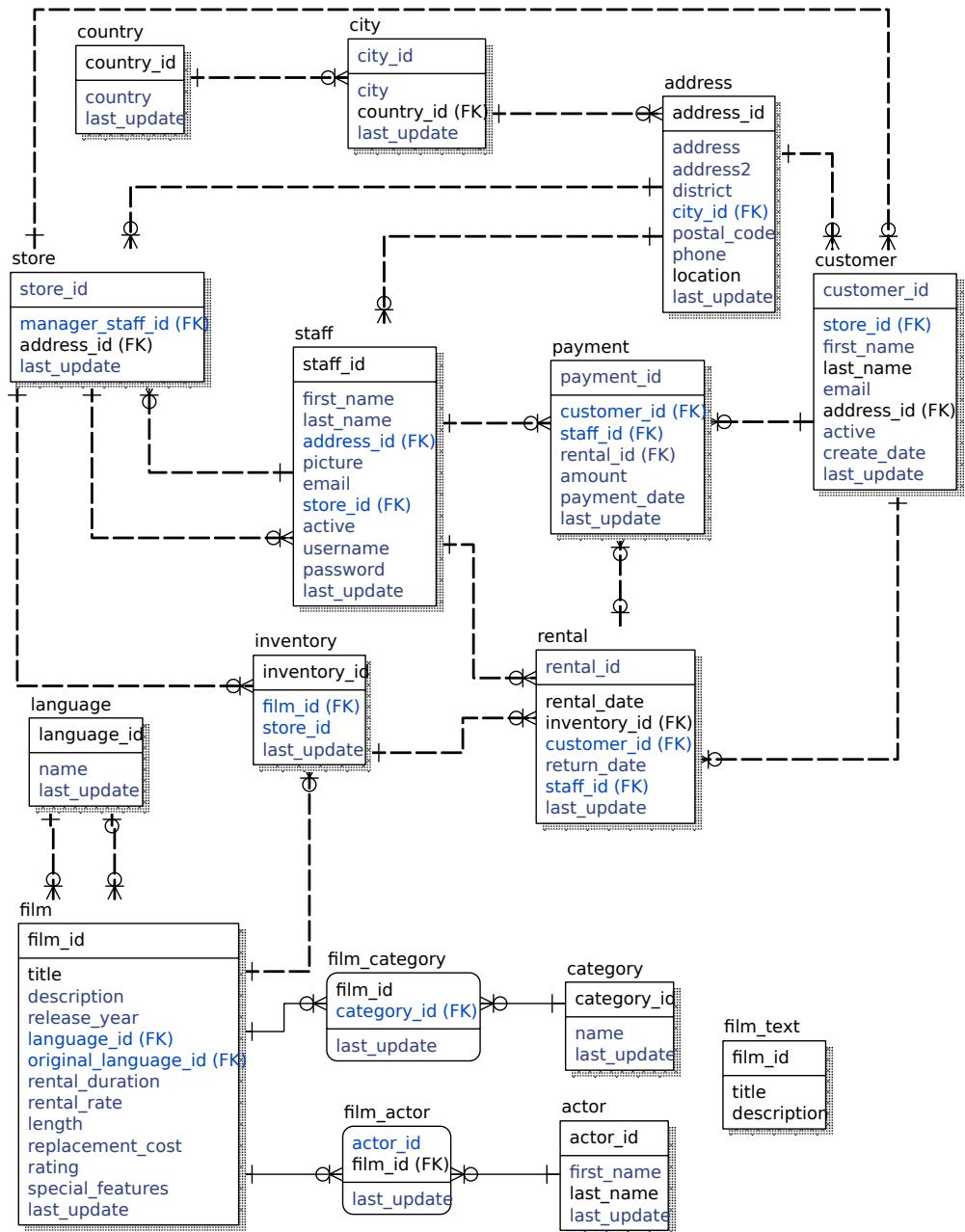


Figura 5.2: O modelo de dados da base Sakila com os nomes de campo

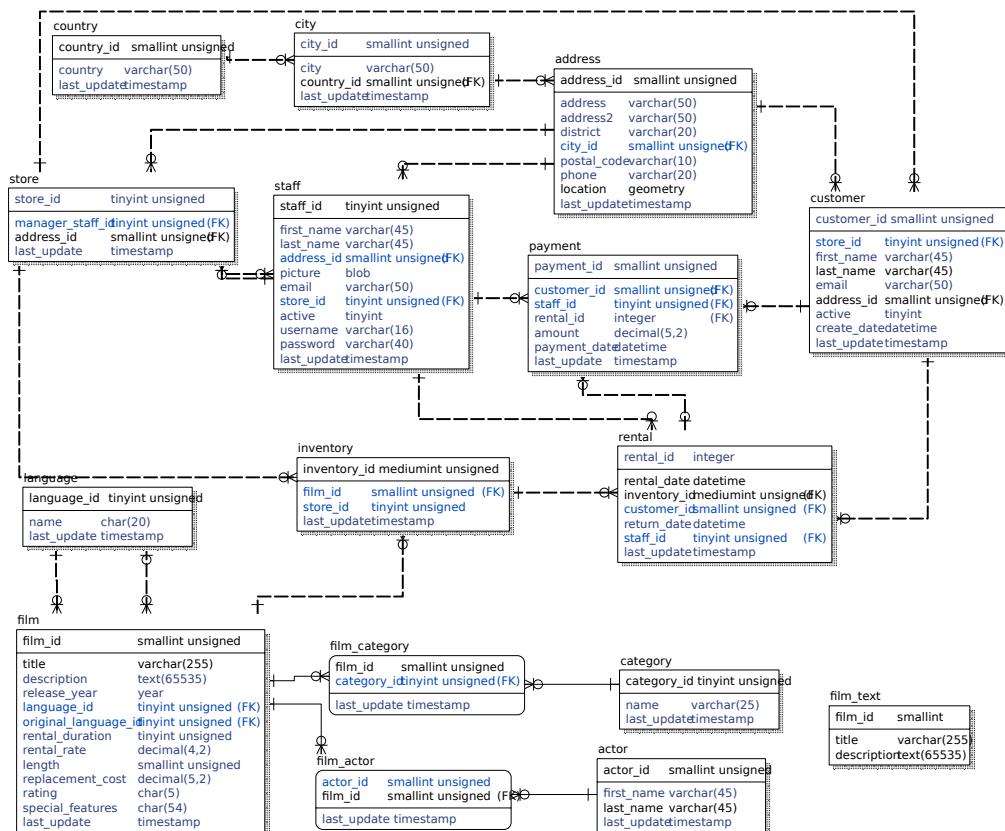


Figura 5.3: O modelo de dados da base Sakila com nomes de campo e seus tipos

## 5.2 O Banco de Dados Northwind

O banco de dados Northwind é um banco de dados fornecido pela Microsoft e que simula informações de um ERP de uma pequena empresa, oferecida pela Microsoft. A versão usada é convertida para MySQL por Scott (2016). É um banco ainda pequeno e fácil de entender, pois tem poucas tabelas.

A Figura 5.4 mostra o modelo de dados da base Northwind só com suas entidades, na notação da Engenharia da Informação.

Nesse modelo clientes (*customers*) fazem pedidos (*orders*). Os pedidos tem um estado (*order\_status*), são enviados por um transportador (*shippers*), tem um estado do imposto (*orders\_tax\_status*) e tem uma fatura (*invoices*) correspondente.

Os pedidos são compostos de itens de pedidos, que descrevem (*order\_details*) de produtos (*products*), que possuem um estado próprio (*order\_details\_status*).

Os produtos estão em um estoque, do qual entram e saem por meio de transações (*inventory\_transactions*) de certos tipos (*inventory\_transaction\_types*). Eles são comprados por meio de ordens de compras (*purchase\_orders*), que também estão descritas por seus detalhes (*purchase\_orders\_details*) e tem um estado (*purchase\_order\_status*). Ordens de compra são feitas para fornecedores (*suppliers*).

Quem faz pedidos e ordens de compra são os funcionários (*employees*), que possuem privilégios (*employee\_privileges*) de certos tipos (*privileges*).

A Figura 5.5 já mostra o modelo da mesma base com atributos.

A base Northwind oferece 45 produtos, 28 ordens de compra, 9 empregados, 48 pedidos e 29 clientes. Sua tabela mais longa é a de transações do estoque, como 102 linhas.

Já a Figura 5.6 mostra não só os atributos como os tipos dos atributos. Aqui é importante notar que os valores financeiro e quantidades estão normalmente descritos como *decimal*.

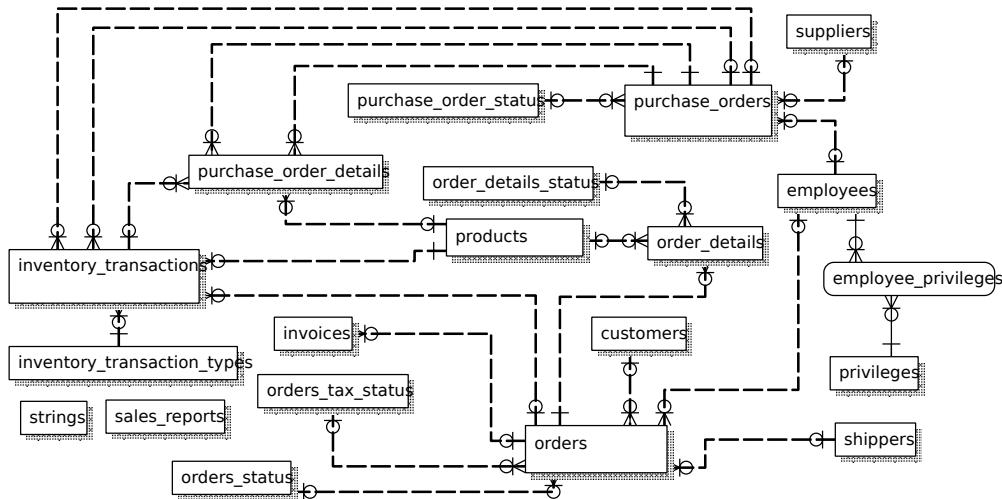


Figura 5.4: O modelo de dados da base Northwind só com as entidades

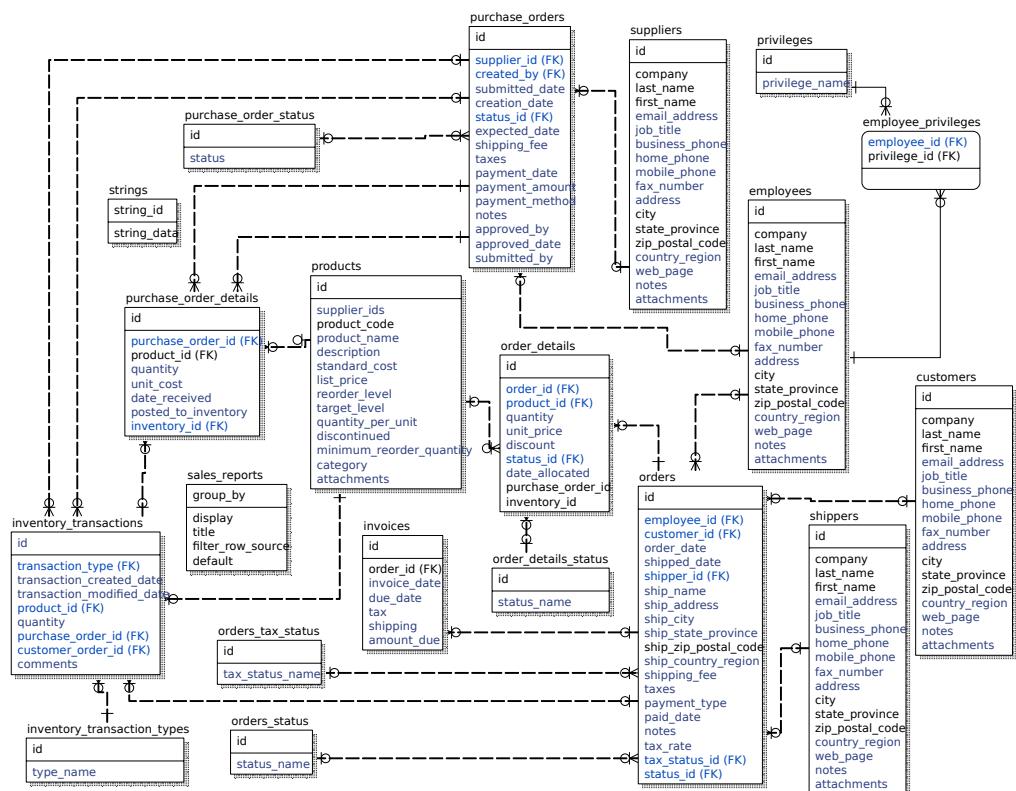


Figura 5.5: O modelo de dados da base Northwind com nomes de campo

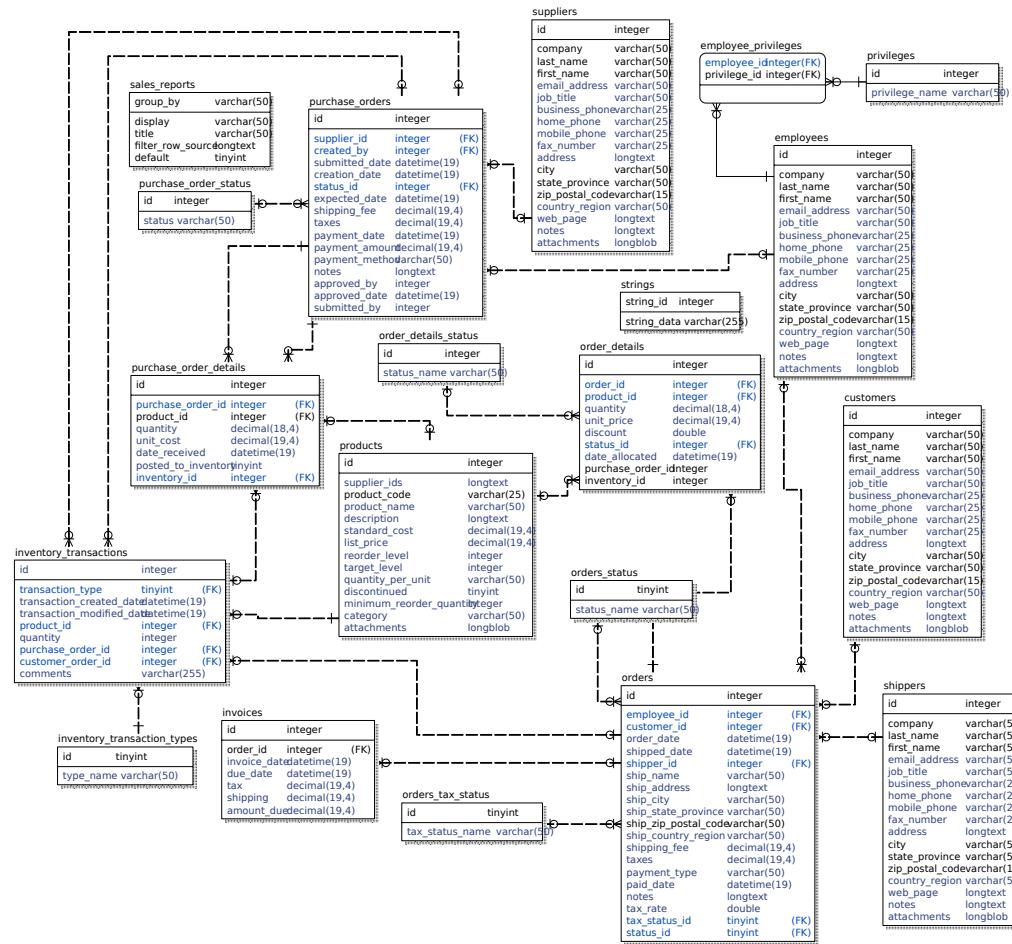


Figura 5.6: O modelo de dados da base Northwind com nomes de campo e seus tipos

### 5.3 A Cadeia de Valor da Northwind

O processo Pedir Produto, na Figura 5.7, possui 4 passos. Primeiro é necessário que alguém submeta uma necessidade de compra. A seguir a ordem de compra é criada. Criada, a ordem de compra fica esperando a aprovação pelo Vice-Presidente de Vendas. Depois dessa aprovação ela pode ser enviada para o fornecedor.

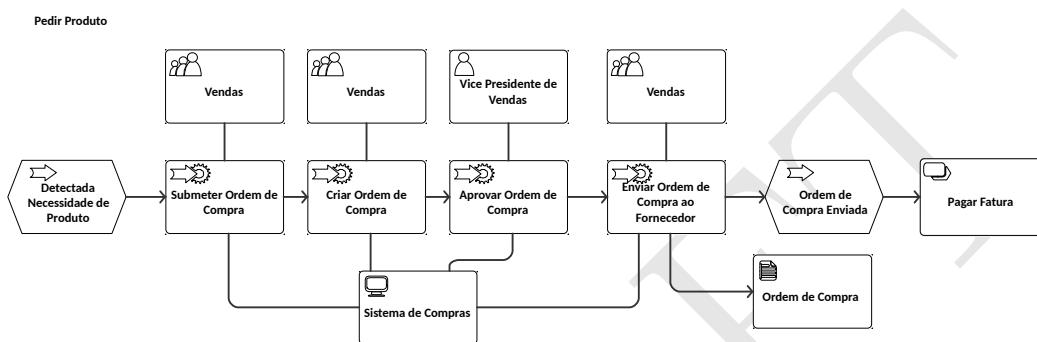


Figura 5.7: O processo Pedir Produto da Northwind

### 5.4 O Banco de Dados Adventure Works

O banco de dados Adventure Works 2014 (OLTP version) é “Uma base de dados exemplo para o Microsoft SQL Server, que substituiu a base exemplo Northwind, que foram lançadas anteriormente. O banco de dados é sobre uma fabricante multinacional de bicicletas, fictícia, chamada *Adventure Works Cycles*.” (Motl, 2019)

DRAFT

## **Parte II**

# **Modelagem de Negócio**

DRAFT

# Capítulo 6

## Modelagem de Negócio

Cada vez mais a Modelagem do Negócio vem sendo usada como uma ferramenta principal ou de auxílio do processo de desenvolvimento de software, incluindo data warehouses e aplicações relacionadas, visando o levantamento completo dos requisitos do sistema.

Nos capítulos da Parte II são tratados tópicos de modelagem de negócio, a saber:

**Organograma** uma diagrama, com muitas variantes, que apresenta a estrutura de uma organização.

**ARIS** um método de descrição de uma organização por vários modelos, sendo que o principal é um modelo de processo de negócio conhecido como **EPC**

**BPMN** um método de modelagem de processos de negócio

### 6.1 Processos de Negócio

Uma das informações importante para o conhecimento de uma organização é entender o que ela faz e como ela faz para atingir seus objetivos de negócio.

Normalmente, isso é obtido por meio da descrição de seus processos de negócio.

Um **processo** “é uma coleção de tarefas inter relacionadas que são iniciadas em resposta a um evento, e que tem como objetivo alcançar um resultado específico para o consumidor do processo” (Scheel et al., 2015).

Processos aparecem em todos os lugares, e podem ser documentados de

maneira mais ou menos formal. Um processo simples que usamos no dia a dia é o de fazer um café expresso em uma máquina de . Os passos desse processo podem ser descritos como:

1. ligar a máquina de expresso;
2. escolher a cápsula de café;
3. colocar água na máquina;
4. colocar a cápsula na máquina
5. colocar a xícara no local indicado
6. apertar o botão de fazer café
7. remover a xícara

É possível notar que mesmo um processo simples que é feito no dia a dia apresenta dificuldades de descrição. Na lista anterior, por exemplo, não se discute o que fazer se a máquina não estiver na tomada, ou se uma cápsula já está na máquina, ou se já tem água na máquina. Isso significa que uma descrição de um processo simples de forma completa pode exigir um cuidado muito grande com detalhes, se tornando, no final, muito complexa.

Todas as organizações realizam processos. Esses processos podem ter nomes como “Atender Pedido do Cliente” ou “Fabricar Peça”.

A **modelagem de processos** demonstra como funciona a organização, passo a passo, no seu dia a dia. A partir dela pode ser possível levantar os pontos a serem automatizados de um processo e como os processos realmente realizados diferem dos processos normatizados da organização.

Um **processo de negócio** “uma coleção de tarefas e atividades (operações e ações de negócio) consistindo de empregados, materiais, máquinas, sistemas e métodos, que são estruturadas de forma a projetar, criar e entregas um produto ou serviço ao consumidor”(Scheel et al., 2015).

Processos de negócio são sempre descritos a partir de um ponto de vista que indica a profundidade com que são tratados. Um processo de “Calcular o Preço Final”, por exemplo, tanto pode ser visto apenas como uma atividade única, como por várias etapas que passam por partes diferentes da organização, como ainda por um algoritmo detalhado.

Entre as características principais de processos de negócio é possível citar que eles sempre são sequências de passos, sempre respondem a uma necessidade ou demanda que define seu início e sempre chegam a um resultados específico e mensurável, seja ele o sucesso ou o fracasso do atendimento a necessidade ou demanda original.

A **Modelagem de Processos de Negócio** tem como objetivo entender

os processos que ocorrem na organização, listá-los, descrevê-los por meio de uma representação adequado, indicando o passo a passo em um nível de abstração adequado ao propósito e, também, verificar se as representações correspondem a realidade.

Linguagens de descrição de processo de negócio, como ARIS e BPMN, se preocupam em fornecer uma abstração razoável para a descrição de processos. Normalmente essa abstração permite descrições tanto de um ponto de vista muito amplo, com a supressão da maioria dos detalhes, que visa mais entender o que é feito, como um ponto de vista razoavelmente detalhado, que permite ao leitor entender realmente como o processo é executado na organização, ou seja, como algo é feito. Cabe ao modelador do processo escolher e manter esse nível de abstração ao longo do trabalho.

## 6.2 Analise de Exemplos de Processos

É interessante ver alguns exemplos de processos sendo descrito na forma de uma narrativa simples que não inclui alternativas.

O processo de entrega de uma pizza pode ser descrito como:

1. O atendente recebe o pedido de pizza e bebidas
2. O pedido da pizza passa para o pizzaiolo, que prepara as pizzas e coloca na fila para assar
3. O pizzaiolo coloca as pizzas do pedido para assar
4. Quando as pizzas estão prontas, o pizzaiolo retira e passa as pizzas para seu ajudante colocar em caixas
5. Se foram pedidas bebidas, o atendente junta as bebidas ao pedido e o separa
6. O entregador pega o pedido completo e passa no caixa para pegar o troco
7. O entregador entrega a pizza
8. O entregador cobra a pizza do cliente
9. O entregador traz o dinheiro para o caixa

Já uma loja de construção pode ter o seguinte processo para atender um pedido:

1. O vendedor recebe o pedido
2. O vendedor levanta o preço de cada produto
3. O vendedor confirma com o cliente os produtos que vão ser comprados
4. O vendedor envia a lista de produtos vendidos para o caixa
5. Enquanto o caixa faz a cobrança, O vendedor separa os produtos dese-

jados

6. Enquanto o vendedor separa o pedido, o caixa cobra o pedido do cliente
7. Enquanto o vendedor separa o pedido, O caixa emite a nota fiscal
8. O vendedor junta a nota fiscal ao pedido
9. O vendedor entrega o pedido ao cliente

O que essas duas descrições permitem entender sobre processos de negócio, e mesmo processos em geral?

Primeiro, elas descrevem um passo a passo do que é feito. Todo processo é um passo a passo para realizar algo. Segundo, elas começam com um evento. Todo processo tem que ter um motivo para começar e deve atender uma necessidade definida por esse motivo.

Outra característica importante: os passos de um processo levam a organização a chegar mais perto do resultado desejado.

Além disso, no processo da pizzaria existe um passo opcional, enquanto no processo da loja de ferragens existem passos paralelos. Tanto passos opcionais como paralelos são comuns em processos de negócio.

Finalmente, os dois são descrições parciais do funcionamento do negócio.

### 6.3 Como são Representados Processos de Negócio

Várias são as formas de representar processos de negócio. É possível dividir essas formas em dois grupos básicos: formas textuais e formas gráficas.

Entre as formas textuais encontramos os Casos de Uso, as narrativas formais ou informais, linguagens declarativas e outras.

Entre as formas gráficas, podemos encontrar formas comuns mas hoje consideradas ultrapassadas, como fluxogramas e diagramas IDEF0 e IDEF3, formas usadas na academia mas com pouco uso na área de negócios, como Redes de Petri, e formas atuais em uso no mercado, como ARIS, Diagramas de Atividade e a principal ferramenta da atualidade: os diagramas BPMN.

# Capítulo 7

## Organograma

A forma mais simples de representar uma empresa é provavelmente o organograma.

Um **organograma** é uma descrição da organização de uma empresa, amplamente divulgada, descrevendo as áreas da empresa e as hierarquias entre elas. O Organograma é ferramenta essencial na compreensão de uma empresa e suas linhas de poder.

Organogramas são diagramas que descrevem a estrutura formal de uma organização incluindo suas relações hierárquicas, normalmente por meio de linhas e retângulos. Normalmente são simples de ler, porém alguns organogramas são mais complicados, de forma a representar informações adicionais.

Normalmente um organograma tem o formato hierárquico, onde no alto está uma caixa com a posição mais importante da organização, e nos níveis abaixo aparecem caixas com seus subordinados. Cada caixa pode representar um cargo, uma função ou mesmo um departamento. A Figura 7.1 mostra um organograma simples.

Aris, uma linguagem de modelagem de processos apresentada na seção XXX, também permite representar organogramas, e isso é feito como na Figura 7.2.

Em geral o analista não precisa levantar o organograma, pois a empresa já o possui, mas é comum que haja algumas mudanças. Na verdade, não é trabalho do analista de sistemas construir o organograma da empresa, porém ele precisa conhecê-lo para melhor desenvolver o seu trabalho. Para isso é importante obter esse documento junto ao cliente, e verificar não só a hierarquia de cargos, mas também quem ocupa cada cargo, e como entrar em contato com cada um dos membros da organização que possa ter interesse

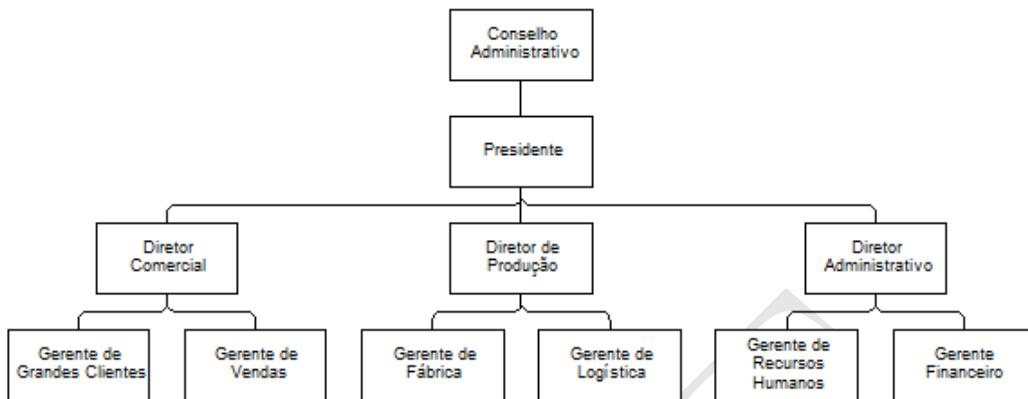


Figura 7.1: Um organograma simples, contendo apenas cargos, de uma empresa hipotética.

no sistema.

A importância de conhecer o organograma da empresa se reflete tanto na modelagem propriamente dita, pois ele fornece a descrição da empresa que será convertida para objetos do modelo, como no processo de modelagem, pois a partir do organograma é obtido o conhecimento de cargos e responsabilidades, definindo pessoas a serem entrevistadas.

Ao levantar o organograma, pode ser interessante também levantar as descrições dos cargos, se elas existirem (o que não é comum, principalmente em pequenas e médias empresas).

Este texto não tratará do processo de levantamento do organograma, pois isso é mais afeito à administração. Fica, porém, o lembrete de sua importância como documento de referência ao analista.

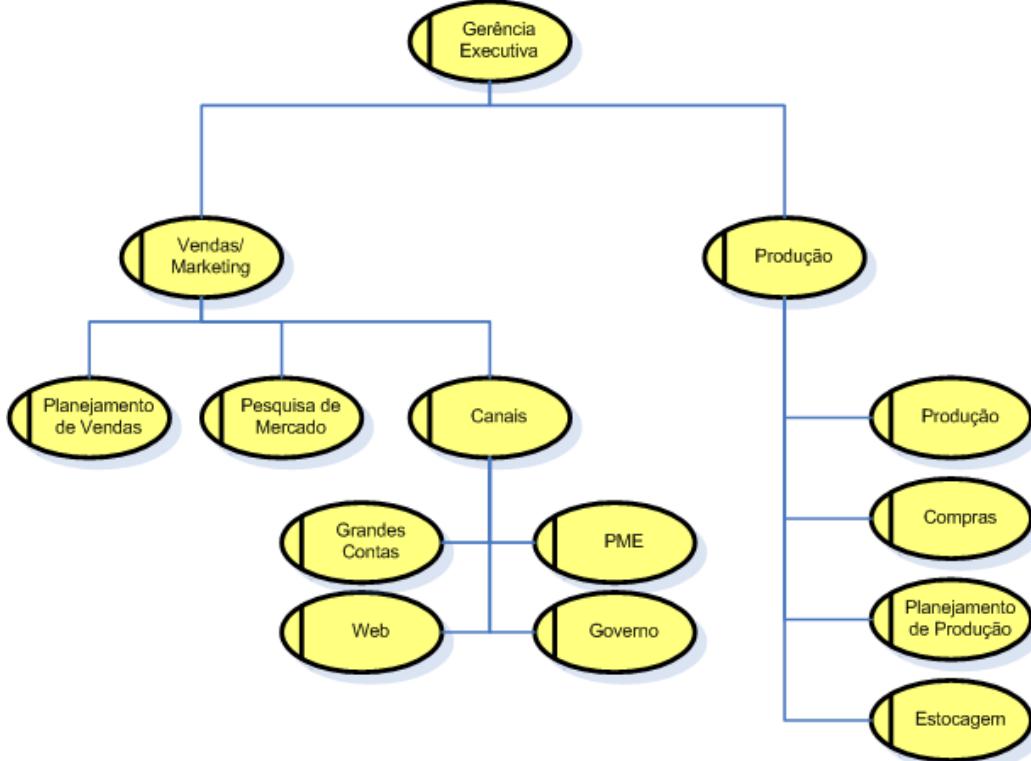


Figura 7.2: Um exemplo de um organograma de uma empresa hipotética escrito em ARIS.

## 7.1 Relações em um organograma

Um organograma pode ser utilizado para representar diferentes formas de subordinação, como a subordinação direta (onde o subordinado deve cumprir as ordens de seu chefe), a assessoria (onde o assessor fornece conselhos e pareceres) e a subordinação funcional (onde o superior pode determinar funções e métodos a outras áreas).

Normalmente a subordinação direta é representada por uma linha cheia vertical, a assessoria por uma linha cheia horizontal e a subordinação funcional por uma linha pontilhada. As Figuras 7.3 e 7.4 exemplificam esses casos.

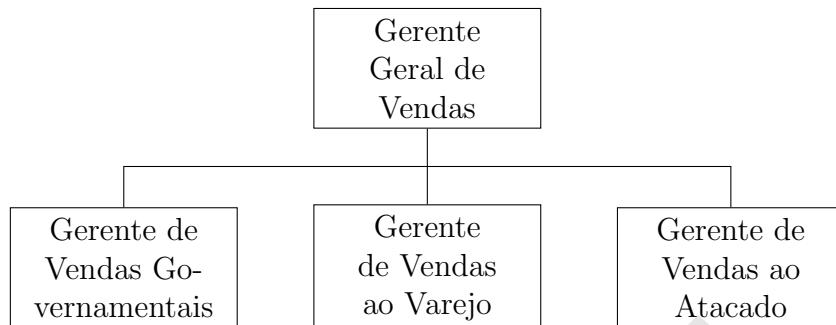


Figura 7.3: A relação de subordinação, normalmente mantida na horizontal, entre o gerente geral e seus gerentes subordinados.

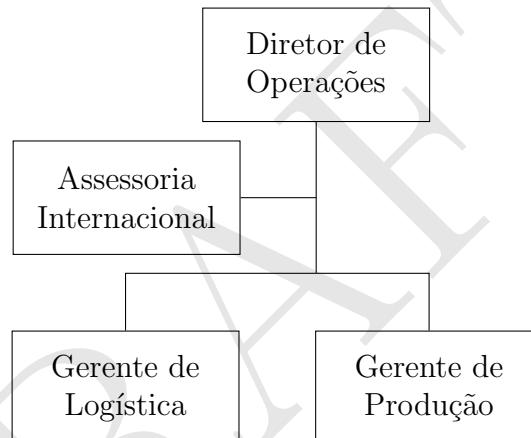


Figura 7.4: A relação de assessoria é normalmente feita com uma barra horizontal, como a Assessoria Internacional aparece no diagrama acima.

## 7.2 Outros formatos de organograma

Na grande maioria das vezes serão encontrados nas organizações organogramas clássicos ou pequenas variações dos organogramas clássicos. Existem, porém, algumas formas alternativas. Uma forma interessante, também com muitas variações, é a radial ou solar. A figura a seguir mostra um organograma radial com

Outra forma possível é inverter o organograma. O organograma da Figura 7.6, criado pelo site Guia de Direitos, mostra a estrutura do Poder Judiciário, não tratando de cargos, mas sim de partes desse poder.

Devido à grande variedade de formas usadas, organogramas podem ser desenhados tanto em sistemas específicos como em softwares genéricos de

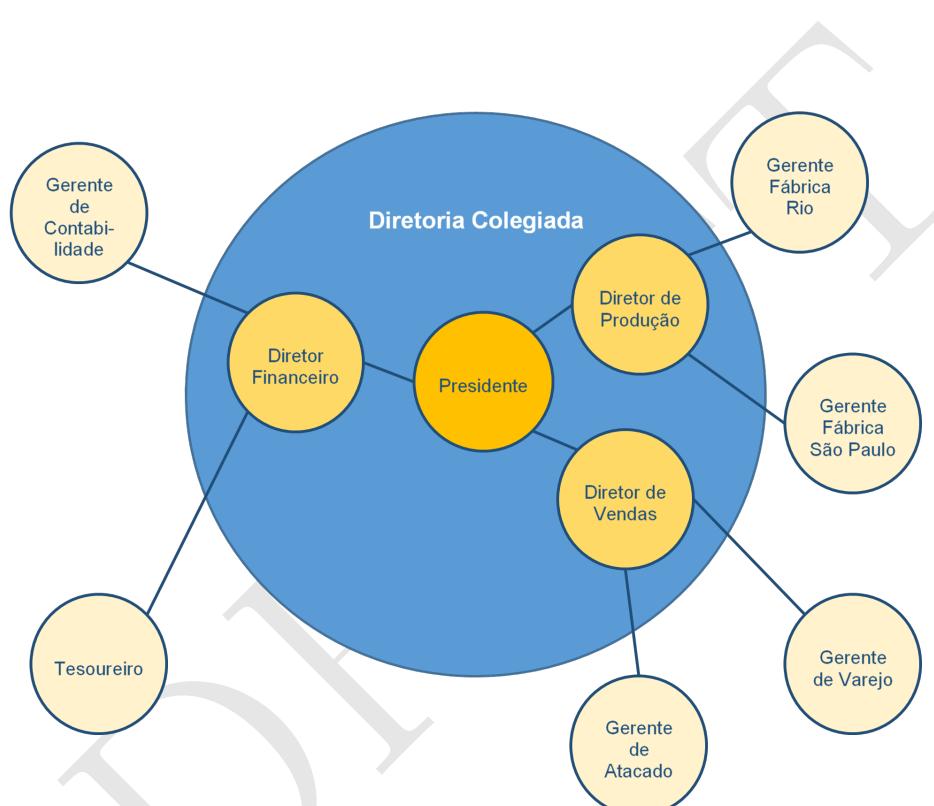


Figura 7.5: Um exemplo de organograma radial



Figura 7.6: Organograma do Poder Judiciário, em forma inversa, criado pelo site Guia de Direitos, deixado em Copyleft.

desenho.

DRAFT

DRAFT

# Capítulo 8

## Introdução ao Aris EPC

**EPC** é a sigla em inglês para **Event Driven Process Chain** (Cadeia de Processos Dirigida por Eventos). Esse método é parte do método **ARIS** usada para modelagem de processo e tem grande aceitação no mundo, estando muitas vezes associado à implantação de sistemas de ERP SAP/R3.

A Figura 8.1 mostra um exemplo de um processo imaginário de recepção e atendimento de um pedido por uma organização, contendo uma decisão. O diagrama foi desenhado usando a ferramenta ARISExpress.

Nesse método, um processo é modelado segundo um fluxo de atividades controladas por eventos. Uma leitura da Figura 8.1 dá a seguinte narrativa para o processo: quando um pedido é recebido, é necessário digitar o pedido, que é então considerado digitado; com o pedido é preciso verificar pedido, o que resulta em o pedido podendo estar correto ou não. Para pedido incorreto é feita a atividade Recusar Pedido, enquanto para um pedido correto é feita a atividade Atender Pedido. Em ambos os casos o processo acaba.

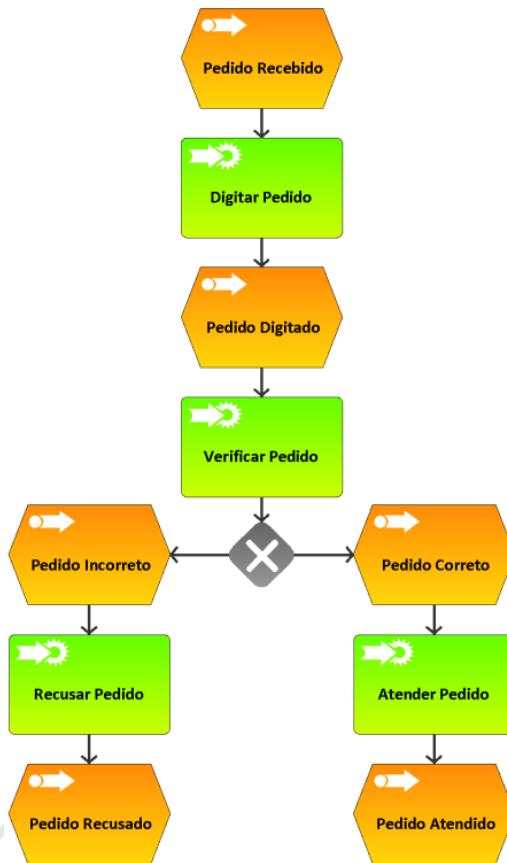


Figura 8.1: Um processo de receção e atendimento de pedido escrito em ARIS/EPC, desenhado no ARISExpress

## 8.1 Os softwares de desenho e as notações ARIS

Apesar de muito divulgado, ARIS é pouco padronizado e existem várias notações, apresentadas em livros, manuais e softwares diferentes. A empresa que mantém o método também já criou notações distintas em função do conhecimento do leitor do diagrama e fez mudanças na notação ao longo do tempo. Atualmente ela mantém pelo menos duas notações diferentes em dois produtos que suporta: o ARISExpress, grátis, e no produto oficial.

As notações mais encontradas atualmente são três, com a notação básica apresentada na Tabela 8.1:

- Uma notação antiga presente em livros e ainda suportada pelo Micro-

soft Visio.

- Uma notação feita basicamente de caixas coloridas com ícones no topo esquerdo, do software ARISExpress.
- Um notação feita de retângulos com uma marcação colorida no seu lado esquerdo, do software Aris<sup>1</sup>.

Apesar deste texto apresentar várias notações, mas é importante notar que elas não podem ser misturadas. A Tabela 8.1 dá um exemplo de três variações existentes para EPC.

Tabela 8.1: Objetos básicos do EPC, em várias versões do Aris.

Significado	Original	Aris Express	Aris Business Architect
<b>Atividade</b>	Entregar Pedido	Digitar Pedido	Preparar Ordem de Compra
<b>Evento</b>	Pedido Entregue	Pedido Recebido	Produto Entregue
<b>Regra</b>	(XOR, AND, OR)	(+, X, O)	Image
<b>Interface de Processo</b>		Process interface	

## 8.2 O Modelo Básico do EPC

O EPC mostra um **fluxo de execução**. Esse fluxo de execução é indicado por **setas direcionadas** em um grafo compostos por nós de vários tipos, como mostra a Figura 8.1.

O fluxo de execução, ou **fluxo de controle**, indica a sequência dos passos descritos no diagrama. Por ele **não trafegam materiais ou informação**. Essa é uma diferença importante entre o EPC e outros modelos mais antigos de fluxograma que misturaram o fluxo de dados com o fluxo de execução.

<sup>1</sup>Esse software pode ser obtido sem custos, por meio de uma licença acadêmica, para alunos e professores

É possível dizer que o diagrama EPC é um grafo direcionado, com vários tipos de nós. Os principais nós desse diagrama são:

- Atividades, também chamadas de funções ou tarefas.
- Eventos
- Conectores
- Interface de Processo

Como diz o nome, o EPC é um cadeia de processo, ou seja, uma sequência de atividades que forma um processo, controlada por eventos. Isso significa que os processos acontecem, ou não, de acordo com os eventos.

É importante notar que Atividades, Eventos e Interfaces de Processo só podem possuir **apenas um** fluxo de entrada, ou seja, apenas uma seta entrando nelas, e **apenas um** fluxo de saída, ou seja, apenas uma seta saindo delas.

Apenas os conectores podem ser usados para tratar as principais formas de variar os fluxos de execução: a concorrência ou paralelismo de caminhos (noção de E) e a seleção de um caminho entre muitos (noção de XOR).

### 8.2.1 Atividades

**Atividades**, também são chamadas de funções ou tarefas. Elas representam passos do processo que precisam ser executadas e suportam um ou mais objetivos de negócio. Não existe processo que não tenha pelo menos uma atividade.

Na prática, atividades podem representar: atividades tangíveis, decisões (mentais), processamento de Informações (automatizadas ou não), ou um subprocesso que é descrito por outro diagrama EPC<sup>2</sup>.

Como atividades registram ações realizadas dentro de um processo, elas possivelmente são iniciadas ou iniciadas ou habilitadas por eventos, geram eventos, consomem recursos e exigem gerenciamento, tempo, e atenção.

O nome de uma atividade é sempre da forma: *verbo no infinitivo*; *objeto direto*, como em “Remover detritos” ou “Registrar Pedido”.

Sua representação tradicional é apresentada na Figura 8.2.

Uma atividade não é completamente definida pelo seu nome. Outros elementos que aparecem no diagrama, se serão apresentados ainda nesse

---

<sup>2</sup>Nesse caso, as ferramentas CASE fornecem alguma marca indicativa de que a função pode ser expandida.



Figura 8.2: Formas de representação de uma atividade em diferentes versões do Aris.

capítulo, detalham sua descrição, como:

- Informação que entra
- Objetivos
- Recursos Consumíveis, como matéria prima
- Recursos não consumíveis, como máquinas
- Procedimentos realizados
- Capacidades necessárias
- Informação que sai
- Produto ou serviço entregue

Como atividades representam ações, elas sempre envolvem algum gasto de tempo, por mais rápido que seja, e algum recurso, por menor que seja.

Na Figura 8.1 são encontradas quatro atividades: Digitar Pedido, Verificar Pedido, Recusar Pedido e Atender Pedido.

### 8.2.2 Eventos

**Eventos**, representam situações, ou estados do sistema, antes ou depois da execução de uma função. Não existem processos sem pelo menos dois eventos: um de início do processo e um do fim do processo.

Os Eventos representam a mudança do estado do mundo enquanto o processo é executado. Três são os tipos de eventos candidatos a serem registrados em um processo :

1. A mudança de estado do mundo que faz com que o processo comece;
2. Mudanças internas de estado enquanto o processo é executado, chamados de eventos intermediários, e
3. O resultado final do processo que tem um efeito externo.

Pelas regras do EPC, **todo diagrama deve começar em um evento**, que indica porque ele começou, e **acabar em um ou mais eventos**, que

normalmente indicam o sucesso ou o fracasso do processo em atender ao evento que iniciou o processo. Um diagrama não pode ter outro tipo de nó como final de um fluxo de execução.

Uma definição para **eventos** no EPC é “estados comercialmente relevantes, que controlam ou influenciam a progressão subsequente de um ou mais processos de negócio”(Software AG, 2019a). Não se pode deixar de criticar o uso da palavra evento nesse contexto, quando sua definição mais se assemelha a um estado do sistema.

Em relação as atividades conectados com eles, cada evento pode representar os gatilhos ou os resultados das mesmas(Software AG, 2019a). Porém, em um modelo de processo, a ocorrência de um evento não indica que a atividade a seguir vai ocorrer imediatamente, ela é apenas permitida. Claramente, eventos definem pré-condições para que uma atividade ocorra, e também pós-condições ao fim da atividade.

É importante notar que eventos funcionam como marcos, relacionados com um instante no tempo(Software AG, 2019a), logo eles não gastam tempo ou recursos.

Na Figura 8.1 são encontrados seis eventos: Pedido Recebido, Pedido Digitado, Pedido Incorreto, Pedido Correto, Pedido Recusado e Pedido Atendido.



Figura 8.3: Formas de representação de uma evento em diferentes versões do Aris.

Assim, na Figura 8.1, o processo começa quando um pedido é recebido e pode acabar de dois modos: com o pedido recusado ou com o pedido atendido. Analisando mais o diagrama é possível perceber que o pedido é recusado porque ele estava incorreto após a atividade “Verificar Pedido”, e é atendido se estiver correto após essa atividade.

Um evento segue a sintaxe: <Sujeito><Verbo na voz passiva>. Ou, mais simplesmente: <Substantivo><Adjetivo>. Essa sintaxe é usada nos exemplos da Figura 8.1.

Os eventos intermediários, isto é, os que aparecem dentro do processo,

devem ser escolhido por dois princípios básicos: de forma a cumprir seu papel de explicar uma tomada de decisão, isto é, sempre aparecem depois de todas as regras OU-Exclusivo, indicando o caminho a ser tomado; e também para indicar mudanças significativas de estado. No passado, como cada atividade implica na mudança do estado do processo, era obrigatório colocar um evento entre cada par de atividades. Com o tempo, porém, se percebeu que não havia utilidade em colocar eventos muito simples e óbvios e essa prática deixou de ser exigida. Ainda é importante, porém, usar os eventos para registrar mudanças importantes no estado do processo, ainda marcos de controle para a gerência, ou ainda outro motivo que torne aquele estado interessante. A Figura 8.4 mostra como era obrigado a desenhar um processo no passado e uma alternativa válida no presente.



Figura 8.4: Exemplos de eventos Segundo a versão original de EPCs, sempre deveria haver um evento entre dois processos. Atualmente é permitido que uma sequência de processos não tenha nenhum evento entre eles. Logo, os diagramas das figuras a seguir podem ser usados como equivalentes.

### 8.2.3 Regras

**Regras**, “representam operadores que são usados para especificar as conexões lógicas entre eventos e atividades”(Software AG, 2019b). Eles permitem a unificação e separação de fluxos segundo os conceitos de E, OU ou OU-exclusivos.

A linguagem oferece três tipos de regras básicas: o que permite que o fluxo de execução siga vários caminhos simultaneamente, conhecido como E, o que permite que um caminho seja escolhido entre vários, conhecido como OU-EXCLUSIVO, e o conector OU, que permite escolher um ou mais

caminhos, mas que por sua ambiguidade não é considerado aconselhável e não será usado neste texto.

Esses conectores E, OU e OU-EXCLUSIVO funcionam de duas formas: como divisores de caminho (split) e como junção de caminhos (join). Não é possível utilizar um conector simultaneamente nas duas formas, o que é indicado na Figura 8.5.

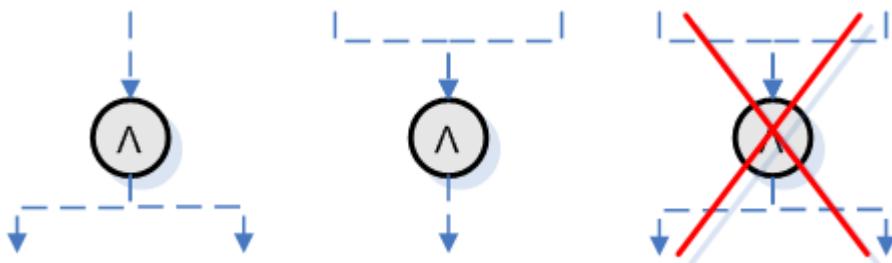


Figura 8.5: Um conector só deve ser utilizado na forma de divisor ou junção, e nunca das duas formas ao mesmo tempo.

De acordo com as regras sintáticas para EPCs, é possível que, em um processo, uma atividade produza um ou mais eventos, sejam eles simultâneos ou uma escolha. Porém o diagrama não pode ligar diretamente atividades a vários eventos, ou eventos a várias atividades. Então são usadas as regras para indicar de que forma isso ocorre.

A Figura 8.1 apresenta um conector que indica uma decisão (OU-Esclusivo ou XOR). Essa decisão acontece, na verdade, na atividade imediatamente anterior ao símbolo da regra. Nessa decisão, um dos caminhos apenas será seguido, de acordo com o estado atual do processo, isto é, se a situação indica um “Pedido Correto” ou um “Pedido Incorreto”.

#### 8.2.4 Interface de Processo

**Interface de Processo**, são símbolos de conexão entre diagramas de processo. Elas servem para indicar que um processo, após terminar, tem alguma continuação em outro processo, ou, vice-versa, que um processo começa como a continuação de outro processo.

As **interfaces de processos** são usadas para indicar que um processo acontece quando outro processo acaba. Caminhos são sempre conectados a eventos, devendo aparecer antes ou depois deles, e recebem o nome do processo origem ou destino. Na Figura 8.7 a seguir é possível ver um exemplo de sua utilização.

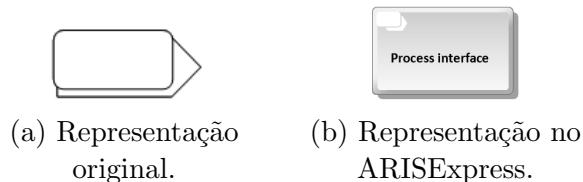


Figura 8.6: Formas de representação de uma interface de processo em diferentes versões do Aris.

Elas só devem ser usadas para ligar processos de um mesmo nível.

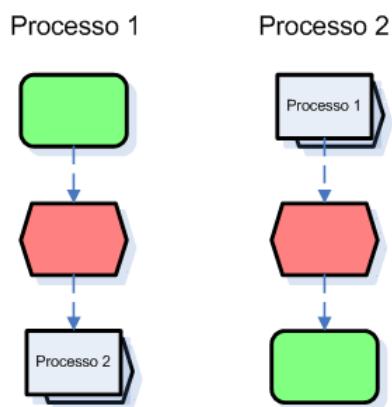


Figura 8.7: Indicando que um processo é continuação lógica de outro processo. No trecho final do “Processo 1” é indicado que ele é seguido pelo “Processo 2”, e nesse processo é indicado que o caminho é proveniente do “Processo 1”.

### 8.3 Exemplos Adicionais de EPC

Seguindo essas regras básicas, o processo de entrega de pizza descrito na Seção 6.2 pode ser representado como na Figura 8.8.

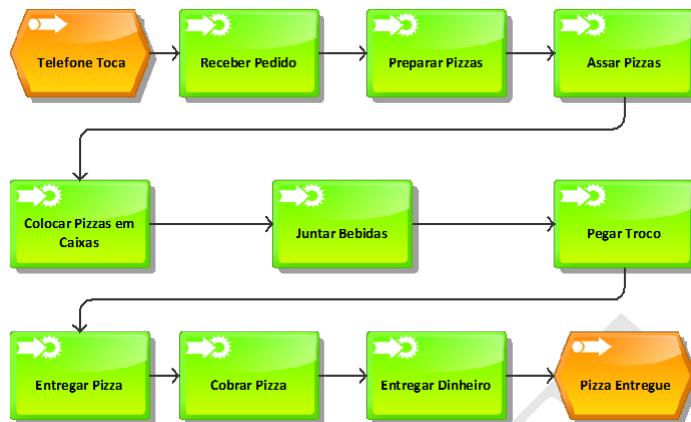


Figura 8.8: Processo simplificado de entrega de pizzas.

## 8.4 eEPC

eEPC é a sigla em inglês para Extended Event Driven Process Chain (Cadeia de Processos Dirigida por Eventos). Nessa extensão é possível declarar mais algumas informações sobre o processo sendo descrito.

Esses elementos adicionais funcionam basicamente como comentários ao processo que está sendo documentado. Assim, depois de descrito o processo pelo método não estendido, colocamos sobre eles novos elementos documentando informações como quem realiza o processo, que informação utiliza, que produtos gera ou consome, etc...

Os principais elementos adicionais em um eEPC, segundo a documentação original, são:

- Unidades Organizacionais, que representam departamentos envolvidos em um processo.
- Papéis e Pessoas, que representam pessoas ou papéis envolvidos em um processo.
- Informação ou dados, que representam informação utilizada ou gerada em um processo.
- Produtos ou serviços, que são gerados ou consumidos pelo processo.
- Objetivos, que representam o motivo da realização de um processo ou tarefa

Versões mais modernas inserem vários outros tipos de informações. No ARISExpress, por exemplo, estão disponíveis dez tipos de objetos que podem ser usados, como mostrado na Figura 8.10.

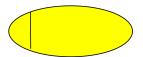
Tipo	Símbolo
Unidade Organizacional	
Informação	
Pessoa ou Cargo	
Fluxo de Informação	
Relações Organizacionais	
Produto ou Serviço	
Objetivo	

Figura 8.9: Elementos complementares dos diagramas eEPC.

A Figura 8.11, apresenta um exemplo do mesmo processo da Figura 8.1 para demonstrar os usos dos elementos adicionais.

#### 8.4.1 5W2H no EPC

Um evento indica quando (*when*) algum processo, função ou tarefa deve ser iniciado. Uma atividade indica o quê (*what*) deve ser feito.

Uma unidade organizacional, um papel ou uma pessoa indicam quem (*who*) deve fazer.



Figura 8.10: Nova lista e imagens dos elementos complementares dos diagramas EPC no ARISExpress.

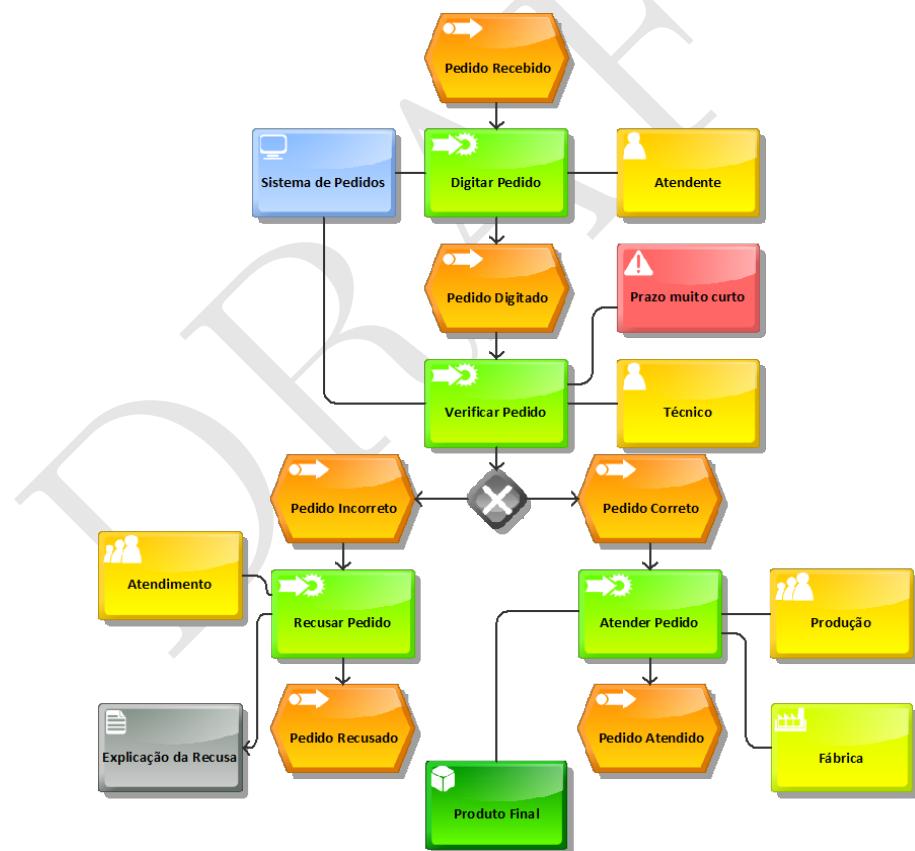


Figura 8.11: Exemplo de eEPC, a partir do EPC anterior da Figura 8.1. Desenho feito no ARISExpress. Os símbolos suplementares podem variar de acordo com a ferramenta case sendo usada.

## 8.5 Passos para construir modelos EPC/EPCE[B23]

1. Identifique os eventos que iniciam as funções, que servem como gatilhos para o processo se iniciar. Normalmente vem de “fora para dentro” do processo. Identifique as funções do processo, associando-as aos eventos que as iniciam e sua seqüência.
2. Decomponha as funções, verificando se são ações lógicas simples ou compostas, executadas por uma ou mais pessoas (ou ainda um sistema de computador). Verifique também se a função é uma transação isolada ou pode ser dividida em partes, se pode ser interrompida em um momento específico e se existe um evento que a interrompa ou que a faça funcionar novamente.
3. Analise os eventos novamente, definindo-os e refinando-os se necessário. Garanta que são necessários e suficientes para iniciar a função. Analise se existem casos especiais nos quais as funções acontecem ou não. Use operadores lógicos para montar as relações entre os eventos.
4. Identifique os eventos de finalização e as saídas (tanto de material quanto de informação). Procure identificar quem processos e pessoas no resto da organização que dependem do processo sendo analisado. EPCs podem ser muito pequenos ou enormes, dependendo unicamente do tamanho do processo que está sendo mapeado.

### 8.5.1 EPC e Loops/Laços

Por diferentes motivos, laços arbitrários não funcionam bem com EPCs. Alguns textos explicitamente proíbem laços, outros não apresentam exemplos, desconsiderando-os implicitamente. Porém, é possível descrever laços em EPML, a linguagem XML para descrever EPCs.

A questão de modelar laços em PEC é relacionada ao fato que processos de negócio não deveriam ter laços, já que eles devem levar o negócio em direção a solução da necessidade definida na sua origem.

Na prática, isto depende do ponto de vista do modelador e do detalhe que quer dar ao modelo.

Assim, devem ser seguidas seguintes recomendações quanto ao uso de laços nos diagramas EPC:

- Analise a necessidade de laços em função do ponto de vista da modelagem
- Evite os laços se o ponto de vista for mais abstrato

- Use apenas os laços estritamente necessários, não se preocupe com correções de erros simples
- Use apenas laços simples, baseados em XOR
- Verifique sempre se o laço representa a realidade do processo e se ele é significativo.

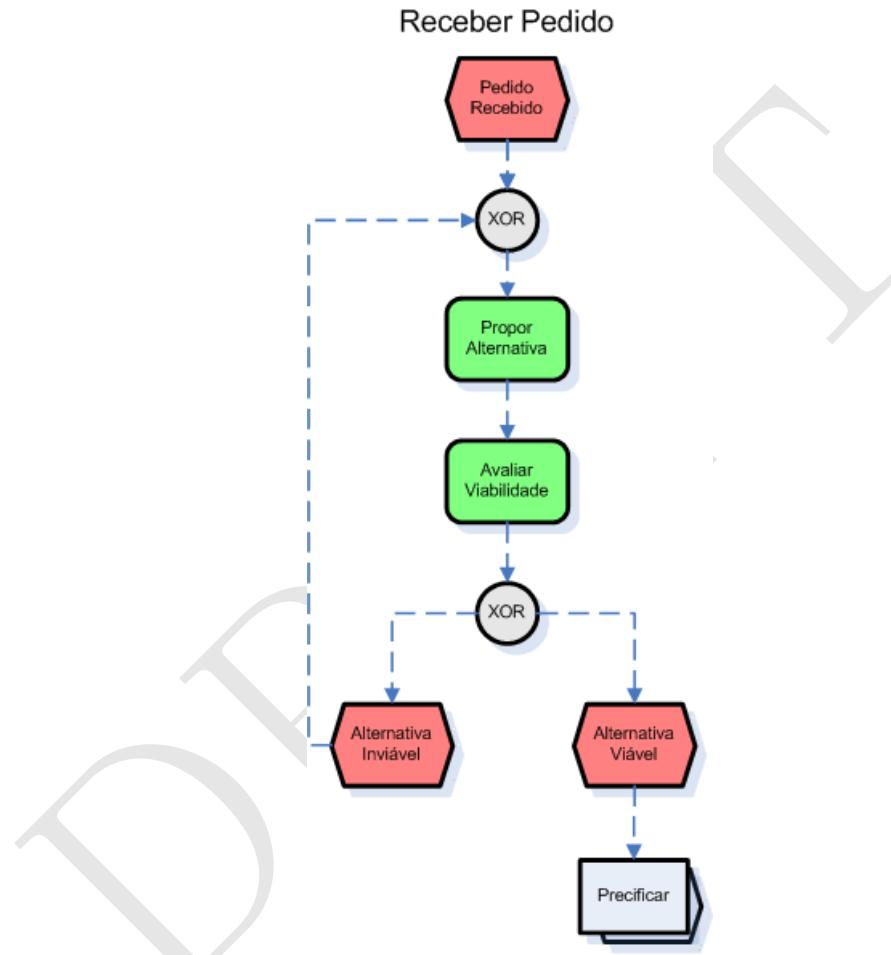


Figura 8.12: Um EPC válido, com um loop simples baseado em XOR. Também é dado um exemplo de como conectar esse EPC a um outro processo.

## 8.6 Deadlocks

O uso descuidado de regras do tipo OU-Exclusivo e E sem estruturá-las corretamente gera o risco de deadlocks, isto é, de um estado em que o processo

não pode continuar porque um caminho está esperando algo que não pode acontecer.

A Figura 8.13 demonstra um exemplo onde é impossível terminar o processo.

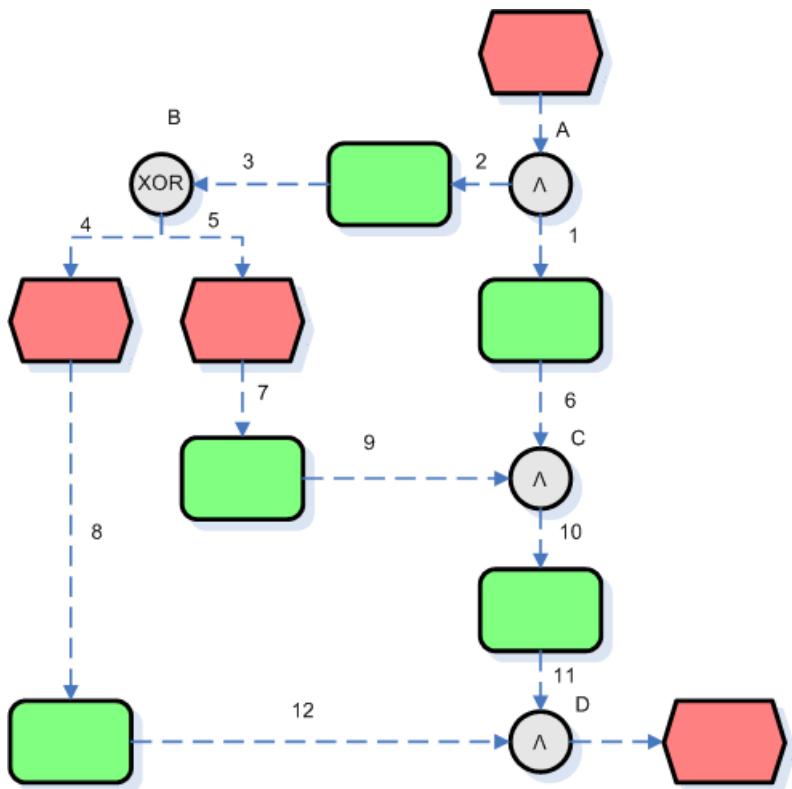


Figura 8.13: O modelo acima, demonstra um deadlock. Como comentário, os números identificam as setas e as letras identificam os conectores . No conector XOR (B), um dos caminhos 4 ou 5 será escolhido. Porém, se for escolhido o caminho 5, o sistema para de funcionar no conector E (D), pois esse E nunca será satisfeito (já que 4-8-12 não será percorrido). Caso seja escolhido o caminho 4, o conector E (C) nunca será satisfeito (pois 5-7-9 não será percorrido), e o sistema vai parar em (C), esperando 9, e (D), esperando 11.

## 8.7 Algumas padrões em EPC

Alguns padrões ocorrem em muitos processos.

Um padrão comum é o de fazer algo que é avaliado em outro passo do processo. A Figura 8.12 mostra um exemplo onde uma alternativa é proposta e depois é avaliada, e se não for aceita outra alternativa deve ser proposta. Esse é um dos casos onde o *loop* pode ser mostrado corretamente.

O segundo padrão é escolher um caminho entre duas alternativas. A Figura 8.14 mostra como esse padrão deve ser desenhado. Esse é um padrão extremamente comum, e o número de opções **não** está limitado a dois. O importante é que as condições que permitem escolher o caminho aparecem nos eventos, logo após o símbolo de Ou-exclusivo, que vem logo após a atividade que toma a decisão.

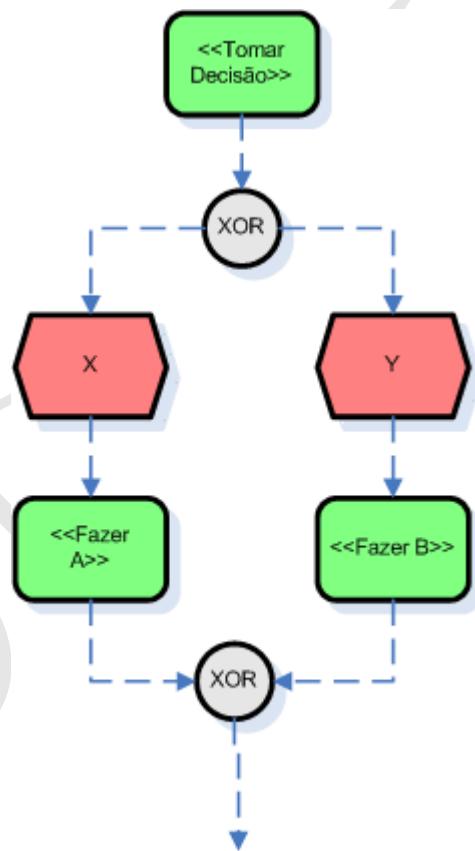


Figura 8.14: Descrição EPC para “Se X então Fazer A, Se Y, então Fazer B”

Caminhos paralelos também são um padrão comum, que são descritos como na Figura 8.15.

Outro padrão é fazer algo se alguma condição não foi alcançada, caso contrário seguir em frente. Isso acontece, por exemplo, se antes de atender

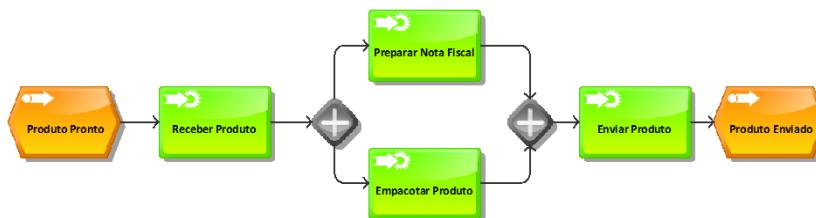


Figura 8.15: Caminhos paralelos, na notação do ARISExpress

um cliente é necessário cadastrá-lo. A Figura 8.16 dá um exemplo desse caso. Esse padrão também é conhecido como “ramo que não faz nada (*do nothing branch*)”.

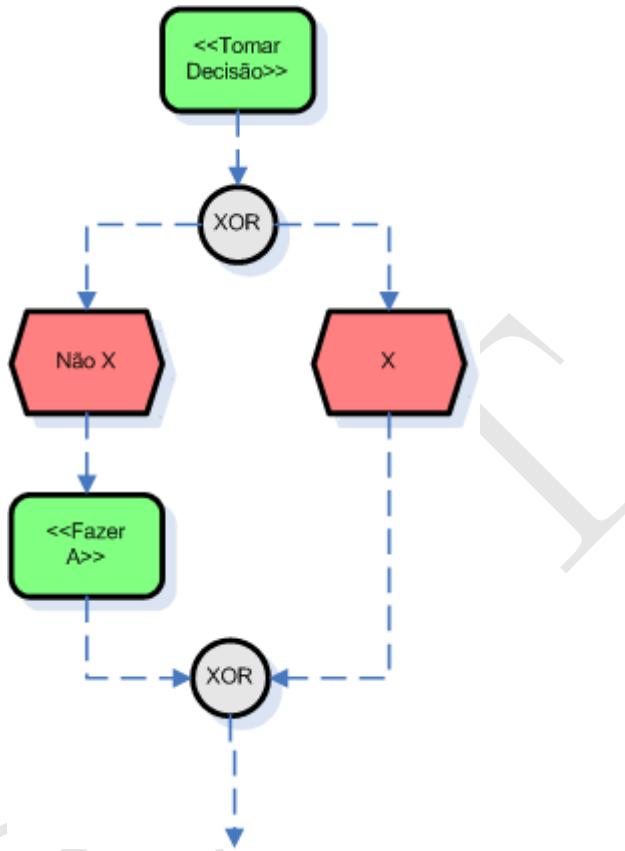


Figura 8.16: Descrição EPC para “Se Não X então Fazer A”

## 8.8 Revisão das regras básicas do EPC

- Não existem nós isolados
- Atividades e eventos têm apenas uma entrada e uma saída
- Operadores lógicos contêm vários fluxos de entrada e um de saída, ou um único fluxo de entrada e vários de saída. Não podem conter vários de ambos.
- Loops devem ser evitados.
- Dois nós só podem possuir um único link entre eles
- Existe um evento inicial e um evento final
- Eventos não tomam decisões, logo só possuem uma saída.
- Evite o uso de OU, pois sua semântica não é clara.
- Cuidado com modelos complexos, que podem levar a deadlocks.
- Mantenha o diagrama estruturado. Sempre que dividir um caminho com XOR ou E, feche o caminho com o mesmo símbolo.
- Não misture notações.

**Parte III**

**Data Warehouse**

DRAFT

# **Capítulo 9**

## **A Situação das Informações nas Organizações**

### **9.1 A necessidade de informação das organizações**

Todas as organizações têm a necessidade de controlar e gerenciar todas as informações que dispõem sobre si mesmas e sobre o mercado, de modo a permitir que suas decisões produzam ações que gerem benefícios e evitem prejuízos, de curto, médio e longo prazo, nos níveis operacional, tático e estratégico.

As informações necessárias para a organização aparecem em todo seu ambiente. Cada ação realizada, cada decisão tomada, gera, ou pelo menos deveria gerar, um registro que pode ser utilizado mais tarde para a tomada de decisões. Por exemplo, cada venda feita por uma cadeia de lojas de varejo pode ser incluída em um gráfico ou relatório que permita entender o desempenho das vendas por local, por produto, por vendedor e até mesmo pela hora do dia. Em um uso mais avançado, esses dados podem ser correlacionados, com o processo de seleção da área de Recursos Humanos, de maneira a tentar desenvolver um perfil do vendedor a ser contratado, tanto de forma geral como para atender necessidades específicas. Com relação aos dados de mercado, podem ser usados para prever o desempenho de uma nova loja em uma localidade específica, de acordo com dados demográficos ou informações sobre outros tipos de comércio ou mesmo de consumo de água ou eletricidade.

Para isso ser possível é necessário informatizar a organização, isto é, im-

plantar dentro da organização sistemas de informação que gerenciem essas informações.

## 9.2 Processo de Informatização nas Organizações

Frente a importância da informação para o sucesso da organização, é fácil entender que, ao longo de sua vida, ela deve passar por um processo contínuo de informatização, a fim de atender as demandas cada vez maiores tanto da própria organização, quanto de seus fornecedores, parceiros, clientes e também do Estado.

Esse processo é necessário e geralmente benéfico, mas não sem problemas.

Entre os principais problemas do processo de informatização das organizações é que ele normalmente é de crescimento vegetativo, *bottom-up* e por demanda imediata. Os sistemas são criados um a um, em função de necessidades específicas trazidas pelas partes interessadas, e construídos de modo a atender principalmente os requisitos dessas partes, muitas vezes desconsiderando como as atividades e informações desses sistemas podem ser úteis para o resto da organização.

Por outro lado, tentativas de criar grandes sistemas integrados que atendam toda a organização parecem estar fadados ao fracasso. O tamanho de um sistema é um fortíssimo fator de risco (Pressman e Maxim, 2016), já que a relação do tamanho com a taxa de fracasso é mais que linear.

Como sempre, as melhores soluções, ou pelos menos as soluções mais viáveis, parecem estar em um meio termo, ou pelo menos em decisões tomadas de forma consciente dos riscos e benefícios de cada opção.

Não se pode também deixar de levar em consideração que organizações diferentes têm gestões diferentes desse processo. Enquanto uma pode escolher os sistemas de informação a serem implantados de forma *ad-hoc*, outra, mais corretamente, pode possuir um planejamento estratégico que indica em que direção a área de Tecnologia da Informação deve seguir e que sistemas serão estratégicos para o seu progresso.

Esse quadro bastante variado, evoluiu muito tecnologicamente e no entendimento do negócio informatizado nos últimos anos. Alguns grandes marcos, como o aparecimento do computador comercial, o aparecimento do computador pessoal, o aparecimento das redes locais e finalmente o aparecimento da Internet e das tecnologias em nuvem, causaram grandes mudanças na forma de pensar a organização, criando até mesmo a possibilidade de organizações

virtuais.

Ao logo desse tempo houve um aprendizado que trouxe soluções bastante adequadas ao problema de gestão das organizações, como o uso de sistemas **ERP** e a compra de software **COTS**.

**COTS** significa **Commercial Of The Shelf**, e indica software pronto que fornece funções específicas com nenhuma ou pouca adaptação para quem o compra ou implementa. Principalmente pequenos negócios e profissionais liberais podem ter quase todas suas necessidades atendidas por meio de esse tipo de software. Algumas áreas de aplicação, mesmo em empresas grandes, podem se beneficiar desse tipo de produto. Deve ficar claro, porém, que nesse caso a organização deve se adaptar as práticas implementadas no produto.

Já os sistemas **ERP**, **Entreprise Resource Planning**, são na prática sistemas de gestão empresarial customizáveis destinados a integrar fortemente as áreas de negócio. Sua principal característica é funcionar em áreas do negócio cuja tarefa é bem definida, e cuja experiência de desenvolvimento de software é muito forte, e seu maior sucesso vem da implementação, já no software, das melhores práticas do negócio.

Soluções ERP hoje são capazes de servir grande parte das necessidades genéricas de todas as empresas, desde funções de Recursos Humanos até detalhes de Contabilidade, sendo que algumas empresas inclusive mudam seus processos para se adaptar mais facilmente a um ERP específico.

Essas soluções, porém, são genéricas. Mesmo quando customizáveis, dificilmente atendem necessidades importantes dos processos das cadeias de valor das empresas, pois estes são específicos e poucas empresas desejam remodelá-los de acordo com uma prática comum no mercado, até mesmo porque suas diferenças podem fornecer justamente o diferencial competitivo que possuem. Assim, o software desenvolvido sob encomenda, interna ou externamente, ainda possui, e certamente sempre possuirá, um presença marcante em todos as organizações.

Todos esses sistemas geram dados. Quase que escondidos nesses dados estão as informações necessárias para a tomada de decisão.

### 9.3 A necessidade centralização dos dados

O outro processo de grande sucesso foi a centralização dos dados dos sistemas operacionais e transacionais (OLTP) em bases de dados integradas na empresa, principalmente em SGDBs Relacionais de grande porte.

## 94 CAPÍTULO 9. A SITUAÇÃO DAS INFORMAÇÕES NAS ORGANIZAÇÕES

O quadro do primeiro quarto do século XXI é que a maioria das organizações entende que seus dados operacionais devem estar sob controle centralizado. O ideal é que todos esses sistemas compartilhem uma mesma base, porém necessidades como desempenho, custo de implementação, estratégia empresarial, permanência da tecnologia usada no mercado, e ainda outras, podem levar a existência de mais de uma base.

Esta situação também não é sem seus problemas. A centralização dos dados sob um controle único pode tanto auxiliar como trazer dificuldades para o desenvolvimento rápido de sistemas, principalmente quando há exigência de muita burocracia para alterar a base ou estruturar informações de forma a atender novas demandas de negócio.

O uso de softwares COTS ou ERP integrados, chave do sucesso para parte da operação da empresa, pode criar uma forte dependência da organização com um vendedor, de forma que seja difícil fazer decisões estratégicas de mudança de produto sem um grande custo para a organização, ou simplesmente que seja difícil extrair os dados desses sistemas para atender outras demandas.

Em todo caso, antes da proposta dos Data Warehouse a situação das informações dentro das organizações podia ser descrita, de forma geral, da seguinte maneira:

- Reconhecimento da necessidade de centralização dos dados operacionais, nas aplicações do tipo OLTP, normalmente na forma do uso de uma base de dados unificada, ou de várias bases controlados por um serviço central.
- Existência de vários sistemas de informação, centralizados ou não. Os sistemas não centralizados existem tanto por motivos históricos, no caso de software legado, quanto de aplicações que foram desenvolvidas fora do sistema padrão por diversos motivos: desenvolvimento distribuído nas partes da organização, necessidade de passar por cima de processos burocráticos para atender demandas imediatas de negócio, necessidade de uso de tecnologias específicas, etc.
- Existência de uma grande quantidade de sistemas criados pelos próprios usuários acima do nível operacional, muitas vezes na forma de planilhas eletrônicas, documentos ou mesmo bancos de dados simples, como o Microsoft Access.
- Pouca integração dos dados e dos sistemas destinado ao apoio da decisão, geralmente criados de forma *ad-hoc* e possivelmente a partir de retratos diferentes da informação disponível na organização.

Esse últimos dois fatores levaram a um quadro caótico nos dados destinados a tomada de decisão dentro da organização. Enquanto os dados operacionais eram mantidos dentro de algum controle, especialmente porque são essenciais para o funcionamento da organização, os dados usados para a tomada de decisão, geralmente sumários ou recortes dos dados transacionais, eram baseados em fotografias tiradas de forma diferente e em momentos diferentes da empresa. Esse é o principal diagnóstico que leva a necessidade da existência de Data Warehouses por Inmon (2005): a falta de organização e o excesso de redundância e incertezas nesses dados.

Um exemplo do que poderia acontecer dentro da organização por causa desta situação caótica são dois gerentes ou diretores chegarem a uma reunião com números diferentes sobre um mesmo objetivo da empresa, devido a formas como esses dados foram obtidos. Um diretor, por exemplo, com os dados de venda por semestre, até o último semestre completo, poderia dizer que as vendas estavam aumentando, enquanto um segundo diretor, com os dados de vendas por mês poderia dizer que estavam diminuindo. Dependendo da fonte da informação, a confusão podia ser maior. Um diretor, por exemplo, poderia estar contabilizando os pedidos, enquanto outro os pagamentos feitos. Ainda, um diretor poderia ter usado dados obtidos com algum viés de seleção, como desconsiderar um canal, ou considerar apenas as vendas para os grandes clientes.

A verdade é que se os dados são muitas vezes extraídos dos sistemas OLTP de forma arbitrária e depois trabalhados manualmente pelos membros da organização. A tendência é que se espalhem, sejam alterados em formato e conteúdo e criem uma miríade de fotografias que não só mostram realidades diferentes da empresa, mas que também foram tiradas de forma diferente, com motivações diferentes e que não podem ser mapeadas ao processo original de extração.

Para dar um ideia do problema do espalhamento de dados em um organização, que ocorre tipicamente por planilhas eletrônicas, uma revisão de diferentes pesquisas sobre erros em planilhas de 1995 a 2008 calculou erros em 88% das planilhas no total, sendo que algumas dessas pesquisas encontrou erros em 100% das planilhas investigadas Panko (1998). Soto (2019) relata erros específicos em planilhas que causaram prejuízos a organizações, como a venda de 10.000 ingressos não existentes para os Jogos Olímpicos de Londres de 2012, obrigando o Comitê a substitui-los por outros ingressos de maior valor.

Todo esse quadro anteriormente exposto mostra que apesar dos enormes esforços e algo custo da gestão das informações nas empresas, mesmo em

2019, na prática essa informação ainda é muito carente de organização e controle.

Mesmo depois de todo o aprendizado que foi obtido, de todas as propostas feitas e adotadas, a verdade é que nas organizações do mundo real acaba levando a soluções não ideais. O resultado é que organização acaba por tomar decisões importantes sobre dados com pouca qualidade.

## 9.4 Momento Histórico da Proposta dos Data Warehouse

Em outra escala, o problema do descontrole dos dados já ocorria na década de 1970, onde começaram a aparecer os primeiros sistemas de informação automatizados.

A partir de experiência em trabalhos semelhantes ao que hoje é chamado de Data Warehouse, em 1992 Inmon (1992) lançou seu primeiro livro sobre o assunto, chamando a atenção de todos para a necessidade de aumentar a qualidade dessa dado, na forma do que ele denominou de Data Warehouse. Nessa época, os microcomputadores, as planilhas e os sistemas criados diretamente pelo usuário já eram parte integradas da Tecnologia da Informação nas organizações. Porém, ainda não tinha sido atingida a fase do **Big Data**, e tantas outras tecnologias comuns hoje em dia não estavam disponíveis.

Mesmo assim, o retrato apresentado por Inmon (1992) e renovado em Inmon (2005) pouco difere do retrato que vemos agora. Se algo aconteceu foi o aparecimento de mais fontes de dados e de demandas cada vez maiores de tratar informação.

## 9.5 O Data Warehouse

Então, da mesma forma que as bases relacionais propiciam a integração dos dados transacionais, Inmon (2005) e outros perceberam que era necessário **centralizar** os dados destinados a tomada de decisão, normalmente por meio de sistemas OLAP. Mais que isso, que esses dados não deveriam estar registrados da mesma forma dos sistemas OLTP, pois eles refletem uma outra visão, que é composta de dados históricos, persistentes organizados pelo assunto, e com outras características que vão definir um **Data Warehouse**.

De certa forma, o Data Warehouse, como conceito, está para os sistemas

OLAP e de tomada de decisão assim como as Bases de Dados Integradas estão para os sistemas OLTP e operacionais. Trazer toda a informação para um ambiente controlado, sanitizado e centralizado é uma virtude a ser perseguida.

Para ser implementado e ser útil, porém, o Data Warehouse precisa ser adaptado ao fato que a base integrada organizacional é apenas uma utopia. Por isso, a implantação de um Data Warehouse também precisa estabelecer uma metodologia de coleta e transformação dos dados. Para isso ele tem que ter acesso as várias fontes de informação primitiva da empresa e a capacidade de transformá-la e armazená-la de forma a atender, de maneira eficiente, as demandas dos sistemas OLAP e de tomada de decisão. Isso vai levar a existência de uma estrutura complexa que inclui mecanismos que usam nomes como *ETL*, *ELT* e *Staging*, tratados neste livro.

O Data Warehouse também tem que ser separado dos sistemas OLTP por questões de desempenho. A forma de acesso as operações diárias gera uma carga no sistema, e exigências de tempo de atendimento, diferente das operações realizadas em um Data Warehouse (Inmon, 2005). A Figura 9.1 mostra a diferença do comportamento da carga desses dois sistemas.

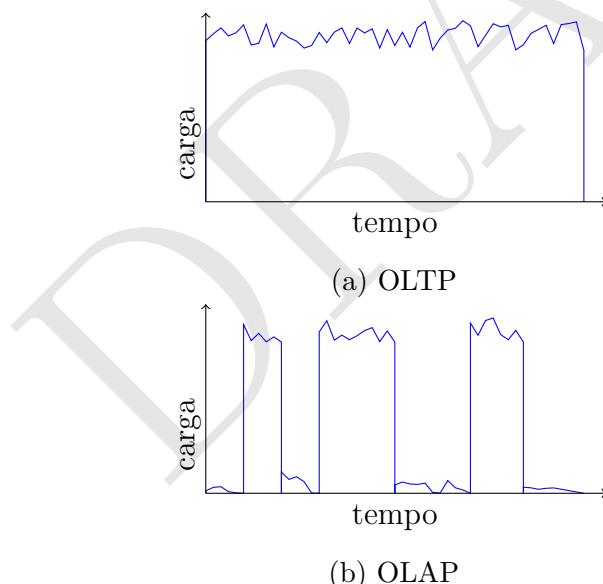


Figura 9.1: Diferentes cargas exigidas de sistemas OLTP e OLAP. Fonte: (Inmon, 2005)

Finalmente, para corretamente distribuir os dados a quem precisa e pode consultá-los, levando em consideração fatores tanto de desempenho quanto como leis como a Sarbanes–Oxley (United States Code, 2002), que fala do

controle do acesso à informação dentro da organização, o Data Warehouse precisa alimentar bases menores conhecidas como **Data Marts**.

## 9.6 Data Warehouse e Bancos de Dados

O mercado atual usa o termo Banco de Dados, majoritariamente querendo dizer que é relacional, para indicar o repositório de dados da organização, em especial o repositório de dados OLTP, visando o rápido acesso a dados únicos, para acelerar essas transações. É possível e desejado fazer relatórios e gráficos a partir dos dados nos bancos de dados, porém eles normalmente refletem o estado atual da organização.

O termo Data Warehouse é reservado para indicar o repositórios de dados históricos da organização, visando o rápido acesso a visões agregadas, como relatórios, e servir operação OLAP.

Outra diferença básica é que o modelo de dados de um Banco de Dados é criado para permitir o uso mais geral possível dos dados, e segue regras, conhecidas como normalização, para evitar redundâncias. Já o Data Warehouse é criado para atender necessidades específicas, com um modelo muitas vezes não normalizado.

Porém, há um erro em tentar criar essa dicotomia. A verdade é que um Data warehouse precisa de um sistema de banco de dados para funcionar, ou seja, os Data Warehouse tem que existir dentro de um sistema de informação com as mesmas características dos sistemas de banco de dados. Devido a existência de SGBD relacionais de alto desempenho, que com o tempo foram adaptados para atender melhor as demandas de desempenho de um Data Warehouse, grande parte dos Data Warehouse funciona dentro de sistemas relacionais.

Existem porém outras bases não relacionais muito utilizadas para Data Warehouse. Normalmente elas se caracterizam por serem otimizadas para certo tipo de consultas, normalmente operações sumários em uma tabela e operações OLAP, e no uso do Modelo Dimensional, que será tratado mais tarde neste texto.

## 9.7 Data Warehouses ainda são necessárias?

Tendo em vista que Data Warehouses foram projetadas no final do século XX, é possível questionar sua aplicabilidade quase 30 anos depois de seu aparecimento.

Uma questão que se põe é causada pelo avanço da tecnologia. A disponibilidade de armazenar um volume muito maior em discos e o uso de múltiplas CPUs nos servidores permite que sejam construídos sistemas de informação OLTP que pouco se preocupam com a necessidade de limpar periodicamente os dados antigos do sistema. Esses dados podem então ser acessados diretamente?

A resposta para essa questão está no fato que sistemas OLTP e OLAP, e também as novas aplicações de **Business Intelligence (BI)**, trazem requisitos diferentes, que podem ser atendidos exatamente pelos Data Warehouses.

Por outro lado, sistemas de *Big Data* e *Data Lake* podem parecer concorrentes ao sistemas de Data Warehouse, quando de fato são complementares e acabam criando uma sinergia.

## 9.8 Mais de um Data Warehouse?

Apesar da ideia básica do Data Warehouse é ser uma base central de informações, como nos casos dos sistemas de informação tradicionais, fatores do dia a dia fizeram que o sonho do Data Warehouse centralizado fosse substituído, no mercado, pela existência, dentro da organização, de mais de um Data Warehouse. Uma pesquisa descrita por Wells e Nahari (2019) mostrou que a maioria das empresas usa mais de Data Warehouses, de acordo com os dados apresentados na Figura 9.2.

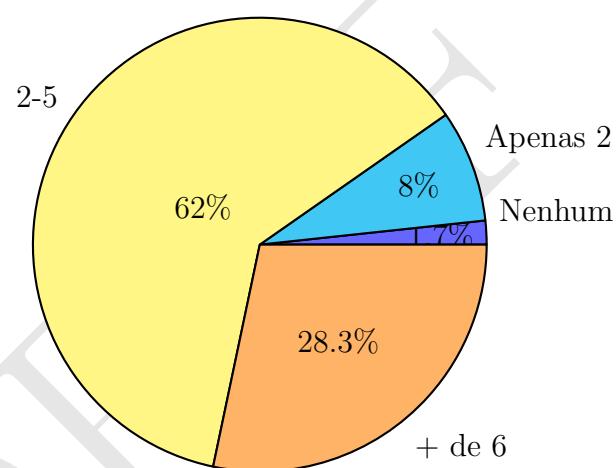


Figura 9.2: Número de data warehouses por empresa(Wells e Nahari, 2019).

# Capítulo 10

## O Que é um Data Warehouse

### 10.1 O Debate Kimball x Inmon

Dois autores dominam as publicações sobre Data Warehouse no mundo.

O primeiro deles é **Bill Inmon**, considerado o pai do Data Warehouse, que começou a discutir o assunto e cunhou o termo ainda em 1970. Seu livro *Building the Data Warehouse*, em sua primeira edição de 1992, foi um marco importante na divulgação do tema(Kempe e Williams, 2012).

O segundo autor é **Ralph Kimball**. que com a primeira edição de *The Data Warehouse Toolkit*, de 1996, trouxe um conjunto de práticas e exemplos que facilitaram a implementação de data warehouses para muitos(Kempe e Williams, 2012).

Inmon e Kimball mostram caminhos diferentes de implementar data warehouses, sendo que Inmon favorece a abordagem *top-down*, baseada em um modelo normalizado, e Kimball a abordagem *bottom-up*, baseada em um modelo dimensional, de criação de data warehouses. A polêmica entre suas ideias é uma das mais conhecidas da Informática.

Em especial, Ralph Kimball (2013) apresentou ao público e defende o uso do do Modelo Dimensional<sup>1</sup>. Ele diz: “*Data in the queryable presentation area of the data warehouse must be dimensional*”(Ralph Kimball, 2013). Já Inmon (2005) diz “*Star schemas are not very good in the long run for a data warehouse.*” Esquemas Estrela são a forma de implementar modelos

---

<sup>1</sup>O Modelo Dimensional e os termos Fato e Dimensão foram inventados nos anos 1960 por um projeto conjunto da *Dartmouth University* e da *General Mills*(Ralph Kimball, 2013)

dimensionais em Bancos de Dados Relacionais.

## 10.2 Definição de Inmon para Data Warehouse

Segundo Inmon (2005), um **Data Warehouse** é “Uma coleção de dados orientada a assunto, integrada, não volátil e variante no tempo que apoia as decisões de gerência.”

### 10.2.1 Orientação a Assunto

A ideia de Inmon é que enquanto os sistemas tradicionais são organizados ao redor das funções da empresa, ou seja, do que a empresa faz, os data warehouses são organizados em torno dos **assuntos** que a empresa trata, ou seja, dos tópicos sobre as quais ela tem que tomar decisão.

Por exemplo, o que faz um mercado de varejo? Em sua cadeia de valor, Figura 10.1, ele compra no atacado, estoca e vende produtos no varejo. Os seus sistemas de informação são orientados para essas atividades. Uma lista de sistemas de informação que podem estar disponíveis em um pequeno mercado é mostrada na Tabela 10.1.

Tabela 10.1: Sistemas em um pequeno mercado.

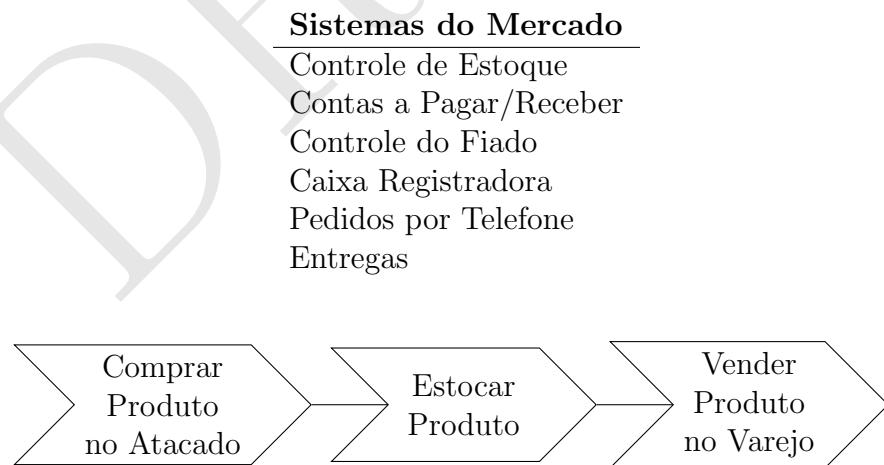


Figura 10.1: A cadeia de valor de um mercado, descrição em ARIS.

Porém, como o mercado analisa suas informações por assuntos? Por exemplo, com perguntas como “vendas por produto” ou “vendas por loja”. O

assunto, nesse caso, é **vendas**, e a informação principal é um agregado de várias instâncias do banco de dados que registram um tipo de venda.

Nas fontes originais de dados dentro da empresa, a informação referente a um assunto pode estar dividida de várias formas. Por exemplo, para o assunto cliente podemos ter seu nome e CPF em um registro de uma venda, porém seu endereço só estará em um registro de uma entrega. Já seu telefone pode estar em um registro de um pedido feito pelo telefone para entrega. Todas essas ações são atendidas por um ou mais sistemas de informação da empresa, e os dados podem estar em um ou mais repositórios, e mesmo dentro de tabelas diferentes em uma base de dados única, dependendo da maturidade da empresa no tratamento de seus dados.

Além disso, tanto os sistemas são trocados e evoluem com o tempo, quanto os dados e seus formatos.

Por exemplo, para o assunto cliente, podemos ter dados dos clientes atuais em um SGDB, dados de clientes antigos guardados em fitas backup de sistemas legados, dados de clientes de um setor específico guardados em uma planilha, etc.

Podemos também ter dados repetidos de um mesmo cliente, e com algumas alterações. Um cliente que mudou de endereço, ou mesmo uma cliente que trocou de nome ao casar (Inmon, 2005). A Figura 10.2 mostra um exemplo de fontes possíveis sobre clientes.

Cada assunto define um conjunto de tabelas correlacionadas (Inmon, 2005). No modelo dimensional (Ralph Kimball, 2013), cada assunto vai definir um (ou mais) modelos compostos de uma tabela central, que indica o tópico consultado, a **tabela fato**, e tabelas auxiliares que mostram como a tabela fato pode ser consultada, as **tabelas dimensão**.

É importante lembrar que quando trazidas para o data warehouse, todas essas informações sobre o cliente tem que ser processadas de forma que um mesmo cliente seja sempre identificado por uma mesma chave.

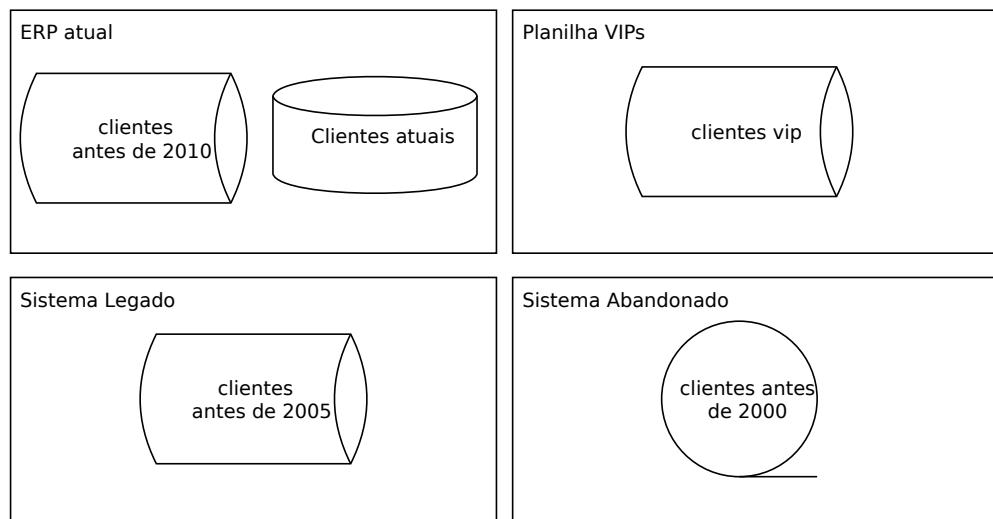


Figura 10.2: Diversas fontes de onde pode ser obtido o assunto cliente

### 10.3 Data Warehouse 2.0

### 10.4 Data Lake

# Capítulo 11

## O Método Beam\*

**Beam\***, **Business Event Analysis and Modelling** é um método ágil de projeto de Data Warehouse proposto por Corr e Stagnitto (2012) que visa geral o modelo estrela de forma colaborativa. Ele se baseia no conceito de 5W2H, que os autores chamam de **7W**. Esse capítulo é um resumo do material apresentado no livro<sup>1</sup>.

O método propõe um ciclo de vida ágil, baseado em iterações sucessivas em busca de obter valor para as partes interessadas. Cada passo do ciclo tem quatro passos principais: Levantar Requisitos, Construir o ETL, Construir o Protótipo de Business Intelligence e Revisão. Dentro desse ciclo de vida, BEAM\* se preocupa principalmente em definir um modelo dimensional, na forma de um Esquema Estrela, de maneira colaborativa, em sessões de trabalho conjuntas.

Para construir o modelo dimensional, o método busca responder as perguntas 5W2H, *Who*, *What*, *Where*, *When*, *How Many*, *Why and How*, a partir de exemplos colocados em uma tabela em um quadro branco. Na prática funciona como uma modelagem de dados por exemplo. Corr e Stagnitto (2012) chamam isso de *Framework 7W*, e o ilustram com a Figura

A **modelagem ágil de dados** tem como característica ser Colaborativa, combinando análise e projeto em um mesmo processo e envolvendo as partes interessadas, de forma evolucionária, incremental e iterativa.

Uma visão rápida do método de modelagem colaborativa pode ser resumida em quatro passos:

---

<sup>1</sup>Todo esse capítulo é um resumo baseado no livro, com alguns comentários deste autor, onde foi feito o maior esforço possível para deixar indicadas as citações literais.

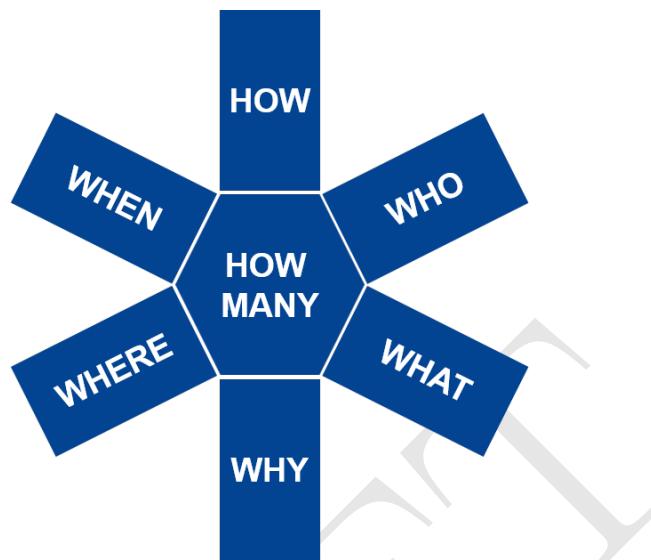


Figura 11.1: Descrição gráfica do *Framework 7W*. Fonte:(Corr e Stagnitto, 2012)

1. Junte seus usuários de uma área de negócios
2. Pergunte o que acontece nessa área, isto é, eventos na forma “quem faz o que?”, levantando uma lista priorizada desses eventos;
3. Peça exemplos de dados reais de “quem” e o “que”, do evento prioritário;
4. Preencha o resto dos dados: quando, onde, quantos, por que e como.
5. Repita para cada “quem faz o que?”, por ordem de prioridade.

Isso permite criar, para cada um desses eventos, um modelo estrela que segue o **Framework 7W**, formado por uma Tabela Fato que descreve ”quantos?” cercada de Tabelas Dimensão que definem como, quem, o que, por que, onde e quando.

O método usa 5 ferramentas: a Tabela Beam\*, a Carta de Hierarquia, a Linha de Tempo, a Matriz de Eventos e o Modelo Estrela Aumentado.

Cada **Tabela Beam\*** modela um evento de negócio, usando exemplos de dados reais para documentar os detalhes para o framework 7W. Ela descreve fatos e dimensões e explica os padrões do projeto dimensional.

A **Carta de Hierarquia** é um diagrama simples que mostra as relações hierárquicas entre os atributos das dimensões. Ela ajuda a levantar quais são esses atributos e também a entender que operações OLAP de *drill down* e *roll up* que serão necessárias.

As **Linhas de Tempo** exploram as relações temporais entre os eventos de negócio, permitindo descobrir detalhes da pergunta “Quando?” Elas também indicam como funcionam os processos e como são sequenciados, além da duração dos fatos.

A **Matriz de Eventos** mostra os relacionamentos entre todos os eventos e as dimensões do modelo.

O **Modelo Estrela Aumentado** provê a visualização do modelo dimensional e permite gerar esquemas físicos. O modelo tradicional é estendido com **short codes BEAM\***, que são siglas de duas letras que indicam propriedades dos dados, fornecendo informação sem ocupar espaços.

A estratégia de trabalho é chamada *Modelstorming* e se baseia na tríade abrir, explorar e fechar, comum em métodos criativos. Esse modelo pode ser descrito na figura 11.2. Nessa figura são descritas as atividades e as ferramentas nelas usadas.

O primeiro passo é descobrir “Quem faz o que”, esse passo fornece as informações iniciais que criam a **Tabela Beam\***. Os dados de cada evento são detalhados. O próximo passo é modelar as dimensões, ainda com a Tabela Beam\* e criando a Carta de Hierarquia. A seguir são modeladas as sequências de evento, o que produz a Matriz de Eventos e a Linha do Tempo. Finalmente se decide que eventos implementar.



Figura 11.2: Descrição gráfica do Modelstorming. Fonte:(Corr e Stagnitto, 2012)

Esse diagrama pode ser definido como um conjunto de processos de negócio escritos em ARIS-EPC, como na Figura 11.3. Isso define basicamente um processo de seis passos:

1. perguntar aos participantes quem faz o que nos negócios, o que gera uma lista de histórias, cada história descrevendo um evento.
2. priorizar as histórias
3. documentar cada evento em uma tabela BEAM\* seguindo uma **ordem de descoberta**;

4. descrever detalhadamente as dimensões relativas a cada evento;
5. criar o Modelo de Sequência de Eventos, e
6. escolher os modelos a implementar.

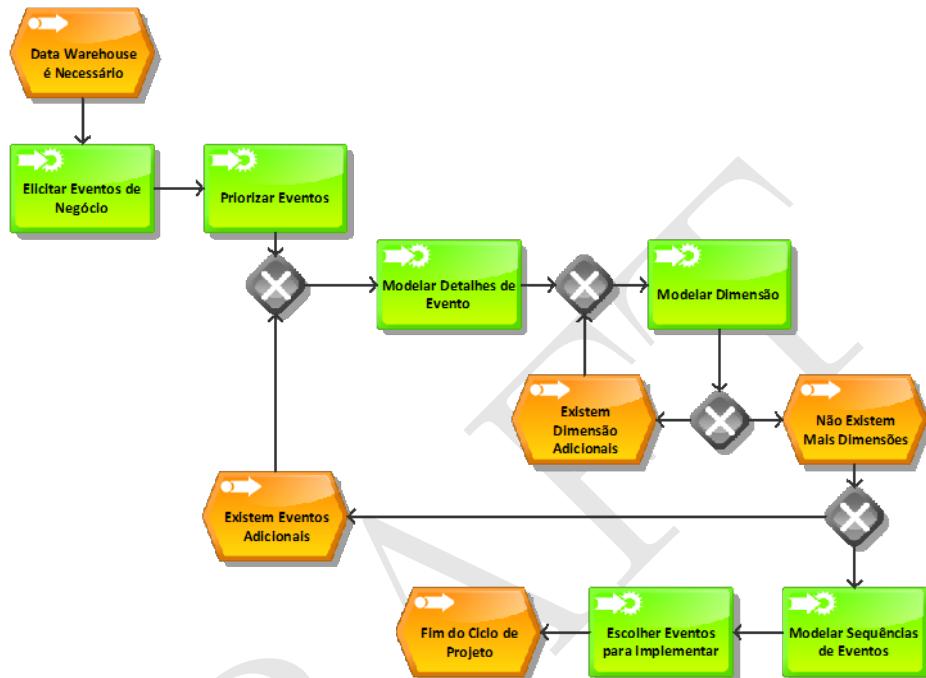


Figura 11.3: Descrição do Modelstorming em ARIS-EPC

## 11.1 Eventos de Negócio OU Histórias de Dados

Segundo Corr e Stagnitto (2012), um **Evento de Negócio** é uma ação individual, executada por pessoas ou organizações durante a execução de seus processos de negócio. Quando a empresa funciona, deixa uma trilha de eventos de negócios dentro dos bancos de dados operacionais, que contém detalhes mensuráveis no nível atômico. Exemplos de eventos de negócios são: uma venda, uma entrega, uma montagem de equipamento, o atendimento de uma reclamação, etc.

Para identificar um evento de negócio, no Beam\* as partes interessadas devem contar suas *data stories*, que vão produzir requisitos de dados concisos.

As *data stories*, ou **histórias de dados**, são semelhantes conceitualmente

às histórias de usuário. Elas são escritas ou contadas pelas partes interessadas envolvidas com o negócio.

Essas histórias devem se concentrar em requisitos de dados, e são escritas como um quadro, tabela ou planilha. Assim, a narrativa de um evento de negócio é usada para descobrir requisitos de dados para o BI.

Basicamente, Histórias de Dados e Eventos de Negócio podem ser usados de formas intercambiáveis, mas a ideia é que Histórias de dados descrevem Eventos de Negócios.

Eventos de negócio são sempre atividades, sendo representados por verbos. Existem três tipos de histórias de dados, ou eventos de negócio: discretas, evolutivas ou recorrentes.

Um **evento discreto** acontece em um instante ou um curto espaço de tempo que pode ser considerado pontual. Eventos desse tipo representam ações atômicas, por exemplo, a compra de um produto por um cliente em uma loja de varejo. Esses eventos são considerados completos no momento em que ocorrem, sendo considerados a partir daí acabados e imutáveis. Isso significa que quando chegam ao BI, provenientes dos sistemas operacionais da empresa, não serão mais alterados. Normalmente esses eventos usam um único verbo e um único *timestamp*. Além disso, seus detalhes não mudam no tempo.

Um **evento evolutivo** representa algo que acontece em períodos irregulares de tempo, podendo migrar dos sistemas operacionais para o BI antes de estarem terminados. Por exemplo, uma venda feita pela internet e que já foi paga mas não foi entregue. Eventos desse tipo possuem vários verbos em sua descrição, como “pagar” e “entregar”. São, normalmente, uma série de eventos discretos, como passos ou marcos de um processo longo.

Um **evento recorrente** acontece em intervalos previsíveis e regulares. Por exemplo, um inventário noturno de um produto ou loja. Eventos desse tipo podem ser amostras ou sumários de eventos discretos, podendo ser cumulativos. São detalhes atômicos de medidas automáticas do sistema, como o saldo no fim do dia de uma conta bancária.

Cada um desses eventos exige um tipo de Tabela Fato, entre as propostas por Kimball et al. (2008), para representá-lo, como na Tabela 11.1.

Tabela 11.1: Equivalência dos tipos de eventos do método BEAM\* e os Tipos de Tabela Fato de Kimball. Fonte: Corr e Stagnitto, 2012

Tipo de Evento	Tipo de Esquema Estrela
Discreto	Transação
Evolutivo	Fotografia Acumulativa
Recorrente	Fotografia Periódica

## 11.2 Descobrindo as Histórias

O primeiro passo da criação da Modelagem de Eventos de Negócio é descobrir quem faz o que, e está indicado na Figura 11.4. No Processo Modelstorming, da Figura 11.3, adaptada de Corr e Stagnitto (2012). Esse a passo corresponde as atividades Elicitar Eventos de Negócio e Priorizar Eventos.

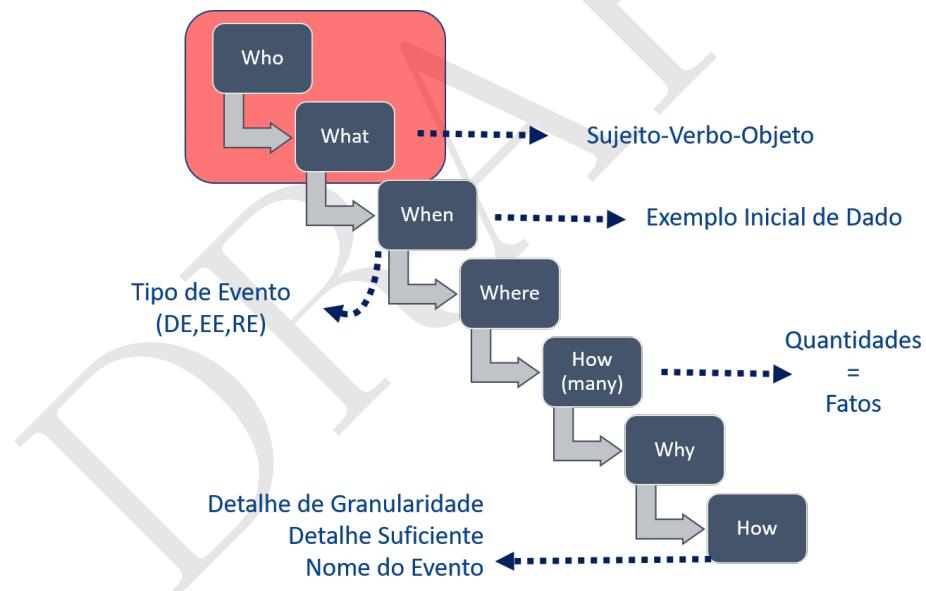


Figura 11.4: Primeiro passo da ordem de descoberta. Fonte:(Corr e Stagnitto, 2012)

Nesse passo, os participantes da reunião devem gerar sentenças na forma sujeito-verbo-objeto que indiquem quem faz o que no processo. Cada resposta dada nesse formato define um evento de negócio.

Um foco a ser dado nessa descrição, ou na seleção dos eventos que serão usados no próximo passo, é a necessidade desse evento ser interessante para

a próxima *release* do projeto. Assim, essas sentenças podem ser levadas para um *backlog*, caso não sejam usadas nessa iteração.

É importante que, levantados os eventos, eles sejam tratados um de cada vez, e que sejam, pelo menos nas primeiras iterações, evento atômicos.

A descrição, feita pelos participantes, não deve se preocupar com o que eles propriamente ditos fazem, mas sim com que pessoa ou organização realmente executa a atividade que o verbo descreve.

Um exemplo simples é dado por Corr e Stagnitto (2012): o grupo pode estar convicto que é importante caracterizar o faturamento gerado por um produto, mas quais eventos ele pode procurar para isso? A resposta indica dois eventos possíveis: cliente compra produto ou vendedor vende produto. É possível ver a estrutura sujeito-verbo-objeto e a diferença de perspectiva do que é mais importante para a organização dada a escolha de uma das sentenças. É válido observar que outros sujeitos poderão ser inseridos depois, isto é, mesmo que a organização prefira privilegiar a narrativa que cliente compra sobre a que vendedor vende no início desse passo a passo, nos passos posteriores o outro agente poderá ser incluído, ou seja, a informação não será perdida.

A escolha do verbo é muito importante nesse passo, pois ele sempre fará parte das perguntas do *framework 7W*. A primeira pergunta é basicamente “Quem faz o que?”, e as próximas perguntas serão “Quando faz?”, “Como faz?”, etc.

Uma forma de descobrir os eventos é partir dos atores envolvidos na organização. Escolher um ator, como motorista, ou cliente, permite investigar o que ele faz. Atores podem fazer mais de uma coisa, por exemplo, um cliente pode: comprar, pedir cotação, devolver, etc.

### 11.2.1 Prática para Levantamento e Seleção de Eventos

O levantamento de eventos pode ser feito de algumas maneiras. Um *brainstorm*, por exemplo, pode ser uma prática adequada para levantar os eventos.

Outra prática comum é o uso de *post-its* para escrever os eventos e colocá-los em um espaço ou quadro. Nesse caso, todos devem ter acesso ao quadro e ficar em pé para trabalhar no mesmo.

O quadro pode ser organizado ou não. Pode se organizar o quadro como uma linha de tempo, ou por ator, ou como o grupo achar mais eficiente.

Também é possível não organizar o quadro inicialmente e depois agrupar os eventos obtidos em alguma organização.

Antes da seleção, porém, é importante localizar os eventos repetidos ou que possam significar a mesma coisa. Corrigidas as redundâncias, e até mesmo algum erro eventual, pode se passar para a fase de seleção.

Uma método comum de seleção é chamado de *dot-voting*, ou votar por pontos. Nesse caso, cada participante recebe um número de pontos que deve gastar marcando os eventos mais importantes ou mais prioritários para essa release. É comum que isso seja feito com 3 votos, na forma de marcar com uma caneta ou uma pequena etiqueta adesiva, ou mesmo um pedaço de *post-it*. Os eventos então são listados em ordem de votação para serem analisados detalhadamente.

### 11.3 Passo a Passo da Modelagem de Eventos de Negócio

Cada história de dados será contada por meio de uma Tabela BEAM\*. Nesta seção o método de construção desta tabela é mostrado passo a passo.

Esse passos correspondem a atividade Modelar Detalhes de Evento no Processo Modelstorming descrito pela Figura 11.3.

Cada história é contada na Tabela BEAM\* por meio da identificação de **detalhes** que a explicam e do uso de exemplos que mostram como podem acontecer.

Esses detalhes mostram as várias informações que podem ser usadas na história para entendê-la melhor e trazer mais entendimento do negócio ou da organização alvo do sistema de BI ou Data Warehouse sendo criado.

Os detalhes correspondem a respostas a perguntas do 5W2H, ou 7W da notação de Corr e Stagnitto (2012). O termo **detalhe** é usado para mostrar que ainda não são as dimensões finais do Modelo Dimensional.

A atividade Modelar Detalhes de Eventos é na verdade um sub-processo que é descrito em ARIS-EPC na Figura e que gera a Tabela BEAM\*.

Dentro desse processo, a primeira atividade a ser realizada, Definir Quem-O Que-Quando, também é um subprocesso, que é descrito em mais detalhe na Figura 11.6.

Nas próximas subseções serão descritos os passos de ambos os processos.

### 11.3. PASSO A PASSO DA MODELAGEM DE EVENTOS DE NEGÓCIO113

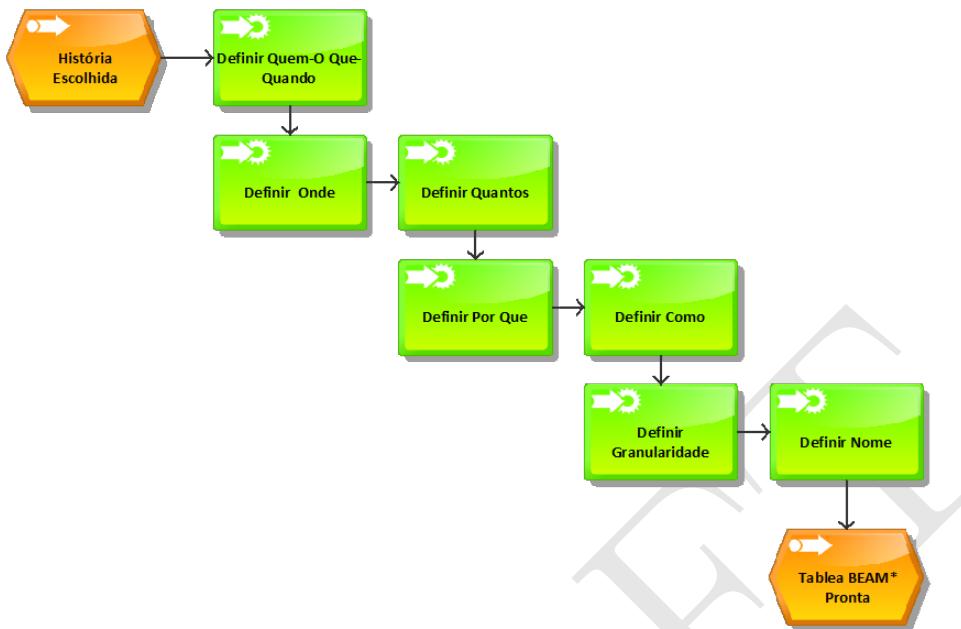


Figura 11.5: O processo Modelar Detalhes de Evento, descrito em ARIS-EPC

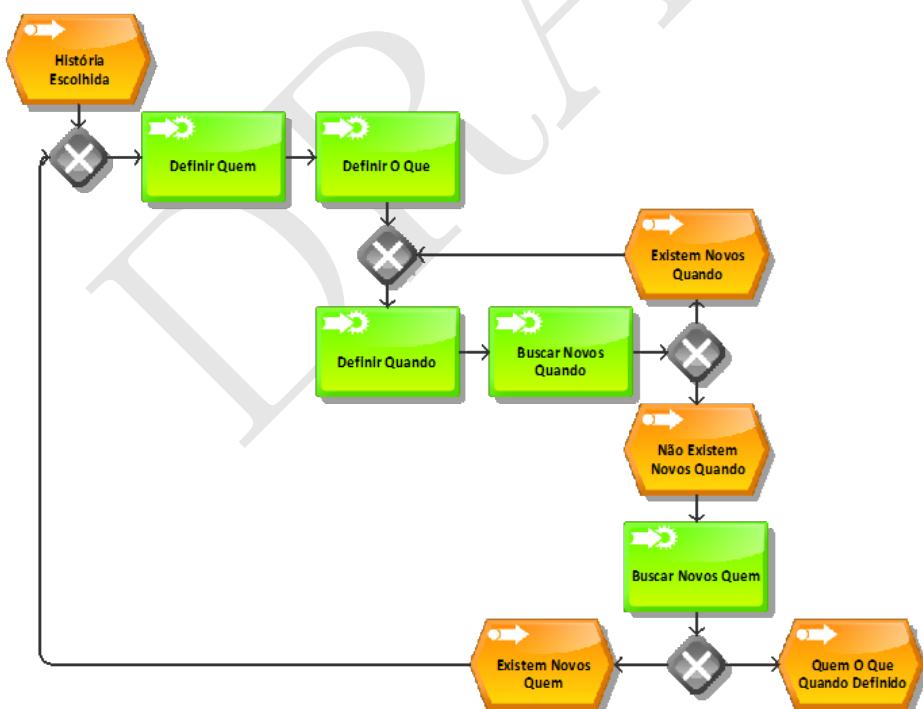


Figura 11.6: Definir Quem-O Que-Quando, descrito em ARIS-EPC

### 11.3.1 A sessão de criação da Tabela BEAM\*

A sessão de criação da Tabela BEAM\* deve seguir o formato ativo e colaborativo que todo o método propõe.

A equipe deve ser a mesma que levantou as histórias, na verdade, o ideal é que essa sessão seja contínua com a anterior, pelo menos para a primeira história contada, para que os participantes possam entender melhor o resultado prático de suas ações.

A dinâmica exige o uso de um quadro branco, método preferido deste autor, ou ainda de grandes folhas brancas, que tem o defeito de não permitir apagar as informações. Obviamente, são necessárias canetas, possivelmente de mais de uma cor.

A ideia é que todos os participantes tenham acesso ao quadro branco, para incentivar a participação. É necessário que alguém com conhecimento do método participe, preferencialmente na forma de mediador e orientador.

Esta forma de trabalhar, participativa, ativa e colaborativa é comum a vários métodos modernos e tem bastante sucesso.

A tecnologia atual permite que sejam tiradas fotografias do quadro, em vários momentos, que podem ser compartilhadas entre os participantes e guardadas como referência.

Durante ou após a sessão é interessante transcrever os resultados para planilhas eletrônicas.

### 11.3.2 Tabela BEAM\* Inicial

Tudo se inicia com a criação da primeira Tabela Beam\* em seu formato inicial. Essa tabela, feita no quadro branco, começa com o formato indicado na Figura 11.7, que indica o evento por meio de seu sujeito, verbo e objeto, como descritos no levantamento de histórias. Deve ser escolhida a história mais prioritária. No caso exemplo, a história escolhida é “Cliente encomenda produto”.

No quadro branco, deve se deixar bastante espaço em baixo e à direita, de forma que possam ser desenhadas novas colunas e usado o espaço da tabela para fornecer os exemplos que ajudarão a entender o evento.

A Figura 11.8 mostra a ordem de descoberta das informações necessárias para construir a Tabela BEAM\* para um evento. Estas informações são os **detalhes** da história e devem ser levantados de forma específica.

<u>CLIENTE</u>	<u>encomenda</u>
<u>PRODUTO</u>	
<u>&lt;who&gt;</u>	<u>&lt;what&gt;</u>

Figura 11.7: Configuração inicial da tabela BEAM\* para o evento Cliente Encomenda Produto. É necessário deixar bastante espaço para continuar a atividade. Adaptado de:(Corr e Stagnitto, 2012)

As subseções a seguir descrevem esse passo a passo.

### 11.3.3 Modelando o quando

Todo evento, e toda história de dados, acontece em algum momento ou intervalo. Por isso tabelas dimensão falando sobre datas e horas são tão pervasivas nos modelos dimensionais.

O tempo em que acontece um evento é normalmente descrito por uma preposição e um substantivo, como em “na (em+a) data combinada”. As preposições que transmitem relações de tempo, muitas vezes em contrução com um artigo, são: a, até, após, de, desde, em, entre, durante e todo.

Os detalhes definidos por “quando” são capazes, muitas vezes, de dar dicas adicionais sobre os detalhes, e por isso é a primeira após aquelas que definem o evento.

A Tabela BEAM\* com o primeiro detalhe de tempo fica como na Figura 11.9. A história passa a ser “Cliente encomenda produto na data do pedido.”

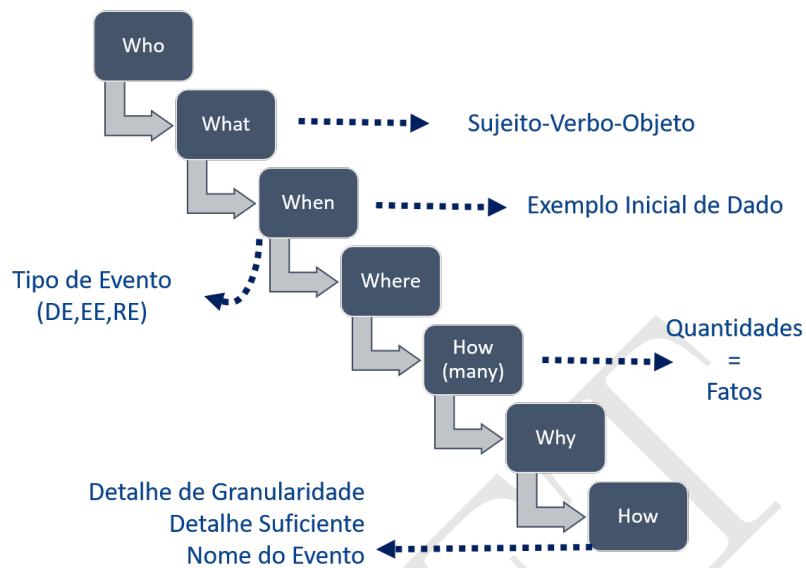


Figura 11.8: Ordem de descoberta das informações do modelo. Fonte:(Corr e Stagnitto, 2012)

### 11.3.4 Primeiros Exemplos

Alcançado o ponto de preenchimento da Tabela Beam\* em que se obtém apenas os cabeçalhos das três primeiras colunas, é o momento de começar a preenchê-la com exemplos de dados.

Para isso devem ser encontrados exemplos úteis para entender essa história, a partir de **temas**. Os temas sugeridos são:

**Típico** exemplos típicos dos dados que representam o evento, como os dados normais de uma venda;

**Diferente** mais exemplos semelhantes aos típicos, porém diferentes do primeiro. Também eventos buscando limites dos dados, como valores mínimos e máximos;

**Repetido** exemplos tão semelhantes quanto possível aos eventos típicos;

**Ausências** exemplos onde faltam algumas informações;

**Grupos** exemplos onde aparecem grupos, como grupos de clientes ou de produto. e

**Limites** exemplos que mostram limites.

No exemplo, que agora evolui com seu caso típico, a história passa a ser “Cliente Antônio Alves encomenda produto suéter azul na data de pedido 10 de maio de 2011.” Isso permite chegar a tabela BEAM\* da Figura 11.10.

### 11.3. PASSO A PASSO DA MODELAGEM DE EVENTOS DE NEGÓCIO117

<b>CLIENTE</b>	<b>encomenda</b>	<i>na</i> <b>DATA</b>
<b>PRODUTO</b>		<b>DE PEDIDO</b>
<b>&lt;who&gt;</b>	<b>&lt;what&gt;</b>	<b>&lt;quando&gt;</b>

Figura 11.9: Configuração da tabela BEAM\* com o o detalhe “quando”. Aparece a preposição utilizada (em+a) e um sustantivo. Adaptado de:(Corr e Stagnitto, 2012)

O ideal é que sejam exemplos reais conhecidos pelos participantes ou tirados diretamente das bases de dados da empresa.

Em busca de histórias diferentes, encontra-se primeiro uma variação simples, como mostra a Figura 11.11.

Seguindo, os participantes podem encontrar um exemplo com repetições, como na Figura 11.12. Nesse caso temos uma compra igual, onde a mesma pessoa comprou o mesmo item na mesma data, ou seja, ela comprou o mesmo item duas vezes e foi registrado assim.

Também é possível que para algumas histórias que ocorreram, algum dado esteja faltando. Não importa o motivo, o que queremos registrar é que, nos sistemas da organização é possível encontrar o registro de um evento assim. Na Figura 11.13 temos um caso onde não foi registrado o nome do cliente, mas onde também a data está registrada de forma incerta, ou é conhecida por uma regra, pois veio de um sistema que era usado “dez anos atrás.”

O processo segue dessa forma, com os colaboradores preenchendo mais e mais exemplos, até que achem que todas as possibilidades foram esgotadas. Na Figura 11.14 foram adicionados dois casos onde o cliente é um **grupo**: Colégio LMS e Empresa EPM. Um grupo é apenas algo coletivo em relação ao normal esperado no campo. Também aparecem registros que contém mais

<u>CLIENTE</u>	<u>encomenda PRODUTO</u>	<u>DATA DE PEDIIDO</u>	
<u>&lt;who&gt;</u>	<u>&lt;what&gt; MD</u>	<u>&lt;quando&gt; MD</u>	
Antônio Alves	Suéter Azul	10-maio-2011	

Figura 11.10: Configuração da tabela BEAM\* com o primeiro exemplo, mostrando uma história típica. Adaptado de:(Corr e Stagnitto, 2012)

de um item, como “Calça+Camisa”

Para trabalhar com a dimensão “quando” é interessante pensar em referências relativas, como ontem, hoje e 5 anos atrás. Isso facilitará aos participantes entender melhor a latência do dado, ou por quanto tempo ele é interessante.

### Sujeitos desconhecidos

Em muitos casos o sujeito existe mas nunca é registrado. Isso é o caso, por exemplo, de uma rede de vendas a varejo que nunca guarda o nome do cliente.

Nesse caso o que acontece é que o sujeito existe no modelo lógico, mas não existirá no modelo físico implementado. Durante a reunião pode ser usado um nome artificial, como “Zé das Couves” e “Maria Chiquinha”, ou pode ser usado apenas um traço ou a palavra anônimo.

#### 11.3.5 Refazendo o ciclo “quem-o que -como”

Após terminar o primeiro passo preenchendo os exemplos com as colunas “quem”, “o que” e “quando”, é **necessário** exaurir as possibilidades de existirem outras colunas semelhantes.

### 11.3. PASSO A PASSO DA MODELAGEM DE EVENTOS DE NEGÓCIO119

CLIENTE	encomenda	na DATA	DE PESSOAS
	PRODUTO	DE PEDIDO	
<who>	<what> MD	<quando> MD	
Antônio Alves Bia Betrand	Suéter Azul Calça Vison	10-maio-2011 29-jun-2011	

Figura 11.11: Configuração da tabela BEAM\* com o segundo exemplo, mostrando uma outra história típica, diferente da primeira. Adaptado de:(Corr e Stagnitto, 2012)

É possível encontrar mais de uma coluna “quando”, principalmente quando em vez de um instante está sendo tratado um intervalo de tempo. Um exemplo típico é um processo que começa com um pedido e termina com uma entrega. A Figura 11.15 mostra como isso pode ser feito, e não há nenhuma complicaçāo.

É possível encontrar mais de uma coluna “quem” quando existem outros participantes na história. Por exemplo, um cliente pode pedir um produto na data do pedido para um vendedor. Assim, podem aparecer novas colunas do tipo “quem”. A Figura 11.16 mostra um exemplo com duas colunas do tipo “quem”: cliente e “com vendedor”.

Personagens adicionais dessa história possuem uma preposição que mostra uma relação com a história, enquanto o personagem principal se relaciona por meio do verbo.

#### Detalhes dos detalhes

Não devem ser confundidos detalhes adicionais da história com detalhes dos detalhes da história.

O participantes podem, nesse ponto, querer registrar na tabela detalhes

<u>CLIENTE</u>	<u>encomenda</u>	<u>DATA</u>	
<u>PRODUTO</u>		<u>DE PEDIDO</u>	
<u>&lt;who&gt;</u>	<u>&lt;what&gt; MD</u>	<u>&lt;quando&gt; MD</u>	
Antônio Alves Bia Betrand Antônio Alves	Suéter Azul Calça Vison Suéter Azul	10-maio-2011 29-jun-2011 10-maio-2011	

Figura 11.12: Configuração da tabela BEAM\* com um exemplo de repetição de um item. Adaptado de:(Corr e Stagnitto, 2012)

adicionais sobre os detalhes que já estão lá.

Por exemplo, existindo o detalhe **produto** na tabela da Figura 11.16, um participante pode achar conveniente colocar uma coluna adicional “Tipo do Produto”. Isso não deve ser colocado na tabela como uma coluna, porém pode ser anotado sobre a coluna “Produto” e tratado mais tarde, na definição da dimensão.

### 11.3.6 Definindo onde

Nesse passo o importante é contar onde a história ocorre, lembrando que, sendo um intervalo de tempo, pode ocorrer em mais de um lugar. A existência de mais de uma coluna “quando” definindo momentos de tempo diferentes dá a dica que estes momentos podem ocorrer em lugares diferentes.

A Figura 11.17 mostra a continuação do exemplo, onde são criadas duas colunas para marcar a loja onde foi feita o pedido e o endereço de entrega. É importante notar que não aparece o endereço da loja, pois seria novamente um caso de detalhe do detalhe, que não deve ser usado nesse instante.

### 11.3. PASSO A PASSO DA MODELAGEM DE EVENTOS DE NEGÓCIO121

<b>CLIENTE</b>	<b>encomenda</b>	<i>na</i> <b>DATA</b>
	<b>PRODUTO</b>	<b>DE PEDIDO</b>
<b>&lt;who&gt;</b>	<b>&lt;what&gt; MD</b>	<b>&lt;quando&gt; MD</b>
Antônio Alves	Suéter Azul	10-maio-2011
Bia Betrand	Calça Vison	29-jun-2011
Antônio Alves	Suéter Azul	10-maio-2011
Carla Caril	Camisa Bege	14-out-2011
<desconhecido>	Sapato Salto 28	10 anos atrás

Figura 11.13: Configuração da tabela BEAM\* com um dados faltando.

Adaptado de:(Corr e Stagnitto, 2012)

#### 11.3.7 Encontrando quantos

Ao chegar ao ponto da criação da história que indica “quantos” deixa-se de tratar das dimensões ao redor da Tabela Fato para tratar especificamente das informações que podem ser colocadas na mesma.

Isto porque essas colunas vão informar quantidades, que normalmente são aditivas e são exatamente os que precisamos na Tabela Fato.

Continuando com o exemplo, a história pode ser estendida com as colunas “quantidade” e “preço”, como na Figura 11.18

CLIENTE	encomenda	na DATA	DE PEDIDO
	<who>	<what> MD	<quando> MD
Antônio Alves	Suéter Azul	10-maio-2011	
Bia Betrand	Calça Vison	29-jun-2011	
Antônio Alves	Suéter Azul	10-maio-2011	
Carla Caril	Camisa Bege	14-out-2011	
<desconhecido>	Sapato Salto 28	10 anos atrás	
Colégio LMS	Camisa Branca	ontem	
Empresa EPM	Calça+Camisa	ontem	

Figura 11.14: Configuração da tabela BEAM\* ao final deste passo.  
Adaptado de:(Corr e Stagnitto, 2012)

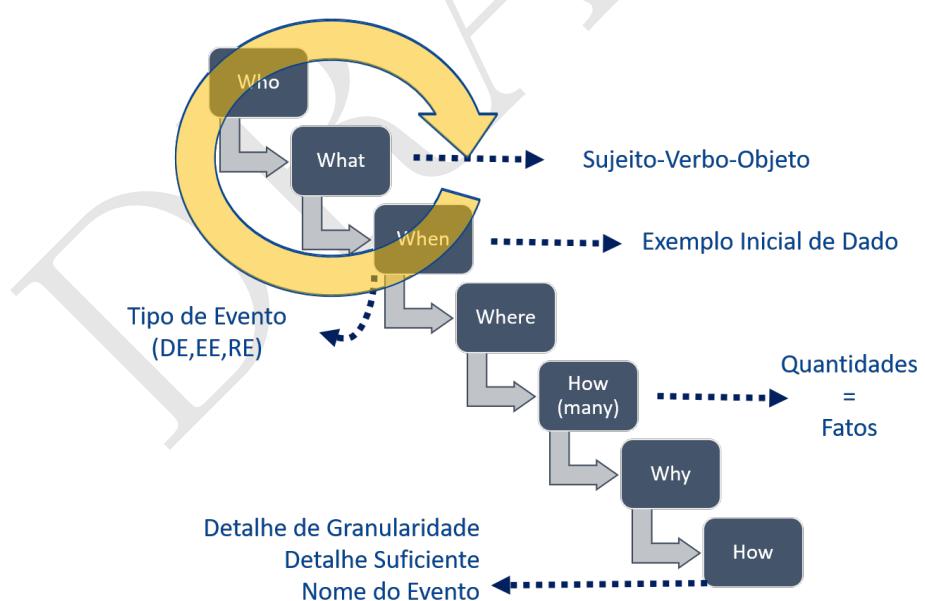


Figura 11.15: Os três primeiros passos do processo podem ser repetidos para exaurir os conceitos de “quem” e “quanto”. Adaptado de:(Corr e Stagnitto, 2012)

### 11.3. PASSO A PASSO DA MODELAGEM DE EVENTOS DE NEGÓCIO123

CLIENTE	encomenda	na	para entrega em	com
	PRODUTO	DATA DE PEDIDO	DATA DE ENTREGA	VENDEDOR
<who>	<what> MD	<quando> MD	<quando>	<who>
Antônio Alves Bia Betrand Antônio Alves Carla Caril <desconhecido> Colégio LMS Empresa EPM	Suéter Azul Calça Vison Suéter Azul Camisa Bege Sapato Salto 28 Camisa Branca Calça+Camisa	10-maio-2011 29-jun-2011 10-maio-2011 14-out-2011 10 anos atrás ontem ontem	12-maio-2011 2-julho-2011 12-maio-2011 não aplicável não aplicável dagui a 3 dias dagui a 10 dias	Zenaide Zimbro Xenia Xavier Zenaide Zimbro Yan Yakiza William Wonder N/A Equipe Comercial

Figura 11.16: Uma tabela BEAM\* com dois “quem” e dois “quando”.  
(Corr e Stagnitto, 2012)

CLIENTE	encomenda	na	para entrega em	na	para
	PRODUTO	DATA DE PEDIDO	DATA DE ENTREGA	LOJA	ENDERECO
<who>	<what> MD	<quando> MD	<quando>	<where>	<where>
Antônio Alves Bia Betrand Antônio Alves Carla Caril <desconhecido> Colégio LMS Empresa EPM	Suéter Azul Calça Vison Suéter Azul Camisa Bege Sapato Salto 28 Camisa Branca Calça+Camisa	10-maio-2011 29-jun-2011 10-maio-2011 14-out-2011 10 anos atrás ontem ontem	12-maio-2011 2-julho-2011 12-maio-2011 não aplicável não aplicável dagui a 3 dias dagui a 10 dias	SEDE amazon.com SEDE 1-800-roupas 1-800-roupas COPA 2 IPANEMA	Rua Seca 23, 22010-222 Rua Arco Íris 123, 10021-101 Rua Seca 23, 22010-222 Av. Américas 2, 20021-11 Rua Salto 199, 42921-394 Não aplicável Não aplicável

Figura 11.17: Uma tabela BEAM\* descreve uma história que acontece em dois lugares, um em cada ponto no tempo. Por espaço algumas colunas já encontradas foram suprimidas. Fonte:(Corr e Stagnitto, 2012)

CLIENTE	encomenda	na	para entrega em	QUANTIDADE	por
	PRODUTO	DATA DE PEDIDO	DATA DE ENTREGA		PRECO
<who>	<what> ID	<quando> ID	<quando>	<unidade>	<R\$ , US\$>
Antônio Alves	Suéter Azul	10-maio-2011	12-maio-2011	1	R\$ 200
Bia Betrand	Calça Vison	29-jun-2011	2-julho-2011	1	US\$ 95
Antônio Alves	Suéter Azul	10-maio-2011	12-maio-2011	1	R\$ 200
Carla Carl	Camisa Bege	14-out-2011	não aplicável	1	R\$ 140
<desconhecido>	Sapato Salto 28	10 anos atrás	não aplicável	50	R\$ 200
Colégio LMS	Camisa Branca	ontem	daqui a 3 dias	100	R\$ 30400
Empresa EPM	Calça+Camisa	ontem	daqui a 10 dias	100	R\$ 20000

Figura 11.18: Introduzindo na tabela BEAM\* as informações quantitativas. Por espaço algumas colunas já encontradas foram suprimidas. Fonte:(Corr e Stagnitto, 2012)

## 11.4 Definindo por que

Este tipo de coluna pode ser um pouco mais difícil de criar porque exige dos participantes o conhecimento de possíveis motivações para o evento que tiveram acontecido ao longo do período estudado e que possam estar registrada na base operacional.

Exemplos de motivações ligados à área de venda são: promoções, lançamentos de produtos, descontos, campanhas publicitárias, etc. Outras áreas podem ter motivações diferentes. Um hospital, por exemplo, pode estar estudando a procura a seu pronto atendimento, e as motivações podem estar relacionadas a temperatura ou clima do dia, existência de grandes eventos, feriados longos, etc.

Possivelmente, estas colunas são exploratórias, isto é, o conhecimento do negócio permite fazer uma premissa que esses são motivos válidos para influenciar, de alguma forma, a ocorrência dos eventos. O sistema criado permitirá analisar o verdadeiro impacto desses motivos. A Figura 11.19 mostra um exemplo onde descontos são identificados.

CLIENTE	encomenda	na:	para entrega em:	por	com
	PRODUTO	DATA DE PEDIDO	DATA DE ENTREGA	PROMOÇÃO	DESCONTO
<who>	<what> MD	<quando> MD	<quando>	<why>	<R\$, US\$>
Antônio Alves	Suéter Azul	10-maio-2011	12-maio-2011	Sem Promoção	0
Bia Bertrand	Calça Vison	29-jun-2011	2-julho-2011	Promoção	10%
Antônio Alves	Suéter Azul	10-maio-2011	12-maio-2011	Sem Promoção	0
Carla Caril	Camisa Bege	14-out-2011	não aplicável	Lançamento	US\$10
<desconhecido>	Sapato Salto 28	10 anos atrás	não aplicável	Pro Teste	R\$ 1000
Colégio LMS	Camisa Branca	ontem	daqui a 3 dias	Novo Negócio	R\$ 5000
Empresa EPM	Calça+Camisa	ontem	daqui a 10 dias	Novo Negócio	R\$ 6000

Figura 11.19: Introduzindo na tabela BEAM\* as informações sobre por que. Por espaço algumas colunas já encontradas foram suprimidas.

Fonte:(Corr e Stagnitto, 2012)

## 11.5 Definindo como

A última coluna a ser criada e preenchida tem relação com como a história acontece ou é registrada no sistema da empresa.

Nesses dados, por exemplo, é possível guardar como uma venda é paga, ou seja, o método de pagamento, cartão, dinheiro, etc, ou como ela é registrada, como um pedido, uma nota fiscal, etc. A Figura 11.20 mostra a Tabela BEAM\* com a identificação do número do pedido.

Este é o último passo de preenchimento das colunas.

	<code>encomenda</code>	<code>na</code>	<code>para entrega em</code>		<code>por</code>	
<code>CLIENTE</code>	<code>PRODUTO</code>	<code>DATA DE PEDIDO</code>	<code>DATA DE ENTREGA</code>	<code>QUANTIDADE</code>	<code>PRECO</code>	<code>ID PEDIDO</code>
<code>&lt;who&gt;</code>	<code>&lt;what&gt; ID</code>	<code>&lt;quando&gt; ID</code>	<code>&lt;quando&gt;</code>	<code>&lt;unidade&gt;</code>	<code>&lt;R\$ , US\$&gt;</code>	<code>&lt; how &gt;</code>
Antônio Alves Bia Betrand Antônio Alves Carla Carl <code>&lt;desconhecido&gt;</code> Colégio LMS Empresa EPM	Suéter Azul Calça Vison Suéter Azul Camisa Bege Sapato Salto 28 Camisa Branca Calça+Camisa	10-maio-2011 29-jun-2011 10-maio-2011 14-out-2011 10 anos atrás ontem ontem	12-maio-2011 2-julho-2011 12-maio-2011 não aplicável não aplicável dagü a 3 dias dagü a 10 dias	1 1 1 1 50 100 100	R\$ 200 US\$ 95 R\$ 200 R\$ 140 R\$ 200 R\$ 30400 R\$ 20000	12312 11295 66343 23435 25232 63345 53435

Figura 11.20: Introduzindo na tabela BEAM\* as informações sobre como. Por espaço algumas colunas já encontradas foram suprimidas. Fonte:(Corr e Stagnitto, 2012)

### 11.5.1 Escolhendo a Granularidade

Com todas as colunas definidas e todos os exemplos colocados, temos a Tabela BEAM\* praticamente completa para uma história. O próximo passo é definir a granularidade

O objetivo é escolher um conjunto de colunas que crie uma identificação única do evento, no nível de detalhe desejado. Pode ser que uma única coluna já garanta, ou que mais de uma seja necessária. Essa escolha é anotada na Tabela BEAM\* com o código **GD**, para o inglês *granularity detail*.

Se, definido uma nível de detalhe, seja possível criar mais de uma linha na Tabela BEAM\* mostrando exemplos que para os quais as colunas escolhidas tem todas o mesmo valor, é necessário verificar se eles podem ser agregados de alguma forma.

No exemplo deste capítulo, foram escolhidas as colunas PRODUTO e ID PEDIDO. Isto significa que se em um pedido um produto aparecer duas vezes, todos os valores devem ser agregados.

### 11.5.2 Dando nome ao evento

O próximo e último passo desta parte do processo BEAM\* é dar o nome a história de dados e identificar o tipo do evento. A Tabela 11.2 mostra novamente os nomes dos tipos de eventos com seus códigos.

Tabela 11.2: Tipos de eventos e seus códigos. Baseado em:(Corr e Stagnitto, 2012)

Tipo de Evento	Código
Evento Discreto	ED
Evento Evolutivo	EE
Evento Recorrente	RE

Para indicar que a Tabela BEAM\* está totalmente definida, são traçadas duas linhas na sua margem direita. A Figura 11.21 mostra a versão final.

Além de manter um registro fotográfico detalhado do passo a passo, é aconselhável documentar a Tabela BEAM\* final em uma planilha eletrônica. Esta planilha pode ser usada para continuar o trabalho nos próximos passos.

### 11.5.3 Resumo do passo a passo

Resumindo o passo a passo, é importante:

- iniciar sempre pela história prioritária no momento;
- realizar os três primeiros passos, que definem quem-o que-quando até esgotar as possibilidades;
- definir os outros detalhes na ordem prescrita;
- definir a granularidade da tabela;
- dar o nome e determinar o tipo da tabela, e

PEDIDO DE CLIENTE <DE>							
CLIENTE	encomenda PRODUTO	na: DATA DE PEDIDO	para entrega em: DATA DE ENTREGA	QUANTIDADE	por: PREÇO	ID PEDIDO	
<who>	<what> MD GD	<quando> MD	<quando>	<unidade>	<R\$, US\$>	<how> GD	
Antônio Alves	Suéter Azul	10-maio-2011	12-maio-2011	1	R\$ 200	12312	
Bia Bertrand	Calça Vison	29-jun-2011	2-julho-2011	1	US\$ 95	11255	
Antônio Alves	Suéter Azul	10-maio-2011	12-maio-2011	1	R\$ 200	66343	
Carla Caril	Camisa Bege	14-out-2011	não aplicável	1	R\$ 140	25435	
<desconhecido>	Sapato Salto 28	10 anos atrás	não aplicável	50	R\$ 200	25232	
Colégio LMS	Camisa Branca	ontem	daqui a 3 dias	100	R\$ 30400	63345	
Empresa EPM	Calça+Camisa	ontem	daqui a 10 dias	100	R\$ 20000	53435	

Figura 11.21: Finalizando a Tabela BEAM\* com nome e tipo. Fonte:(Corr e Stagnitto, 2012)

- registrar fotograficamente e, possivelmente, copiar a tabela final para uma planilha eletrônica.

## 11.6 Modelando Dimensões

O próximo grande passo do método BEAM\* visa definir as dimensões a partir dos detalhes dos eventos. Nesse passo, então, os detalhes serão transformados em dimensões, por meio da elicitação e definição de todos os campos de cada dimensão.

Para isso são realizados quatro passos:

1. iniciar a tabela dimensão com os dados levantados na Tabela BEAM\*;
2. especificar os atributos da dimensão;
3. especificar as hierarquias existentes nas dimensões, e
4. Controlar como as colunas podem mudar.

Esse subprocesso, Modelar Dimensão, do processo Modelstorming definido na Figura 11.3, está descrito na Figura 11.22

No caso do exemplo, os detalhes que serão considerados como parte da história são:

- cliente (quem),

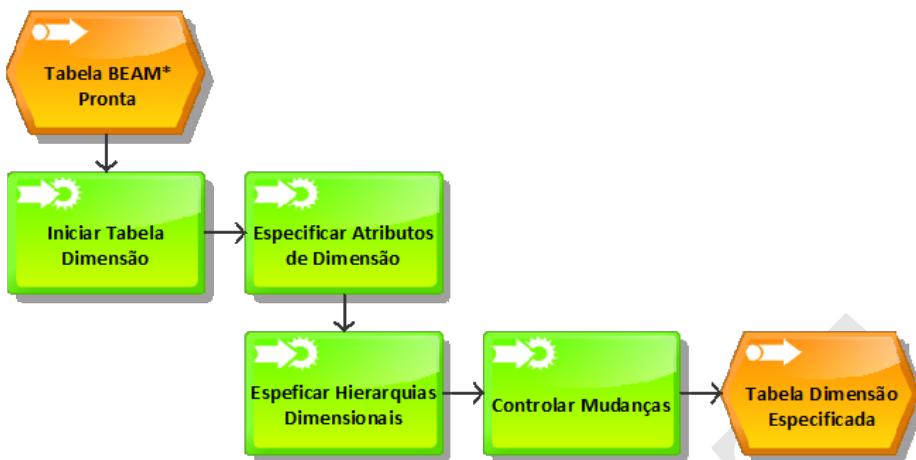


Figura 11.22: Modelar Dimensão descrito em ARIS-EPC.

- produto (o que),
- data do pedido (quando),
- data da entrega (quando),
- vendedor (quem),
- loja (onde),
- endereço de entrega (onde),
- promoção (por que), e
- id pedido (como).

Normalmente, em busca das dimensões, **não são tratadas** os detalhes do tipo **quanto**, que deverão pertencer a Tabela Fato.

### 11.6.1 Codificação Usada

Na prática, serão criadas tabelas que mostram como é cada dimensão, semelhantes as Tabelas BEAM\*. e as colunas destas tabelas serão anotadas com alguns códigos que permitem entender melhor algumas de suas características.

O códigos apresentados na Tabela 11.3 são usados neste passo.

Partindo da Tabela BEAM\* serão criadas novas tabelas que representam as dimensões. A ordem de leitura da Tabela BEAM\* para gerar as Tabelas Dimensão é da esquerda para a direita, logo, no exemplo, é preciso começar pela dimensão CLIENTE.

Definida a Tabela Dimensão que será elicitada, é feito então um novo

Tabela 11.3: Códigos usados na modelagem dos campos da dimensão

Código	Significado	Explicação
MD	<b>Mandatory</b>	Campo obrigatório
BK	<b>Business Key</b>	Chave única do negócio que identifica o membro da dimensão
DC	<b>Defined by Characteristic</b>	Campo que precisa ser definido por uma característica, um tipo
X1, X2, ...	<b>eXclusive</b>	Campos exclusivos de alguns tipos e não usados para outros
FV	<b>Fixed Value</b>	Única mudança possível é uma correção
HV	<b>Historical Value</b>	Mudanças acontecem e são rastreadas
CV	<b>Current Value</b>	Dispensa histórico, basta o valor atual

passo a passo para gerar a definição dos seus campos.

### 11.6.2 Início da tabela

O primeiro passo é transferir a informação usada na Tabela BEAM\* para o início de um novo desenho.

Desta forma se inicia a Tabela Dimensão dando um nome a mesma (Cliente), normalmente usando o próprio nome definido na Tabela BEAM\* para o detalhe. Os exemplos usados na Tabela BEAM\* são, a seguir, transferidos e usados para definir o nome do campo.

No exemplo, o nome escolhido é **Nome do Cliente**. Nesse momento, como o trabalho ainda é feito com os usuários, deve-se pensar apenas em nomes lógicos e não em regras arbitrárias usadas na organização no processo de molagem física do banco de dados.

Definido o nome da coluna, usa-se uma linha abaixo dele para fazer marcações que indicam características dessa coluna. No caso de “Nome do Cliente” o exemplo exige que o campo seja **mandatório**, o que indicado pela notação **MD**. Esse trabalho pode ser verificado na Figura 11.23.

## Cliente

Nome do Cliente	
<b>MD</b>	
Antônio Alves	
Bia Bertrand	
Carla Caril	
Unknown	
Colégio LMS	
Empresa EPM	

Figura 11.23: Início da definição da dimensão Cliente. Fonte: (Corr e Stagnitto, 2012)

### 11.6.3 Granularidade e chaves

A partir da definição do campo e da informação principal que o caracteriza, é necessário definir a granularidade e, por consequência, as colunas que definem a chave principal do campo.

Esta chave deve possuir as características de serem únicas, estáveis e obrigatórias. Além disso, é essencial que sejam chaves ligadas ao negócio. São marcadas com o código **BK**, significando **Business Key**.

Para definir a chave de negócios, o exemplo escolhe o identificador único que um cliente possui na organização sendo modelada, o que pode ser visto na Figura 11.24. Normalmente toda chave é mandatária.

Estes dois passos definem o início da Tabela Dimensão. Os próximos passos buscam a completar a informação para a Dimensão poder auxiliar as consultas das partes interessadas.

### 11.6.4 Os Atributos de Dimensão

Para descobrir que atributos são necessários para a Tabela Dimensão, é importante pesquisar como essas informações serão usadas, isto é, quais os dados

## Cliente

Nome do Cliente	ID Cliente
<b>MD</b>	<b>BK, MD</b>
Antônio Alves	C0012
Bia Bertrand	C9533
Carla Caril	C2321
Unknown	N/A
Colégio LMS	B0012
Empresa EPM	B2323

Figura 11.24: Definição da chave de negócios para a dimensão Cliente.

Fonte: (Corr e Stagnitto, 2012)

são interessantes para estar nos relatórios, gráficos e consultas em geral que serão obtidas do Data Warehouse.

Os participantes devem buscar atributos fazendo as seguintes perguntas:

- que atributos serão usados como colunas em relatórios;
- que atributos serão usados para ordenar relatórios;
- que atributos serão usados para filtrar relatórios, e
- que atributos serão, e usados para agrupar itens de relatório e fazer totalizações ou outras operações de agregação, como média, porcentagem, etc?

Também é importante aplicar a cada dimensão as perguntas 5W2H, como:

- quem está associada a esse item,
- que datas são importantes para esse item,
- onde está esse item,
- que quantidades definem item, e
- por que esse item aparece?

Essas perguntas são apenas exemplo. No caso de uma Venda, por exemplo, na Dimensão “Cliente”, é possível perguntar onde mora o cliente, em que data ele nasceu, quantas pessoas há na sua família, etc.

### Atributos de Tipo

Alguns tipos de atributos podem ser buscados ativamente. Atributos muito importantes são aqueles que classificam o assunto do domínio de alguma forma, ou seja, aqueles que definem um “tipo de” em relação ao atributo. Exemplos são tipo de cliente, marca de produto, tipo de exame, etc.

A Figura 11.25 mostra a escolha de um Tipo de Cliente. O atributo foi definido como mandatório. Os tipos identificados são “Consumidor” e “Empresa”.

<b>Cliente</b>	
<b>Nome do Cliente</b>	<b>ID Cliente</b>
<b>MD</b>	<b>BK, MD</b>
Antônio Alves	C0012
Bia Bertrand	C9533
Carla Caril	C2321
Unknown	N/A
Colégio LMS	B0012
Empresa EPM	B2323

Figura 11.25: Definição de um tipo para a Dimensão Cliente. Fonte: (Corr e Stagnitto, 2012)

### Atributos Múltiplos

Alguns atributos, descrito informalmente, podem parecer atributos múltiplos. Um exemplo claro disso é a possibilidade de um mesmo evento de negócios usar vários atributos do tipo “endereço”, como “endereço de entrega” e “endereço de cobrança”.

Quando o número de atributos múltiplos é fixo, como no caso exemplificado, eles podem ser colocados na mesma tabela dimensão, pois não é problema a quantidade de campos que a tabela tem.

Porém, quando o atributo tem múltiplos valores e não é possível determinar sua cardinalidade, na prática eles não podem fazer parte da dimensão como definida no momento. Isso pode acontecer, por exemplo, se um cliente pode ter vários endereços de entrega, e que não existe um número máximo de endereços.

A solução pode ser colocá-los em outra dimensão, colocá-los em outro modelo dimensional ou alterar a granularidade da dimensão.

Este problema é semelhante ao que ocorre em modelos ER ou Relacionais. Se um conjunto de atributos se repete um número não controlado de vezes, a solução é criar uma tabela para aquele conjunto de atributos. Por exemplo, se uma pessoa pode ter vários telefones, há duas opções: limitar o número de telefones, por exemplo em 3, e criar 3 campos na tabela pessoa, ou criar uma tabela adicional para colocar quantos telefones forem necessários para essa pessoa.

Kimball et al. (2008) cita o exemplo do caso da Tabela Fato “saldo de conta de banco”. Esta Tabela Fato se relaciona de forma N:1 com a Tabela Dimensão “Conta de Banco”. Porém, em uma conta de banco deve haver um, mas podem haver mais correntistas, o que leva a impossibilidade de colocar esse dados, quem possui a conta, na tabela. A solução proposta, uma extensão ao Modelo Estrela, é usar Dimensões M:N, na forma de Tabelas Ponte(Kimball et al., 2008, p.6.4).

### Atributos exclusivos

Quando a dimensão tem um tipo, é possível que o valor do tipo defina a necessidade de um atributo que só existe se o item for daquele tipo. Por exemplo, no caso do exemplo, o “Cliente” do tipo “Consumidor” tem um atributo “Sexo”, enquanto o “cliente” do tipo “Empresa” em um atributo “CNPJ”.

Se isso acontece, é necessário não só criar colunas para os atributos, mas também definir que coluna define a sua existência e associá-los a grupos distintos.

Para isso se usa a marcação **DC**, para **Definition Column**, na coluna que define o tipo que divide os atributos, e a notação **X1,X2,...** para as colunas que são exclusivas de um tipo. O exemplo citado aparece na Figura 11.26.

<b>Cliente</b>				
Nome de Cliente	ID Cliente	Tipo de Cliente	Sexo	CNPJ
<b>MD</b>	<b>BK, MD,DC</b>	<b>MD</b>	<b>X1,MD</b>	<b>X2,MD</b>
Antônio Alves	C0012	Consumidor	M	N/A
Bia Bertrand	C9533	Consumidor	F	N/A
Carla Caril	C2321	Consumidor	F	N/A
Unknown	N/A	N/A	Desconhecido	N/A
Colégio LMS	B0012	Empresa	N/A	23929939291-1299
Empresa EPM	B2323	Empresa	N/A	40912839120-1453

Figura 11.26: Definição dos atributos exclusivos. Fonte: (Corr e Stagnitto, 2012)

### 11.6.5 Prevendo mudanças

A última anotação em uma Tabela de Dimensão é a previsão de mudanças no mesmo. Isso é importante e Kimball et al. (2008) propõe vários métodos distintos para o tratamento de mudanças que podem ser aplicados após esse passo.

Cada coluna deve receber uma marcação adicional que indica como podem ocorrer mudanças na mesma. Os valores possíveis são:

**FV**, ou *Fixed Value*, que indica que o valor é fixo e mudanças podem ocorrer apenas por motivos de correção;

**HV** ou *Historical Value*, onde ocorrerão mudanças e todas devem ser rastreadas, o que pode exigir campos adicionais e outras estratégias de controle;

**CV** ou *Current Value*, onde o valor é alterado, mas só é preciso guardar o valor atual, e

**HV/CV ou CV/HV** onde há uma combinação de comportamentos segundo regras de negócio, porém o primeiro valor é o default.

No exemplo em questão, todos os valores são fixos. Em uma dimensão “Produto”, por exemplo, o “preço do produto” poderia ser definido como **CV** ou **HV** ou combinações, dependendo da necessidade das partes interessadas na organização.

### 11.6.6 Construindo a hierarquia da dimensão

A partir da tabela de dimensão, e possivelmente provando correções na mesma, é necessário agora documentar as hierarquias que existem nesse dimensão. É importante notar que o Método BEAM\* segue o Modelo Estrela,

e não pretende criar o Snowflake, mas se for desejado pela organização, este é o ponto em que o modelo pode ser adaptado para levantar as hierarquias.

Existem vários tipos de hierarquia, como mostra a Figura 11.27. Nela aparecem hierarquias com pai único e com múltiplos pais, e hierarquias balanceadas, desequilibradas e de profundidade variável.

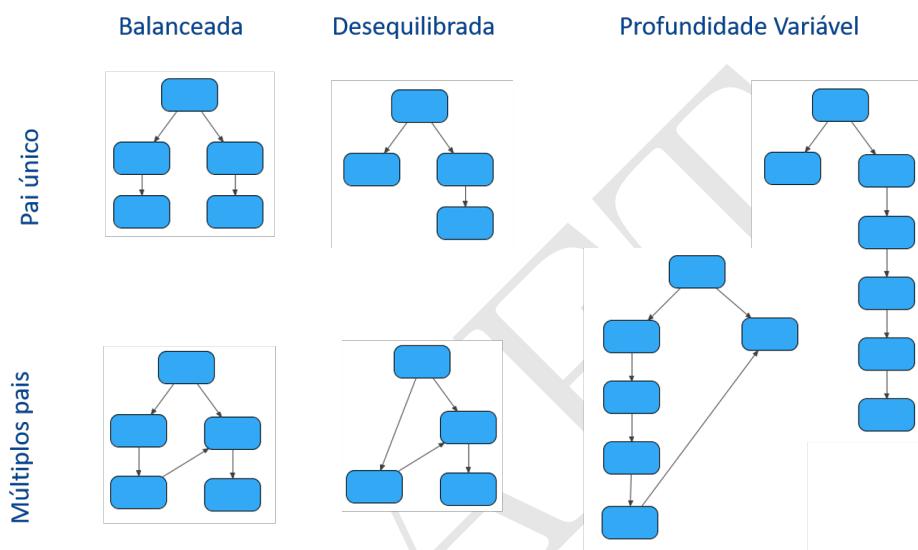


Figura 11.27: Tipos de hierarquias para dimensões. Fonte: (Corr e Stagnitto, 2012)

Mapas de hierarquia mostram de maneira simples essas hierarquias e podem ser vistos nas Figuras 11.28 e 11.29. A Figura 11.29, especialmente, mostra duas notações alternativas para hierarquias com *loops*.

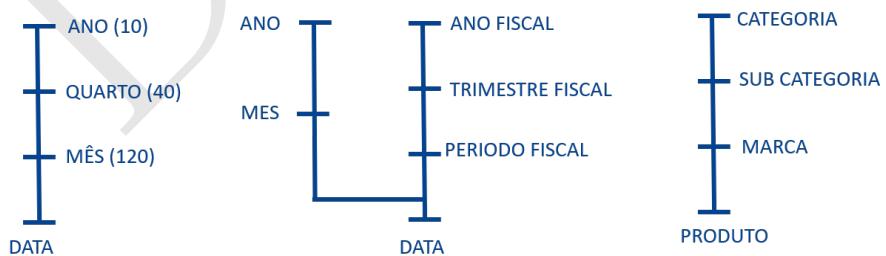


Figura 11.28: Hierarquias para dimensões de data, com duas versões, e produto. Fonte: (Corr e Stagnitto, 2012)

Esses mapas podem ser usados para marcar como são feitas as consultas esperadas, como na Figura 11.30.



Figura 11.29: Notações para hierarquias com loops. Fonte: (Corr e Stagnitto, 2012)

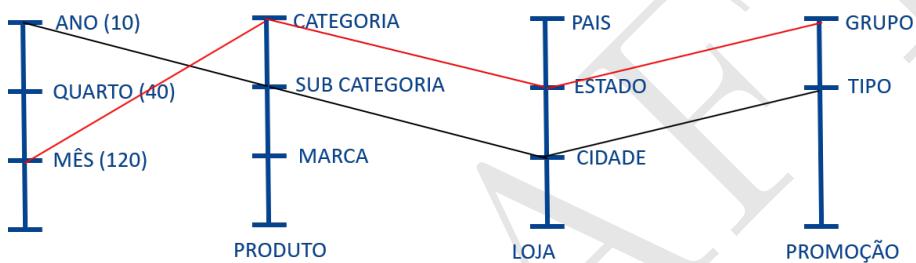


Figura 11.30: Notações para consultas perpassando hierarquias. Fonte: (Corr e Stagnitto, 2012)

Para levantar as hierarquias é importante saber se os itens inferiores podem pertencer a mais um item superior, ou se um item superior pode ter mais de um item inferior simultaneamente.

## 11.7 Matriz de Evento

Corr e Stagnitto (2012) apresenta o seguinte exemplo de Matriz de Evento.

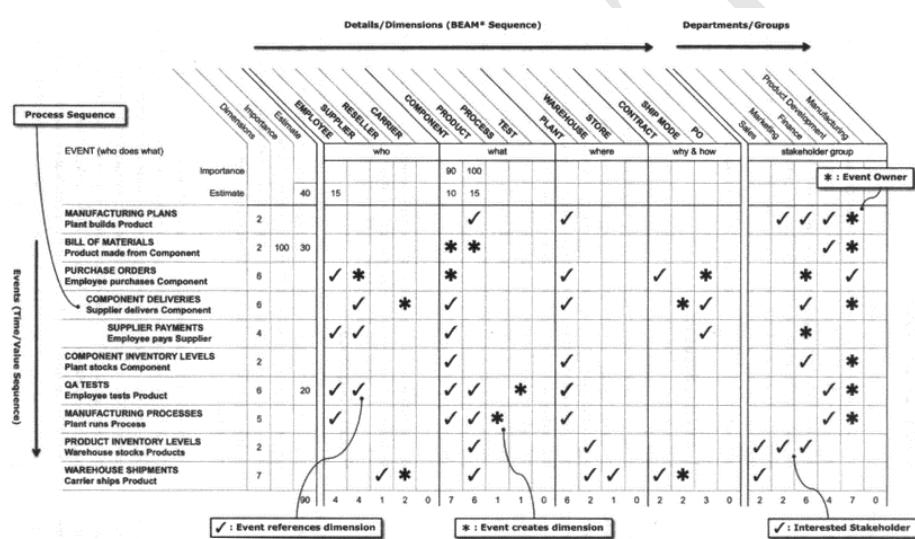


Figura 11.31: Exemplo de Matriz de Evento. Fonte: (Corr e Stagnitto, 2012)

# Índice

- 7W, 105  
ARIS, 71  
assuntos, 102  
atividades, 74  
atributos, 20  
  
Banco de Dados Relacional, 17  
Beam\*, 105  
BI, 99  
Bill Inmon, 101  
BK, 131  
Business Event Analysis and Modeling, 105  
Business Intelligence, 99  
Business Key, 131  
  
Carta de Hierarquia, 106  
Commercial Of The Shelf, 93  
computador, 7  
COTS, 93  
CREATE TABLE, 22  
  
dado, 6, 7  
Data Control Language, 22  
Data Definition Language, 22  
Data Definition Languagem, 22  
Data Manipulation Language, 22  
Data Marts, 98  
Data Query Language, 22, 24  
Data Transaction Language, 22  
Data Warehouse, 6, 96, 102  
DC, 134  
  
DCL, 22  
DDL, 22  
deadlock, 84  
Definition Column, 134  
detalhes, 112, 114  
dice, 35  
DML, 22  
do nothing branch, 87  
domínio, 19  
DQL, 22  
drill across, 31  
drill down, 40  
drill down and roll up, 31  
drill up, 41  
DROP TABLE, 23  
DTL, 22  
  
entidades, 13  
Entreprise Resource Planning, 10, 93  
EPC, 71  
    Loops, 83  
    paralelismo, 86  
ER, 13  
ERP, 9, 10, 93  
esquema de relação, 20  
evento, 76  
Evento de Negócio, 108  
evento discreto, 109  
evento evolutivo, 109  
evento recorrente, 109  
eventos, 75

- filtro, 35  
fluxo de controle, 73  
fluxo de execução, 73  
Framework 7W, 106  
GD, 126  
grau da relação, 20  
grupo, 117  
histórias de dados, 108  
informação, 6, 7  
instância, 14  
instância de entidade, 14  
interativo, 9  
interface de processo, 78  
interfaces de processos, 78  
junção, 24  
Linhas de Tempo, 107  
mandatório, 130  
Matriz de Eventos, 107  
MER, 13  
modelagem ágil de dados, 105  
Modelagem de negócio, 59  
modelagem de processos, 60  
Modelagem de Processos de Negócio, 60  
Modelo de Entidades e Relacionamentos, 13  
Modelo de Entidades Relacionamentos, 13  
Modelo Estrela Aumentado, 107  
Modelo Relacional, 17  
OLAP, 31  
OLTP, 31  
Online Analytical Processing, 31  
Online Transaction Processing, 31  
operações OLAP, 31  
ordem de descoberta, 107  
organização, 5  
organograma, 63  
pivot, 31, 44  
pivot table, 33  
planilhas eletrônicas, 33  
processamento de transações em tempo real, 31  
processo, 59  
processo de negócio, 60  
Ralph Kimball, 101  
reativo, 9  
regras, 77  
relação, 20  
respostas planejadas, 9  
roll up, 41  
SGDB, 21  
short codes BEAM\*, 107  
sistema, 7  
Sistema de Informação, 8  
Sistemas de Gestão Empresarial, 10  
Sistemas de Informação, 6, 7  
sistemas de informação automatizados, 8  
Sistemas Gerenciadores de Banco de Dados, 21  
slice, 35  
slice and dice, 31, 35  
SQL, 21  
Structured Query Language, 21  
Tabela Beam\*, 106, 107  
tabela fato, 103  
tabelas dimensão, 103  
tema, 116  
tipo de dados, 19  
tipo de entidade, 13  
X1, 134  
X2, 134

# Bibliografia

- Celko, Joe (2005). *Joe Celko's SQL for smarties: advanced SQL programming*. Third. The Morgan Kaufmann series in data management systems. Los Altos, CA 94022, USA: Morgan Kaufmann Publishers, pp. xxviii + 808. ISBN: 0-12-369379-9 (paperback). URL: <http://www.loc.gov/catdir/enhancements/fy0626/2005279919-d.html>; %20<http://www.loc.gov/catdir/enhancements/fy0626/2005279919-t.html>.
- Codd, E. F. (jun. de 1970). “A relational model of data for large shared data banks”. Em: *Communications of the ACM* 13.6, pp. 377–387. doi: 10.1145/362384.362685.
- Columbus, Louis (2014). *Gartner's ERP Market Share Update Shows The Future Of Cloud ERP Is Now*. <https://www.forbes.com/sites/louiscolumbus/2014/05/12/gartners-erp-market-share-update-shows-the-future-of-cloud-erp-is-now/>. Accessed: 2019-10-15.
- Corr, Lawrence e Jim Stagnitto (2012). *Agile Data Warhouse Design. Agile Data Warhouse Design: Collaborative Dimensional Modeling from Whiteboard to Star Schema. Collaborative Dimensional Modeling from Whiteboard to Star Schema*. Leeds, UK: Decision One Press.
- Cougo, Paulo (1999). *Modelagem Conceitual e Projeto de Banco de Dados*. Rio de Janeiro: Campus.
- Date, C. J. (2004). *An introduction to database systems*. Eighth. Boston, MA, USA: Pearson/Addison Wesley, pp. xxvii + 983 + 22. ISBN: 0-321-19784-4.
- Elmasri, Ramez e Shamkant B. Navathe (2016). *Fundamentals of Database Systems*. 7th Edition. Boston: Pearson.
- Heuser, Carlos A. (2001). *Projeto de Banco de Dados*. Vol. 4. Série Livros Didáticos: Instituto de Informática da UFRGS. Porto Alegre: Editora Sagra Luzzatto.
- INEP, Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (5 de set. de 2016). *Microdados do Exame Nacional do Ensino Médio - Enem 1998*. <http://portal.inep.gov.br/basica-levantamentos->

- acessar. Acessado em 25-10-2109. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira - INEP. URL: %5Curl%7B%20ftp://ftp.inep.gov.br/microdados/micro\_enem1998.zip%7D (acedido em 26/10/2019).
- Inmon, William H. (1992). *Building the Data Warehouse*. Wellesley, MA, USA: John Wiley & Sons. ISBN: 0-89435-404-3.
- (2005). *Building the Data Warehouse*. Fourth edition. Wiley Publishing, Inc. ISBN: 978-0-7645-9944-6.
- ISO/IEC (jul. de 2004). *ISO/IEC 5218:2004 Information technology — Codes for the representation of human sexes*. ISO/IEC.
- Kempe, Shannon e Paul Williams (ago. de 2012). *A Short History of Data Warehousing*. <https://www.dataversity.net/a-short-history-of-data-warehousing>. Acessado em 25/10/2019. URL: <https://www.dataversity.net/a-short-history-of-data-warehousing/>.
- Kimball, Ralph et al. (2008). *The Data Warehouse Lifecycle Toolkit: Practical Techniques for Building Data Warehouse and Business Intelligence Systems*. 2nd. Wiley.
- Laudon, Kenneth e Jane Laudon (2011). *Sistemas de Informação Gerencias*. 9<sup>a</sup> edição. São Paulo: Pearson.
- Microsoft (27 de out. de 2019). *Microsoft Excel 365*. [Computer Software]. M.
- Motl, Jan (2019). *Adventure Works*. URL: %5Curl%7B<https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15%7D>.
- Oracle Corporation (2019a). *License for the Sakila Sample Database*. <https://dev.mysql.com/doc/sakila/en/sakila-license.html>. Accessed: 2019-10-23.
- (2 de nov. de 2019b). *MySQL 8.0 Reference Manual: Including MySQL NDB Cluster 8.0*. Acessado em 2019-11-02. Oracle Corporation. URL: %5Curl%7B<https://dev.mysql.com/doc/refman/8.0/en/%7D>.
- (2019c). *Sakila Sample Database*. <https://dev.mysql.com/doc/sakila/en/>. Accessed: 2019-10-23.
- Panko, Raymond R. (1998). “What We Know About Spreadsheet Errors”. Em: *Journal of End User Computing's* 10.2 (Spring 1998). Revised Mauy 2008, p. 15.21. URL: %5Curl%7B<http://panko.shidler.hawaii.edu/SSR/Mypapers/whatknow.htm%7D> (acedido em 04/11/2019).
- Pressman, Roger e Bruce Maxim (2016). *Engenharia de Software-8<sup>a</sup> Edição*. McGraw Hill Brasil.
- Ralph Kimball, Margy Ross (12 de jul. de 2013). *The Data Warehouse Toolkit*. Wiley John + Sons. 608 pp. ISBN: 1118530802. URL: <https://www>.

- ebook.de/de/product/20197316/ralph\_kimball\_margy\_ross\_the\_data\_warehouse\_toolkit.html.
- Scheel, Henrik von et al. (2015). "Phase 1: Process Concept Evolution". Em: *The Complete Business Process Handbook*. Elsevier, pp. 1–9. DOI: 10.1016/b978-0-12-799959-3.00001-x.
- Scott, Dale (2016). *MySQL version of Northwind demo database*. <https://github.com/dalers/mywind>. Accessed: 2019-10-23.
- Software AG (2019a). *Event*. Help ↴ ARIS Express ↴ Glossary ↴ Event. acessado em 2019-11-01. Aris Community by Software AG. URL: %5Curl%7B<https://www.ariscommunity.com/help/arис-express/35908%7D> (acedido em 01/11/2019).
- (2019b). *Rule*. Help ↴ ARIS Express ↴ Glossary ↴ Rule. Help ↴ ARIS Express ↴ Glossary ↴ Rule. acessado em 1/11/2019. Softwre AG. URL: %5Curl%7B<https://www.ariscommunity.com/help/arис-express/35909%7D> (acedido em 01/11/2019).
- Soto, Christiane (2019). *12 of the Biggest Spreadsheet Fails in History*. Acessado em 2019-11-04. Oracle Small-to-Medium Business Blog. URL: %5Curl%7B12%20of%20the%20Biggest%20Spreadsheet%20Fails%20in%20History%7D (acedido em 04/11/2019).
- United States Code (jul. de 2002). *Sarbanes-Oxley Act of 2002, PL 107-204, 116 Stat 745*. Codified in Sections 11, 15, 18, 28, and 29 USC.
- Wells, Dave e Eav Nahari (2019). *Modernizing the Legacy Data Warehousr: What, Why and How*. Acessado em 2019-11-3. Cloudera. URL: %5Curl%7B<https://www.slideshare.net/cloudera/modernizing-the-legacy-data-warehouse-what-why-and-how-12319%7D> (acedido em 03/11/2019).

DRAFT

# Apêndice A

## Exercício Um

O exercício tem como finalidade usar o software PENTAHO Data Integration - Kettle Spoon - Hitachi Vantara - para criar um Data Warehouse a partir da base Sakila(Oracle Corporation, 2019c).

Para isso deve ser criada uma Tabela Fato Aluguel, baseada nas tabelas *Rental* e *Payment*, e as seguintes Tabelas Dimensão: Data, Cliente, Funcionario<sup>1</sup>, Loja, Linguagem e Filme. A Figura A.1 mostra o modelo dimensional a ser criado. A Tabela A.1 mostra a relação básica dessas tabelas do Data Warehouse com a base Sakila original.

Tabela A.1: Relação entre as tabelas do DW e as tabelas da base Sakila.

Tabela do DW	Tabela Original
Aluguel	rental e payment
Data	rental e payment
Cliente	customer e address
Funcionario	staff e store
Loja	store e address
Filme	film
Linguagem	language e film

Há 3 ligações de Data com Aluguel: Data de Retirada, Data de Devolução, Data de Pagamento. Isso implica em 3 chaves estrangeiras na Tabela Fato para a mesma Tabela Dimensão. Não é necessário modelar o inventário, ou seja, que fita/CD foi retirado, só o filme. Aluguel e pagamento são de certa

<sup>1</sup>Sem acentos para não confundir os nossos bancos de dados.

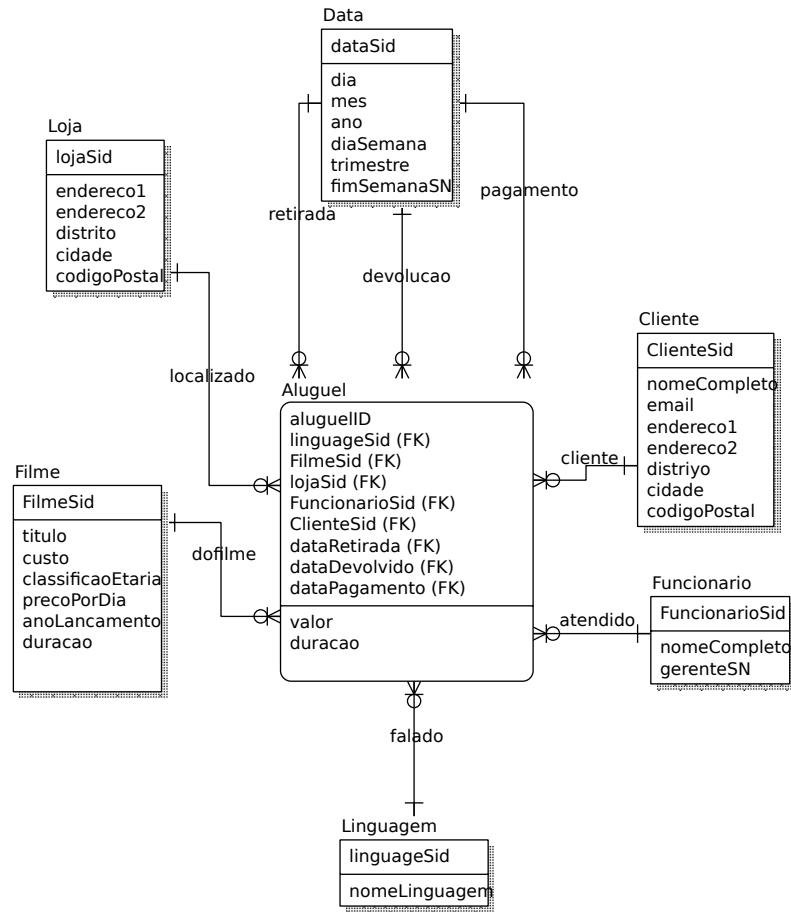


Figura A.1: Modelo estrela para a Data Warehouse do exercício

forma unificados, perceber que a ligação é 1:1 no modelo, mas que pode existir um sem o outro. Alguns campos devem ser calculados, o que é indicado na Tabela

Tabela A.2: Campos a serem calculados

Tabela	Campo	Cálculo
Aluguel	duracao	diferença entre as datas de entrega e retirada da fita
Funcionário	gerenteSN	pelo relacionamento entre <i>store</i> e <i>staff</i> .
Data	fimDeSemana	S se for sábado ou domingo, N caso contrário
Data	diaDaSemana	segunda, terça... de acordo com o dia da semana

## A.1 Algumas questões

Por que, nesse modelo, ATOR não pode ser dimensão?

Nesse modelo, ator e aluguel tem um relacionamento NxM, o que não é facilmente compatível com o modelo estrela. Existem soluções que são complicadas para esse exercício, mas podem ser tentadas em outro contexto.

Nesse modelo é pedido que todos os relacionamentos FATO - DIMENSÃO sejam 1xN, como é o padrão.

Solução alternativa para ter essa informação no Data Warehouse: Criar um fato que fosse ALUGUEL/ATOR/FILME, possivelmente em outro modelo dimensional, ou criar

No caso do fato ALUGUEL/ATOR/FILME, não poderia ter o valor do aluguel, pois não se poderia somar (já que um filme tem vários atores, o valor do aluguel sairia multiplicado). Uma divisão pró-rata do valor do aluguel entre todas as instâncias desse tipo não seria AMIGÁVEL ao USUÁRIO, logo também não seria uma boa solução.

No caso de outro modelo dimensional com ALUGUEL/ATOR/FILME, não haveria valores financeiros ou de data, contando apenas o número de aluguéis (FACTLESS TABLE)

Aplicar a técnica de tabela ponto usada em saldo de conta de banco com cliente, onde uma conta pode ter vários cliente, mas que quebra o modelo estrela.

## A.2 Orientações

Voce precisa de 2 esquemas, o sakila e um outro, por exemplo, sakiladw no MySQL. As transformações devem levar de uma base para a outra.

As criações de tabelas devem ser feitas diretas no MYSQL, mas terão que ser apresentadas no trabalho final, guarde os s ou gere eles.

Todo o trabalho de transferência deve ser feito no PENTAHO.

Perguntas podem ser feitas na aula ou no Grupo Whats App

## A.3 Dicas de implementação

Como gerar corretamente chaves artificiais pode ser visto em [https://etl-tools.info/en/examples/surrogate-key\\_pdi.htm](https://etl-tools.info/en/examples/surrogate-key_pdi.htm). Esse exemplo é complicado e é possível que existam soluções melhores quando você está agindo diretamente na tabela, porém ele mostra muitas soluções interessantes. O exemplo, porém, é incompleto e um iniciante no Pentaho tem dificuldade de acertar a configuração de todos os passos. No site que acompanha esse livro um exemplo para a tabela Filme está disponível.

Funções interessantes de SQL;

**DATEDIFF** calcula a diferença em dias entre duas datas.