



**Unioeste - Universidade Estadual do Oeste do Paraná**  
**CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS**  
Colegiado de Ciência da Computação  
*Curso de Bacharelado em Ciência da Computação*

## **Análise de Performance entre SGBDs para Ambientes OLAP Utilizando TPC-H**

*Letícia Torres*

**CASCADEL**  
**2017**

**LETÍCIA TORRES**

**ANÁLISE DE PERFORMANCE ENTRE SGBDS PARA AMBIENTES  
OLAP UTILIZANDO TPC-H**

Monografia apresentada como requisito parcial  
para obtenção do grau de Bacharel em Ciência da  
Computação, do Centro de Ciências Exatas e Tec-  
nológicas da Universidade Estadual do Oeste do  
Paraná - Campus de Cascavel

Orientador: Prof. Dr. Marcio Seiji Oyamada

CASCADEL  
2017

**LETÍCIA TORRES**

**ANÁLISE DE PERFORMANCE ENTRE SGBDS PARA AMBIENTES  
OLAP UTILIZANDO TPC-H**

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em  
Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel,  
aprovada pela Comissão formada pelos professores:

---

Prof. Dr. Marcio Seiji Oyamada (Orientador)  
Colegiado de Ciência da Computação,  
UNIOESTE

---

Gustavo Rezende Krüger (Co-orientador)  
Orbit Sistemas

---

Prof. Dr. Clodis Boscarioli  
Colegiado de Ciência da Computação,  
UNIOESTE

Cascavel, 25 de abril de 2018

## **DEDICATÓRIA**

## **AGRADECIMENTOS**

# Lista de Figuras

2.1	Arquitetura de um <i>Data Warehouse</i> . Fonte: adaptado de Kimball e Ross [1] . . .	7
2.2	Exemplo de esquema <i>Star</i> com tabelas Fato e Dimensão . . . . .	8
2.3	Exemplo de esquema <i>Snow Flake</i> adaptado da Figura 2.2 . . . . .	8
2.4	Exemplo de leitura em um SGBD orientado à linha. Fonte: adaptado de Matei e Bank [2] . . . . .	10
2.5	Exemplo de leitura em um SGBD orientado à coluna. Fonte: adaptado de Matei e Bank [2] . . . . .	11
5.1	Esquema do ambiente normalizado . . . . .	20
5.2	Esquema do ambiente desnormalizado . . . . .	22

# Lista de Tabelas

5.1	Número mínimo de sessões para uma classe de banco de dados . . . . .	19
6.1	Cronograma do Trabalho de Conclusão de Curso . . . . .	25

# Lista de Abreviaturas e Siglas

BD	Banco de Dados
TPC	<i>Transaction Processing Performance Council</i>
TPC-H	TPC <i>Benchmark H</i>
DW	<i>Data Warehouse</i>
OLAP	<i>On-Line Analytical Processing</i>
OLTP	<i>On-Line Transaction Processing</i>
SGBD	Sistema Gerenciador de Banco de Dados
SF	Factor Scale
PK	<i>Primary Key</i>
FK	<i>Foreign Key</i>
RF	<i>Refresh Function</i>



# Lista de Símbolos

$Q_i$	Consulta, onde $1 \leq i \leq 22$
$S$	Número de sessões de consulta do Teste de Vazão
$s$	Sessão, onde $1 \leq s \leq S$
$RF_j$	<i>Refresh Function</i> – Função de Atualização
$T_s$	Tempo em segundos da execução de todo o processo do Teste de Vazão

# Sumário

<b>Lista de Figuras</b>	<b>vi</b>
<b>Lista de Tabelas</b>	<b>vii</b>
<b>Lista de Abreviaturas e Siglas</b>	<b>viii</b>
<b>Lista de Símbolos</b>	<b>ix</b>
<b>Sumário</b>	<b>x</b>
<b>Resumo</b>	<b>xii</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Recuperação de Informação</b>	<b>4</b>
2.1 <i>Data Warehouses</i> e Aplicações OLAP . . . . .	4
2.2 Sistema Gerenciador de Banco de Dados . . . . .	9
<b>3 Sistemas Gerenciadores de Banco de Dados</b>	<b>12</b>
3.1 SGBD Relacionais . . . . .	13
3.2 SGBD NoSQL . . . . .	13
<b>4 NoSQL</b>	<b>14</b>
<b>5 TPC <i>Benchmark H</i></b>	<b>15</b>
5.1 Metodologia . . . . .	17
5.1.1 Teste de Força . . . . .	18
5.1.2 Teste de Vazão . . . . .	19
5.2 Ambiente Original Normalizado . . . . .	20
5.3 Ambiente Desnormalizado . . . . .	21
<b>6 Próximos Passos</b>	<b>24</b>
6.1 Cronograma . . . . .	25

<b>A Consultas do Ambiente Original Normalizado</b>	<b>26</b>
<b>B Consultas do Ambiente Desnormalizado</b>	<b>46</b>
<b>Referências Bibliográficas</b>	<b>63</b>

# Resumo

**Palavras-chave: OLAP, TPC-H, SGBD, Data Warehouse, Benchmark**

# Capítulo 1

## Introdução

Vivencia-se atualmente uma economia caracterizada por uma rápida e constante mudança de mercado e oportunidades de negócio, tal que se tornou essencial às empresas a tomada de decisão correta e de forma rápida baseada em alguma decisão de negócio. Essas decisões são tomadas com base na análise de situações passadas e presentes de uma empresa, ou seja, com base nos dados armazenados por ela. Além disso, uma boa decisão também se utiliza da análise de mercado e previsões.

Sob a ótica operacional, de acordo com Wremble e Koncilia [3], os dados de uma empresa são persistidos em sistemas de armazenamento de dados que podem ser heterogêneos, autônomos, e geograficamente distribuídos. Estas características diminuem a eficiência no acesso e processamento dos dados. A gestão de uma empresa requer, no entanto, uma visão abrangente de todos os seus aspectos, exigindo acesso eficiente a todos os dados de interesse. Por este motivo, a capacidade de integrar informações de várias fontes de dados é crucial para uma boa decisão de negócios [3].

Para se obter êxito em uma decisão de negócio é trivial (i) recuperar os dados mantidos por uma empresa; e sobretudo (ii) desenvolver um ambiente de análise cujo objetivo deve ser não apenas informar o significado destes dados, mas sim especular cenários sobre eles com questionamentos como "*e se*" ou "*por quê?*" [4]. Segundo Chaudhuri e Dayal [5] dois elementos são essenciais, dadas as condições anteriores, para uma boa decisão de negócio: *Data Warehouses* (DWs), responsáveis pelo armazenamento homogêneo de dados oriundos de sistemas heterogêneos, e recuperação destes dados; e aplicações OLAP (*On-Line Analytical Processing*).

De acordo com a literatura, existe uma série de princípios que devem ser seguidos ao projetar e implementar um ambiente OLAP dentro de um DW. Dentre estes princípios, destaca-se

a rapidez com que os dados devem ser recuperados e processados no DW. Este princípio é considerado fundamental na maior parte da literatura referente à construção de ambientes OLAP, a exemplo de Codd; Codd e Salley [4], Kimball e Ross [1] e Wremble e Koncilia [3]. Sendo assim, vários aspectos técnicos devem ser considerados sob diferentes pontos de vista. Políticas de *cache* específicas para um servidor de estruturas utilizadas pela aplicação OLAP para armazenar e recuperar dados em memória e o SGBD (Sistema Gerenciador de Banco de Dados) utilizado para o gerenciamento do DW são exemplos de aspectos técnicos que também fazem parte de um ambiente OLAP, e que devem ser considerados. De acordo com Elmasri e Navathe [6] SGBD são importantes para a manutenção e proteção de um banco de dados por um longo período; e também atuam no processo de recuperação de dados de um DW. Neste contexto, a escolha do SGBD no processo de desenvolvimento de um ambiente de análise para organizações com grande quantidade de dados e que utilizam ferramentas OLAP como auxílio na tomada de decisões é importante.

Duas classes de SGBD podem ser utilizadas para realizar o gerenciamento de um DW: SGBD relacionais tradicionais orientados à linha e uma nova abordagem de SGBD NoSQL que são orientados à coluna (SGBD colunares), que fornecem um melhor desempenho à recuperação de dados [7]. Dada a importância na escolha do SGBD e as diferentes soluções apresentadas por cada um, torna-se útil o uso de um benchmark<sup>1</sup> para a realização de uma análise entre SGBD. Existe uma organização sem fins lucrativos fundada com o objetivo de definir padrões para avaliar o desempenho de transações e de bancos de dados com o uso de benchmarks, o TPC (*Transactional Processing Performance Council*) [9]. Esta organização é responsável pela criação de um benchmark voltado para decisões de negócio, o TPC-H [10].

Para refletir a realidade de uma empresa, o TPC-H propõe um ambiente de análise normalizado. Segundo Bax e Souza [11] existe uma discussão sobre a normalização e a denormalização dos dados de um DW. Algumas empresas utilizam um modelo normalizado e têm como justificativa a flexibilidade e a facilidade em situações de manutenção do DW. Por outro lado, existem empresas que utilizam modelos denormalizados e têm como justificativa o ganho de desempenho nas consultas, visto que um modelo denormalizado tende a diminuir o número de tabelas e, por consequência, os *joins* entre tabelas.

---

<sup>1</sup>ferramentas utilizadas para medir e validar o desempenho de alguma tecnologia sob condições de avaliação [8]

O objetivo deste trabalho é a realização de um estudo comparativo entre SGBD, relacionais orientados à linha e orientados à coluna, como gerenciadores de DWs em ambientes OLAP. A comparação será realizada utilizando-se o benchmark TPC-H em sua proposta original, um modelo de banco normalizado; além de uma adaptação para um modelo DW denormalizado. Desta maneira, será possível avaliar os SGBD selecionados de acordo com o modelo do DW, não apenas no contexto original proposto pelo benchmark. Para alcançar tal objetivo, é necessário a execução de uma série de tarefas. A primeira parte engloba o desenvolvimento dos dois ambientes de análise, o modelo padrão proposto pelo TPC-H e o modelo denormalizado, adaptado do TPC-H. Após, são gerados os dados para popular os SGBD e as consultas para as questões de negócio propostas pelo TPC-H; então é feita a população dos dados gerados nos SGBD. A segunda parte consiste na execução das consultas para cada um dos ambientes de análise, sendo que para o ambiente adaptado as consultas devem ser escritas de acordo com as alterações efetuadas. E por fim a aplicação do cálculo de desempenho proposto pelo TPC-H para os dois ambientes.

A organização do documento é da seguinte forma:

- **Capítulo 2:** apresenta detalhes sobre recuperação de informações, explicando os conceitos de *Data Warehouses*, ambientes OLAP e a conexão entre ambos; e SGBD relacionais orientados à linha e à coluna.
- **Capítulo 3:** SGBD
- **Capítulo 4:** NoSQL
- **Capítulo 5:** apresenta detalhes sobre o benchmark TPC-H, abrangendo informações sobre o ambiente normalizado proposto pelo próprio benchmark; bem como sobre o ambiente denormalizado proposto neste trabalho. É também apresentada a metodologia utilizada para medição e análise do desempenho dos SGBD.
- **Capítulo 4:** apresenta o cronograma de desenvolvimento deste trabalho e uma discussão sobre os próximos passos deste desenvolvimento.

## Capítulo 2

# Recuperação de Informação

Um ativo importante em qualquer organização é a informação. Segundo Kimball e Ross [1] essa informação é mantida sob duas formas: sistemas de banco de dados operacionais, onde a informação é armazenada; e DWs, onde ela é recuperada. Em sistemas operacionais geralmente os usuários lidam com o mesmo registro e realizam a mesma tarefa exaustivamente sob uma única informação, permanecendo no domínio das transações; enquanto que em um DW pode-se ver o progresso da organização utilizando dados armazenados continuamente, de forma otimizada para a recuperação de dados. Também, são formuladas perguntas com a finalidade de responder a alguma questão de negócio, como *"quantos pedidos foram recebidos pelo fornecedor X no período de tempo Y?"*, ou *"qual foi o impacto no número de vendas ao mudar o formato de envio de A para B?"*. Para responder questões dessa natureza não é viável lidar com dados de forma individual, mas sim recuperar um conjunto de dados a fim de formular uma resposta.

### 2.1 *Data Warehouses* e Aplicações OLAP

De acordo com Inmon [12], DWs são base de todos os Sistemas de Decisão de Suporte (*Decision Support Systems* – DSS). DSS são tecnologias utilizadas para decisões de negócio e solução de problemas, e incluem componentes que realizam gerenciamento de banco de dados e que permitem uma interação com o usuário de forma a simplificar consultas e geração de relatórios [13]. DWs foram as primeiras ferramentas a surgirem como solução para o suporte à decisão de negócio, integrando dados de diferentes bancos de dados operacionais [12, 1].

De forma geral, um DW é um repositório de dados capaz de fornecer rapidamente informa-



ções consistentes e cruciais para a tomada de decisão de uma organização, de tal forma que essa informação possa ser acessada de maneira intuitiva e legível pelo usuário, a fim de combinar diferentes informações entre os dados armazenados [1]. Deve também se adaptar a possíveis mudanças, sejam elas mudanças comerciais, mudanças nos dados ou na tecnologia. Inmon [12] define um DW como: uma coleção de dados não-volátil, ou seja, que não muda após inserida no *warehouse*; orientado ao assunto principal da organização; integrado e variante no tempo para que seja mantido um histórico a fim de analisar situações passadas. Do ponto de vista estrutural, Wremble e Koncilia [3] definem um DW como uma base de dados homogênea, local e centralizada.

Para analisar os dados de um DW além de implementá-lo é necessário que alguma aplicação leia seu conteúdo e apresente-o de forma gráfica e intuitiva ao analisador. Aplicações OLTP (*On-Line Transactional Processing*) são utilizadas por bancos de dados operacionais e operam transações atômicas e isoladas de forma repetitiva, que correspondem ao dia-a-dia de uma organização [5]. DWs trabalham com suporte à decisão e são intensivos à consultas *ad hoc* complexas, que acessam milhões de registros. Sendo assim, os dados históricos, a taxa de vazão de uma consulta e o tempo de resposta são mais importantes que pequenas transações. À aplicação aceita por um DW dá-se o nome de OLAP, cujo objetivo, de acordo com Codd; Codd e Salley [4], é identificar tendências, padrões de comportamento e anomalias, bem como relações em dados aparentemente não relacionados. Os resultados dessas análises são a base para tomada de decisões de negócio. Portanto, DWs e aplicações OLAP são componentes chave para a construção de um ambiente de análise.

Dentro do domínio de um DW existem diversos componentes que realizam funções específicas a fim de construir um ambiente de *warehouse* desde a obtenção dos dados de fontes externas e sistemas operacionais de bancos de dados, até o acesso a esses dados através do DW por alguma consulta analítica definida sob uma aplicação OLAP. Para entender esta arquitetura fim-a-fim, ilustrada na Figura 2.1, é necessário compreender alguns componentes e conceitos que formam um DW [1]:

- **Sistemas de Fonte Operacional:** possuem detalhes sobre as transações do negócio, correspondendo a ambientes OLTP. Engloba os dados que irão estruturar as informações do DW, portanto se encontram externos ao *warehouse*. Podem ser tanto sistemas de banco

de dados ou alguma outra fonte de dados, como um documento no formato XLS, CSV, TXT; e sistemas CRM.

- **Staging Area:** compreende tanto uma área de armazenamento temporária quanto um conjunto de processos denominado ETL (*extract-transformation-load*). De forma geral é uma área a qual os usuários não têm acesso, onde os dados são traduzidos para algo que possa ser enviado de maneira compatível ao *warehouse* e não se trabalha diretamente sobre os dados transacionais. Quanto aos processos ETL, a fase de Extração (*Extraction*) consiste na leitura da fonte de dados, transferindo o conteúdo necessário para a *staging area*; após essa extração pode ser necessário realizar uma "limpeza" nos dados; unir dados de diferentes fontes; tratar duplicatas e atribuir chaves do *warehouse*. A isto dá-se o nome de Transformação (*Transformation*). A última fase, fase de Carregamento (*Load*), é responsável por carregar, ou popular, os dados na área de estruturação de dados do DW.
- **Estruturação de Dados:** trata de como os dados serão organizados, armazenados e disponibilizados para consultas de usuários, relatórios e outras aplicações. No que tange à comunidade empresarial, a fase de apresentação de dados é o DW em si, pois corresponde ao que pode ser acessado via ferramentas de acesso a dados. A etapa de estruturação é comumente definida como sendo um conjunto de *data marts*. *Data marts* são subconjuntos do total de informações de um DW, cada qual representando os dados de um determinado assunto, departamento, ou processo de negócio. Nesta fase é definida a modelagem conceitual do ambiente de análise do DW.
- **Ferramentas de Acesso aos Dados:** são formas de aplicar uma consulta, dentro de aplicações OLAP, aos dados organizados na fase de estruturação. Pode ser uma consulta *ad hoc* ou algo mais complexo, como consultas aplicadas à mineração de dados.

Ainda não há um consenso acerca da modelagem conceitual da área de apresentação de dados de um ambiente de análise nos DWs [14]. Segundo Sen e Sinha [14] as duas técnicas de modelagem mais utilizadas são a ER (Entidade-Relacional) e a Dimensional. A primeira segue o padrão de modelagem para ambientes OLTP, que traduz a modelagem ER para um esquema relacional em seguida normalizando-o geralmente até a Terceira Forma Normal (3NF) [1]. O modelo Dimensional, ou multidimensional, por sua vez, evita atingir o mesmo nível de

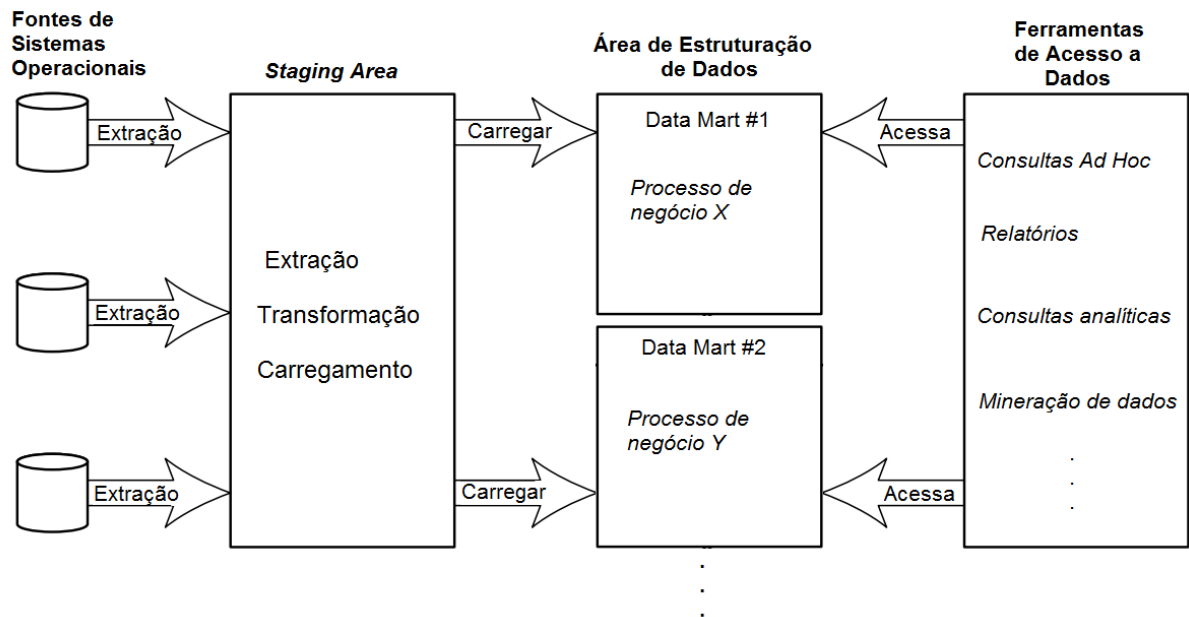


Figura 2.1: Arquitetura de um *Data Warehouse*. Fonte: adaptado de Kimball e Ross [1]

normalização da modelagem ER. Ele é composto por tabelas denominadas Tabelas Fato e Tabelas Dimensão [1], conhecido comumente como modelo *Star Join*, ou apenas *Star* [14]. Uma Tabela Fato é a principal tabela do modelo Dimensional, contemplando atributos responsáveis por determinar as métricas de negócio. Em sua maioria são atributos numéricos, relacionados à *quantidade*; e aditivos, visto que uma consulta em um DW pode retornar até milhares de tuplas, tornando interessante o conhecimento de informações como o total de um atributo dada alguma questão de negócio. As Tabelas Fato são auxiliadas pelas Tabelas Dimensão no que concerne à descrição textual das questões de negócio. É comum estas tabelas terem de 50 a 100 atributos, e que estes atributos sejam responsáveis pelas restrições de uma consulta, sendo também comumente utilizados em agrupamentos. Todas as Tabelas Fato tem duas ou mais chaves estrangeiras (*Foreign Keys – FKs*) relacionadas às chaves primárias (*Primary Keys – PKs*) das Tabelas Dimensão, como mostra o exemplo da Figura 2.2, onde a tabela Vendas corresponde à uma Tabela Fato e as demais à Tabelas Dimensão. Note que esta figura também faz referência a um modelo *Star*.

Mesmo que a modelagem Dimensional não atinja a normalização 3NF, modelos *Star* podem ser trabalhados de forma a oferecer suporte à hierarquia de atributos das Tabelas Dimensão, permitindo que estas tenham "Tabelas Subdimensão". A esse refinamento se dá o nome de

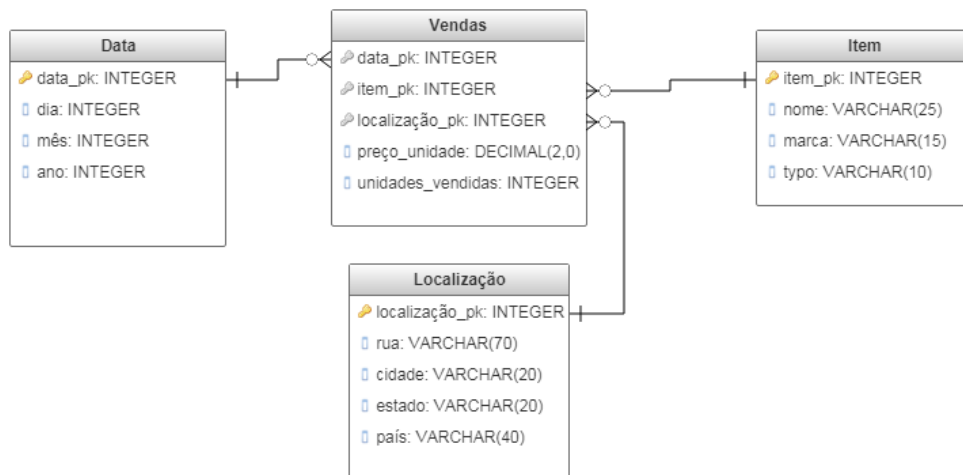


Figura 2.2: Exemplo de esquema *Star* com tabelas Fato e Dimensão

*Snow Flake* [6]. Embora tenham uma estrutura mais simplificada, segundo Levene e Loizou [15] a escolha do uso de esquemas *Snow Flake* se dá por serem um esquema intuitivo, de fácil entendimento, passíveis à otimização de consultas, e de fácil extensão – uma vez que pode-se adicionar atributos às tabelas sem interferir em programas já existentes. Uma possível adaptação de um modelo *Star* para *Snow Flake* é como mostrado na Figura 2.3, no qual foi adaptado o modelo da Figura 2.2, adicionando a Tabela Subdimensão Cidade.

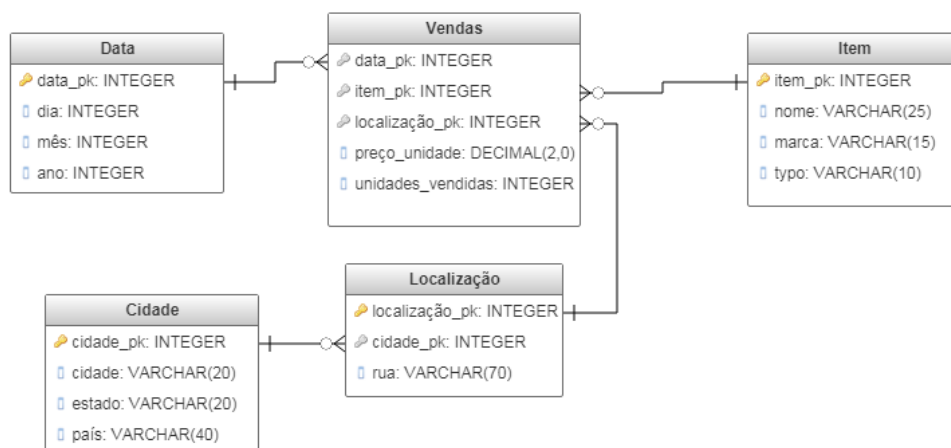


Figura 2.3: Exemplo de esquema *Snow Flake* adaptado da Figura 2.2

Para que possa ser construído um DW e aplicada a modelagem acima descrita, é necessário alguma ferramenta que possa fazer o gerenciamento deste DW. De acordo com Elmasri e Navathe [6] um banco de dados pode ser gerenciado por um sistema que *facilite* este processo

de gerenciamento no banco de dados. Esse sistema é conhecido por o Sistema Gerenciador de Banco de Dados, ou SGBD.

## 2.2 Sistema Gerenciador de Banco de Dados

Elmasri e Navathe [6] citam algumas vantagens do uso de um SGBD como que fazem com que possam ser aplicados como gerenciadores de um DW:

- **Controle de redundância:** dados podem ser adicionados em um DW por diferentes fontes, e são passíveis de duplicação, causando desperdício na capacidade de armazenamento. A verificação de duplicatas pode ser imposta automaticamente no SGBD ao projetar o banco de dados, ou neste caso, o DW.
- **Restrição de acesso:** em uma organização nem todos os funcionários devem ou podem ter acesso ao DW. Este acesso pode ser restringido utilizando as permissões de usuários definidas em um SGBD.
- **Execução de consultas e atualizações de forma eficiente:** SGBDs utilizam estruturas de *índices*, implementadas normalmente utilizando árvores ou *hash* para tornar mais eficiente pesquisas em disco.
- **Restrições de Integridade:** o SGBD é responsável por garantir que (i) tipos de dados não sejam inseridos de forma inconsistente; (ii) relações entre registros sejam efetuadas e (iii) restrições de chave sejam mantidas.

Para realizar o gerenciamento de DWs podem ser utilizados SGBDs relacionais tradicionais orientados à linha, em que todas as informações de uma entidade são mantidas juntas. Porém, de acordo com Matei e Bank [2] o tamanho dos DWs está chegando à casa dos *petabytes*. O maior desafio portanto é garantir o bom desempenho destes *warehouses*, bem como acesso do usuário; aspectos estes que acabam sendo degradados por estruturas orientadas à linha conforme o tamanho do DW aumenta. Ademais, consultas no domínio analítico percorrem todo o banco de dados processando somente os atributos necessários de um grande volume de dados, ao contrário de operações transacionais normalmente executadas por SGBDs orientados à linha,

que percorrem a tupla toda, como mostra a Figura 2.4. Mesmo a adição de índices prejudicaria o desempenho, pois o elevado número de diferentes consultas faz com que seja necessário mais processamento para ler os índices [2]. Para que essa leitura seja aprimorada uma nova abordagem de SGBDs orientados à coluna pode ser aplicada como gerenciador de DWs.

ID	Número da Conta	Tipo de Conta	Agência	Saldo
1				→
2				→
3				→
...				
N				→

Figura 2.4: Exemplo de leitura em um SGBD orientado à linha. Fonte: adaptado de Matei e Bank [2]

Em um SGBD colunar, todas as instâncias de um mesmo atributo são armazenadas juntas. Com isso, em uma leitura, são retornados apenas os atributos requeridos pelo usuário, sem realizar a leitura de uma tupla inteira [16], como mostra a Figura 2.5, o que casa com o modo de leitura de um ambiente analítico. Isso também torna mais eficiente as operações de agrupamento, bastante utilizado em ambientes OLAP, visto que os valores de um mesmo atributo são armazenados consecutivamente. Ainda, a compressão de dados, de acordo com Abadi; Madden e Ferreira [17], é mais eficiente em um SGBD colunar, pois aumenta-se a chance de haver atributos iguais em linhas adjacentes na mesma coluna, visto que são do mesmo tipo. Em um SGBD orientado à linha para realizar compressão tuplas inteiras, com diferentes tipos de atributo, devem coincidir. Atributos com valor *NULL* são tratados mais facilmente em um SGBD orientado à coluna, pois ele pode ser tratado como um valor a ser comprimido.

Exemplos de SGBDs colunares são o (i) Sybase IQ [18], que é utilizado para gerir DWs e otimizado para trabalhar com *Big Data*, possuindo escalabilidade na casa dos *petabytes*; o (ii) MonetDB [19], pioneiro na abordagem colunar; o (iii) Vertica [20], projetado para DWs em nuvem; e o (iv) C-Store [21], desenvolvido por colaboradores do MIT (*Massachusetts Institute of Technology*), Yale, Brandeis University, Brown University e UMass Boston.

ID	Número da Conta	Tipo de Conta	Agência	Saldo
1				
2				
3				
...				
<i>N</i>				

Figura 2.5: Exemplo de leitura em um SGBD orientado à coluna. Fonte: adaptado de Matei e Bank [2]

## Capítulo 3

# Sistemas Gerenciadores de Banco de Dados

Antes de se ter um sistema voltado para gerenciamento de um banco de dados, eram utilizados para persistência de dados o sistema de arquivos. Apesar de simples, esta abordagem apresentava alguns problemas por não apresentar suporte à redundância de informações; não garantir integridade de dados; falta de segurança; e o acesso e gerenciamento dos dados depende de programas e aplicativos, fazendo com que seja necessário criar um novo aplicativo a cada requisição de dados diferente. Outro problema crítico é não se ter informação de relacionamento entre arquivos diferentes.

Como forma de manter o gerenciamento de dados independente de aplicações e programas bem como solucionar os demais falhas do sistema de arquivos foram criados os Sistemas Gerenciadores de Bancos de Dados, os SGBD. De acordo com Elmasri e Navathe [6], os SGBD são uma coleção de programas para criação e manutenção de um banco de dados, que facilita a definição, construção, manipulação e compartilhamento de dados entre usuários e aplicações.

Dentre as vantagens que os SGBD trouxeram em detrimento ao sistema de arquivos estão:

- **Controle de redundância**, para que não seja permitido duplicação de dados, pois isto causaria desperdício na capacidade de armazenamento.
- **Restrição de acesso aos usuários**, pois não será permitida manipulação do banco a todos os usuários, ou funcionários de uma empresa por exemplo.
- **Execução de consultas eficiente** através do uso de índices, normalmente implementadas utilizando hash ou árvores, para que o acesso ao disco seja mais rápido.



- **Restrições de integridade**, a fim de garantir que (i) dados não sejam inseridos de forma inconsistente de acordo com o tipo de atributo definido; (ii) as relações entre entidades sejam efetuadas e (iii) restrições de chave sejam mantidas.
- **Persistência de dados**, para garantir que os dados serão inseridos e de fato armazenados no modelo.
- **Backup** de dados periodicamente para evitar problemas caso aconteça alguma perda no banco e posterior **restauração**, para recuperar uma imagem do último backup feito no banco.

SGBD podem ser classificados de acordo com diferentes critérios,

### 3.1 SGBD Relacionais

### 3.2 SGBD NoSQL

# **Capítulo 4**

## **NoSQL**

## Capítulo 5

### TPC *Benchmark* H

A essência de um *benchmark* é identificar e definir os melhores padrões de excelência para produtos, serviços e/ou processos, e então realizar aperfeiçoamentos de forma que esses padrões sejam alcançados [22]. De maneira geral, *benchmarks* se consolidaram como uma ferramenta para melhorar o desempenho de organizações e a competitividade nos negócios [23].

O TPC *benchmark* H é um padrão de decisão de negócio internacionalmente aceito para comparações de desempenho entre SGBDs utilizados em ambientes OLAP criado pela organização TPC. De acordo com a página oficial do TPC<sup>1</sup>, esta organização sem fins lucrativos foi fundada com o objetivo de definir padrões para *benchmarks* no processamento de transações e bancos de dados. Ela é responsável pelo desenvolvimento e atualização destes *benchmarks*, bem como pela divulgação dos resultados apresentados por eles. Entre os sócios<sup>2</sup> responsáveis por isto estão as empresas Dell, Hewlet Packard, IBM, Microsoft, Oracle, Intel e Cisco. A lista de todos os sócios pode ser acessada na página oficial da organização.

Alguns trabalhos já foram realizados utilizando o TPC-H [24, 25, 26, 27, 28]. Em [24] as consultas do TPC-H são aplicadas no MySQL Cluster a fim de avaliar seu desempenho. O trabalho de [25] tem uma proposta parecida com [24], porém o intuito é avaliar o desempenho de um *cluster* Java EE baseado nas características das consultas do TPC-H. Thanopoulou et al. [26] foca em aplicar o TPC-H em bancos de dados de pequenas empresas que não podem arcar com configurações de *hardware* melhores para administrar um banco de dados. Em [27] é feita uma descrição teórica de dois *benchmarks*, o YCSB (*Yahoo Cloud Serving Benchmark*), voltado para a área de *Big Data*, e o TPC-H. Por fim, em [28] é realizado uma análise comparativa

---

<sup>1</sup><http://www.tpc.org/information/about/abouttpc.asp>

<sup>2</sup>Lista de sócios acessada em Junho de 2017

entre PostgreSQL e MongoDB aplicando o TPC-H. O primeiro é um SGBD relacional clássico, e o outro um SGBD NoSQL com algumas consultas do TPC-H adaptadas. O desempenho do SGBD NoSQL foi inferior ao do relacional. Embora seja feito o uso de uma abordagem NoSQL, o trabalho realizado em [28] é similar ao objetivo apresentado neste.

De acordo com o manual de especificação fornecido pelo TPC [29], o TPC-H é um *benchmark* de suporte à decisão de negócios, constituído de uma série de consultas comerciais *ad-hoc* e modificações simultâneas de dados com finalidade de retratar a realidade das empresas. Ele representa DSS que examinam grandes volumes de dados; executam consultas com um alto grau de complexidade; e respondem questões críticas de negócio. Como o *benchmark* trata de grandes volumes de dados, o tamanho mínimo de banco de dados proposto pelo TPC-H é de 1GB, seguido por 10GB, 30GB, 100GB, 300GB, 1000GB, 3000GB, 10000GB, 30000GB e 100000GB. Estes valores também correspondem ao Fator de Escala (*Factor Scale* – SF).

Com o intuito de avaliar o resultado do desempenho de SGBDs como DW, é necessário ter conhecimento de como os dados que irão popular o DW estão distribuídos. Para tal, o TPC-H propõe um ambiente normalizado *Snow Flake*. Ele é composto por oito tabelas e é descrito na Seção 5.2. Do mesmo modo, a fim de obedecer as regras de modelagem deste ambiente, é preciso ter dados conhecidos para popular os DWs. Estes dados são fornecidos pelo *software* DBGen.

O DBGen foi implementado pelo TPC-H, com o objetivo de realizar a população de dados em um DW seguindo a modelagem original *Snow Flake*. São gerados oito arquivos separados no formato <nome da tabela>.tbl – exemplificando, a tabela de dados *Part* será gerada como *part.tbl*; onde as linhas de cada arquivo representam as tuplas dentro da respectiva tabela, tendo seus atributos separados por um delimitador *pipe* ("|"). Por padrão, os arquivos são gerados para uma base de dados da classe de 1GB, quando nenhum SF é especificado.

Assim como é necessário conhecer os dados de modo a seguir a modelagem proposta pelo *benchmark*, é preciso formular consultas que visam responder às questões de negócio definidas pelo TPC-H sobre o modelo de dados. O próprio TPC define para seu modelo 22 consultas, cada qual definida por (i) uma questão de negócio, que ilustra o contexto no qual a consulta pode ser aplicada; (ii) uma definição funcional, que corresponde à implementação da consulta utilizando a Linguagem SQL-92; (iii) parâmetros de substituição, que geram os valores necessários para

completar os parâmetros da consulta; e (iv) validação, que descreve como validar a consulta no BD. Igualmente à geração de dados, a geração de consultas também é feita com o uso de um programa fornecido pelo TPC, o QGen.

Com o QGen é possível gerar as consultas de acordo com a especificação do TPC-H inserindo os parâmetros de substituição adequados em cada uma. A Consulta 1 presente no Apêndice A, Consulta de Relatório de Resumo de Preços, possui o parâmetro de substituição [DELTA], correspondente a um número inteiro entre 60 e 120, inserido ao executar o programa QGen para a respectiva consulta. Esses valores podem ser inseridos de maneira aleatória, entretanto o TPC-H define valores padrão para os parâmetros de substituição para fins de validação; portanto as consultas no QGen são geradas utilizando os valores padrão. As 22 consultas com suas respectivas questões de negócio e parâmetros de substituição são descritas no Apêndice A.

## 5.1 Metodologia

Após a geração dos arquivos é criado em cada SGBD um esquema correspondente às classes utilizadas, tanto para o Ambiente Normalizado quanto para o Desnormalizado. Cada SGBD é então executado sobre os dois ambientes propostos. Em cada ambiente será executado o teste de performance, que consiste de duas execuções: o Teste de Força (*Power Test*) e o Teste de Vazão (*Throughput Test*), descritos nas Subseções 5.1.1 e 5.1.2, respectivamente. O tempo resultante de cada etapa é utilizado para calcular o desempenho final do SGBD, medido em  $QphH@Size$  – quantidade de consultas executadas por hora, dada uma classe de tamanho de banco de dados. Uma execução é composta por:

- $SF$ , que representa a classe do banco de dados.
- $Q_i$ , que representa uma consulta, onde  $1 \leq i \leq 22$ .
- $S$ , que define o número de sessões de consulta do Teste de Vazão.
- $s$ , que representa uma dada sessão, onde  $1 \leq s \leq S$ .
- $T_s$ , que representa o tempo, em segundos, resultante do processo inteiro.
- $RF_j$ , que representa a Função de Atualização (*Refresh Function* – RF), onde:

- RF1: inserção de novos registros. O número de registros inseridos deve ser igual para o número de registros removidos pela RF2. O pseudocódigo para a RF1 é como:

```

loop (SF * 1500) times
    insert <new row into> ORDERS table
    loop random(1, 7) times
        insert <new row into> LINEITEM table
    end loop
end loop

```

- RF2: remoção de registros antigos. O pseudocódigo para a RF1 é como:

```

loop (SF * 1500) times
    delete from ORDERS where O_ORDERKEY = [valor]
    delete from LINEITEM where I_ORDERKEY = [valor]
end loop

```

O conjunto de dados para executar com sucesso as RFs também são gerados pelo DBGen utilizando a *flag* -U.

### 5.1.1 Teste de Força

O Teste de Força objetiva medir a execução de uma dada consulta do sistema com um único usuário ativo. Neste teste é criada uma única sessão com o respectivo SGBD e as seguintes instruções são executadas:

- Execução da RF1.
- Execução de cada consulta proposta pelo TPC-H, ou adaptada, de forma sequencial uma única vez até a última consulta.
- Execução da RF2.

Será armazenado ao fim do teste o tempo em segundos resultante de cada consulta, bem como o tempo de execução de cada função. Este resultado será utilizado na Equação 5.1.

$$Power@Size = \frac{3600 * SF}{\sqrt[24]{\prod_{i=2}^{i=22} Q(i,0) * \prod_{j=1}^{j=2} RF(j,0)}} \quad (5.1)$$

### 5.1.2 Teste de Vazão

Este teste mede a capacidade do sistema de processar a maior quantidade possível de consultas no menor intervalo de tempo. Aqui o TPC-H exige um número mínimo de sessões de consulta de acordo com a classe de tamanho do banco de dados, como mostra a Tabela 5.1.

Tabela 5.1: Número mínimo de sessões para uma classe de banco de dados

Classe do Banco de Dados	Número de Sessões
1 GB	2
10 GB	3
30 GB	4
100 GB	5
300 GB	6
1000 GB	7
3000 GB	8
10000 GB	9
30000 GB	10
100000 GB	11

É criada uma sessão para executar as Funções e  $N$  sessões para executar as consultas, de acordo com a Tabela 5.1. As instruções são executadas concorrentemente. Para a sessão das Funções, as seguintes instruções são executadas:

- A RF1 e RF2 deverão ser executadas sequencialmente  $N$  vezes, onde  $N$  é o número de sessões para a execução de consultas.
- Para cada sessão de consultas, cada consulta será executada sequencialmente até a última.

Ao fim do teste, é armazenado o tempo em segundos da execução do processo inteiro. O processo inicia quando a primeira sessão, seja de consulta ou de Função, executa sua instrução e finaliza quando a última sessão recebe uma resposta. O resultado é utilizado para calcular o desempenho do SGBD para o Teste de Vazão conforme a Equação 5.2.

$$Throughput@Size = \frac{S*22*3600}{T_s*SF} \quad (5.2)$$

O resultado dos cálculos de Força e Vazão serão utilizados para calcular o desempenho final do SGBD da seguinte forma:

$$QphH@Size = \sqrt{Power@Size * Throughput@Size} \quad (5.3)$$

## 5.2 Ambiente Original Normalizado

O modelo de ambiente proposto pelo TPC-H é um esquema normalizado *Snow Flake*. Ele é composto por oito tabelas, sendo que destas, seis têm o tamanho multiplicado por um Fator de Escala, que corresponde ao tamanho da classe do banco de dados; enquanto que as demais, *NATION* e *REGION*, têm tamanho fixo. O relacionamento *one-to-many* entre as tabelas do esquema *Snow Flake* é apresentado na Figura 5.1, adaptada do manual de especificação do TPC-H [29].

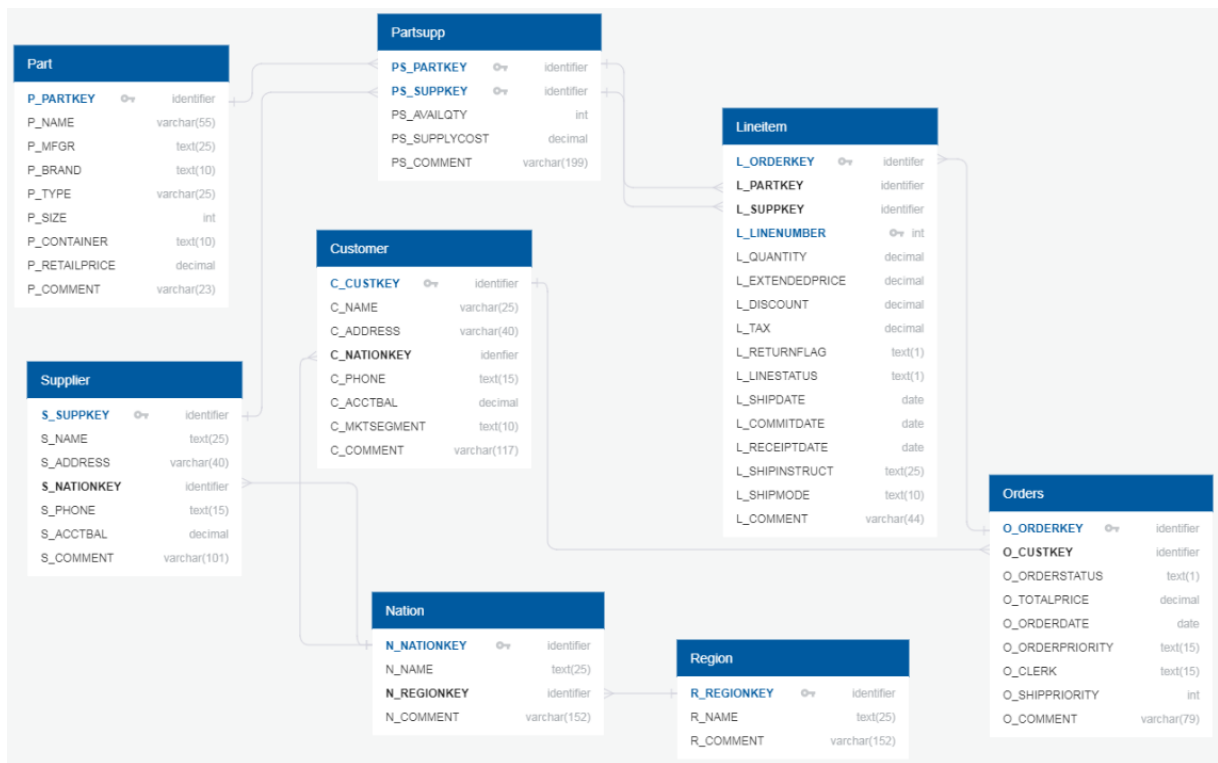


Figura 5.1: Esquema do ambiente normalizado

Os tipos de dados atribuídos aos atributos de uma entidade podem ser:

- Identificador: *identifier* – corresponde a um inteiro que define a PK de uma entidade
- Inteiro: *int*
- Decimal: *decimal*
- Texto fixo de tamanho N: *text(N)*



- Texto variável de tamanho N: varchar(N)
- Data: date

Detalhes sobre as consultas implementadas na Linguagem SQL relacionadas a esse ambiente são encontradas no Apêndice A.

### 5.3 Ambiente Desnormalizado

Embora modelos de dados normalizados sejam eficientes em processos operacionais, SGBDs relacionais não executam uma consulta de maneira eficiente em um esquema normalizado [1]. A otimização é afetada pelo número de *joins*, o que vai contra um dos objetivos de um *warehouse*: recuperação de dados com rapidez.

As modificações no modelo do Ambiente Normalizado se fundamentaram na criação de um modelo parecido com o apresentado pela modelagem *Star*, com uma Tabela Fato central descrita por Tabelas Dimensão. Também procurou-se eliminar quaisquer tabelas cujos atributos sejam frequentemente utilizados em operações de *join*, alguns característicos de um modelo com Tabelas Subdimensionais, e que possam ser incluídos nas tabelas que possuam relacionamento com eles. O esquema final do modelo *Star* é mostrado na Figura 5.2.

A primeira modificação no modelo *Snow Flake* foi a exclusão das entidades Dimensão *Nation* e *Region*. Essas entidades são Tabelas Subdimensionais das entidades *Customer* e *Supplier*, sendo frequentemente requeridas quando há a necessidade de consultar a nação e/ou a região de um cliente e/ou de um fornecedor, assim optou-se por incluir os atributos *N\_NAME*, *N\_COMMENT*, *R\_NAME* e *R\_COMMENT* nas entidades *Supplier* e *Customer*.

Com a intenção de criar uma única Tabela Fato no esquema desnormalizado, observou-se (i) a possibilidade de unir as entidades *Lineitem* e *Orders* em uma única entidade nomeada como *Item*, visto que as duas possuem um relacionamento. Isto também eliminaria alguns *joins* realizados entre as duas entidades nas consultas listadas no Apêndice A. A PK *C\_CUSTKEY* de *Customer* que antes estava em *Orders* como FK é transferida agora para *Item*. Também nota-se que (ii) as entidades *Part* e *Supplier* têm suas PKs como FKs na entidade antes denominada *Lineitem*, provenientes da entidade intermediária *Partsupp*. Optou-se assim por incluir os atributos de *Partsupp* na nova entidade *Item*, excluindo a primeira do esquema e mantendo

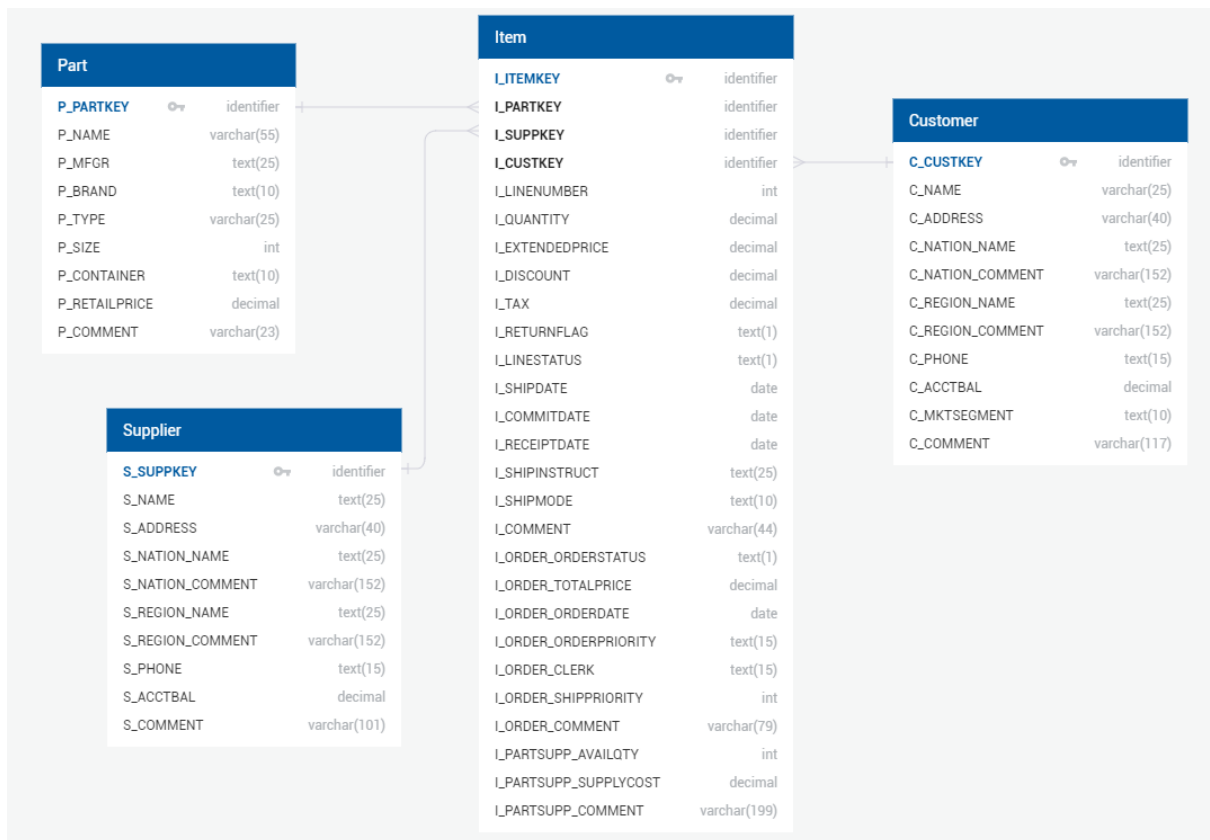


Figura 5.2: Esquema do ambiente desnormalizado

o relacionamento de *Supplier* e *Part* diretamente com *Item*, através das chaves P\_PARTKEY e S\_SUPPKEY.

O esquema final agora possui uma Tabela Fato, *Item*; e três tabelas Dimensão, *Part*, *Customer* e *Supplier*, cada qual com sua chave mantendo o relacionamento com a tabela *Item*. Os dados presentes nas entidades são do mesmo tipo descritos na Seção 5.2. As consultas em Linguagem SQL relacionadas a esse ambiente são encontradas no Apêndice B.

A inserção dos dados do Ambiente Desnormalizado será realizada após todos os dados gerados pelo DBGen do Ambiente Normalizado terem sido populados nos SGBDs. Com isso, é possível realizar *joins* de acordo com as mudanças e o resultado destes será armazenado no esquema correspondente. Por exemplo, ao realizar um *join* entre as entidades *Nation* e *Region* selecionando todos os seus atributos, estes são enviados como entrada para a inserção em uma nova tabela do esquema Desnormalizado correspondente, digamos, *Nation\_Region*. Após, é realizado um novo *join* com as entidades *Supplier* e *Part*, separadamente, para incluir os atributos de *Nation\_Region*: N\_NAME, N\_COMMENT, R\_NAME e R\_COMMENT; bem como os

atributos originais das duas entidades anteriores, nas novas entidades *Supplier* e *Part*.

# Capítulo 6

## Próximos Passos

Neste trabalho foi realizado uma revisão dos conceitos de (i) *Data Warehouse* e ambientes OLAP; (ii) SGBDs relacionais orientados à linha e à coluna e como eles podem atuar como gerenciadores de DWs; e (iii) *benchmark* e TPC-H, bem como a metodologia utilizada por este *benchmark*. Ainda, foram construídos os ambientes nos quais esta metodologia será aplicada, bem como gerados os dados e as consultas. De acordo com o cronograma apresentado na Tabela 6.1, para concluir o objetivo principal deste trabalho é necessária a execução dos testes para análise de desempenho, e para tal deve-se:

1. Definir os SGBDs que serão utilizados para a comparação;
2. Popular tais SGBDs com os dados gerados;
3. Implementar as consultas e RFs utilizando uma Linguagem de Programação (Java ou C#) que ofereça suporte aos *drivers* dos SGBDs;
4. Executar os testes de desempenho de acordo com a metodologia apresentada no Capítulo 5;
5. Apresentar e discutir os resultados utilizando gráficos.

## 6.1 Cronograma

<b>Etapa</b>	<b>Atividade</b>
1	Elaboração e entrega do Projeto de TCC
2	Revisão bibliográfica sobre OLAP, DW, SGBDs colunares, e TPC-H
3	Elaboração dos modelos de DW, consultas e funções ao ambiente adaptado
4	Construção do ambiente de análise
5	Elaboração do texto da prévia
6	<b>Entrega da prévia – 23/08</b>
7	Execução dos testes
8	Elaboração do texto final
9	Entrega do texto final – 20/11
10	Elaboração da versão final
11	Entrega da versão final – 22/11

Tabela 6.1: Cronograma do Trabalho de Conclusão de Curso

	Abril	Maio	Junho	Julho	Agosto	Setembro	Outubro	Novembro	Dezembro
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									

# Apêndice A

## Consultas do Ambiente Original Normalizado

São apresentadas aqui as 22 consultas realizadas no Ambiente *Snow Flake*, cada qual com sua questão de negócio brevemente explicada.

### 1. Consulta de Relatório de Resumo de Preços

Emite um relatório com o resumo de preços de todos os itens enviados em uma certa data.

Esta data corresponde a 60-120 dias antes da maior data de envio contida no banco de dados.

Parâmetro de Substituição	Valor
DELTA	90

```
1  select
2      l_returnflag,
3      l_linestatus,
4      sum(l_quantity) as sum_qty,
5      sum(l_extendedprice) as sum_base_price,
6      sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
7      sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as ←
          sum_charge,
8      avg(l_quantity) as avg_qty,
9      avg(l_extendedprice) as avg_price,
10     avg(l_discount) as avg_disc,
11     count(*) as count_order
12 from
13     lineitem
14 where
15     l_shipdate <= date '1998-12-01' - interval '[DELTA]' day (3)
16 group by
17     l_returnflag,
```

```

18     l_linestatus
19 order by
20     l_returnflag,
21     l_linestatus;

```

## 2. Consulta de Fornecedor de Custo Mínimo

Encontra o fornecedor que pode fornecer um produto com o menor custo em uma dada região.

Parâmetro de Substituição	Valor
SIZE	15
TYPE	BRASS
REGION	EUROPE

```

1  select
2      s_acctbal,
3      s_name,
4      n_name,
5      p_partkey,
6      p_mfgr,
7      s_address,
8      s_phone,
9      s_comment
10 from
11     part,
12     supplier,
13     partsupp,
14     nation,
15     region
16 where
17     p_partkey = ps_partkey
18     and s_suppkey = ps_suppkey
19     and p_size = [SIZE]
20     and p_type like '[TYPE]'
21     and s_nationkey = n_nationkey
22     and n_regionkey = r_regionkey
23     and r_name = '[REGION]'
24     and ps_supplycost = (
25         select
26             min(ps_supplycost)
27         from
28             partsupp,
29             supplier,
30             nation,
31             region
32         where

```

```

33         p_partkey = ps_partkey
34         and s_suppkey = ps_suppkey
35         and s_nationkey = n_nationkey
36         and n_regionkey = r_regionkey
37         and r_name = '[REGION]'
38     )
39 order by
40     s_acctbal desc ,
41     n_name,
42     s_name,
43     p_partkey;
44 set rowcount 100
45 go

```

### 3. Consulta de Prioridade de Envio

Retorna a prioridade de envio dos pedidos com a maior receita entre aqueles que ainda não foram enviados em uma determinada data.

Parâmetro de Substituição	Valor
SEGMENT	BUILDING
DATE	1995-03-15

```

1  select
2      l_orderkey,
3      sum(l_extendedprice * (1 - l_discount)) as revenue,
4      o_orderdate,
5      o_shippriority
6  from
7      customer,
8      orders,
9      lineitem
10 where
11     c_mktsegment = '[SEGMENT]'
12     and c_custkey = o_custkey
13     and l_orderkey = o_orderkey
14     and o_orderdate < date '[DATE]'
15     and l_shipdate > date '[DATE]'
16 group by
17     l_orderkey,
18     o_orderdate,
19     o_shippriority
20 order by
21     revenue desc ,
22     o_orderdate;
23 set rowcount 10
24 go

```



#### 4. Consulta de Verificação de Prioridade de Pedido

Faz a contagem do número de pedidos solicitados em um dado trimestre de um determinado ano no qual pelo menos um item foi recebido pelo cliente após a data de entrega.

Parâmetro de Substituição	Valor
DATE	1993-07-01

```
1  select
2      o_orderpriority,
3      count(*) as order_count
4  from
5      orders
6  where
7      o_orderdate >= date '[DATE]'
8      and o_orderdate < date '[DATE]' + interval '3' month
9      and exists (
10         select
11             *
12         from
13             lineitem
14         where
15             l_orderkey = o_orderkey
16             and l_commitdate < l_receiptdate
17     )
18  group by
19      o_orderpriority
20  order by
21      o_orderpriority;
```

#### 5. Consulta de Volume do Fornecedor Local

Lista para cada nação em uma dada região o volume de receita originado de uma transação na qual cliente e fornecedor eram daquela nação.

Parâmetro de Substituição	Valor
REGION	ASIA
DATE	1994-01-01

```
1  select
2      n_name,
3      sum(l_extendedprice * (1 - l_discount)) as revenue
4  from
5      customer,
6      orders,
7      lineitem,
```

```

8     supplier,
9     nation,
10    region
11 where
12     c_custkey = o_custkey
13     and l_orderkey = o_orderkey
14     and l_suppkey = s_suppkey
15     and c_nationkey = s_nationkey
16     and s_nationkey = n_nationkey
17     and n_regionkey = r_regionkey
18     and r_name = '[REGION]'
19     and o_orderdate >= date '[DATE]'
20     and o_orderdate < date '[DATE]' + interval '1' year
21 group by
22     n_name
23 order by
24     revenue desc;

```

## 6. Previsão de Mudança de Receita

Informa o quanto a receita pode aumentar eliminando alguns descontos em um dado ano.

Parâmetro de Substituição	Valor
DATE	1994-01-01
DISCOUNT	.06
QUANTITY	24

```

1 select
2     sum(l_extendedprice * l_discount) as revenue
3 from
4     lineitem
5 where
6     l_shipdate >= date '[DATE]'
7     and l_shipdate < date '[DATE]' + interval '1' year
8     and l_discount between [DISCOUNT] - 0.01 and [DISCOUNT] + ↵
9         0.01
10    and l_quantity < [QUANTITY];

```

## 7. Consulta de Quantidade de Envio

Determina o valor de bens enviados entre nações para auxiliar na renegociação de contratos de envio.

Parâmetro de Substituição	Valor
NATION1	FRANCE
NATION2	GERMANY

```

1  select
2      supp_nation,
3      cust_nation,
4      l_year,
5      sum(volume) as revenue
6  from
7      (
8          select
9              n1.n_name as supp_nation,
10             n2.n_name as cust_nation,
11             extract(year from l_shipdate) as l_year,
12             l_extendedprice * (1 - l_discount) as volume
13         from
14             supplier,
15             lineitem,
16             orders,
17             customer,
18             nation n1,
19             nation n2
20         where
21             s_suppkey = l_suppkey
22             and o_orderkey = l_orderkey
23             and c_custkey = o_custkey
24             and s_nationkey = n1.n_nationkey
25             and c_nationkey = n2.n_nationkey
26             and (
27                 (n1.n_name = '[NATION1]' and n2.n_name = '[NATION2]↵
28                     ')
29                 or (n1.n_name = '[NATION2]' and n2.n_name = '[↵
30                     NATION1]')
31             )
32             and l_shipdate between date '1995-01-01' and date '↵
33                 1996-12-31'
34         ) as shipping
35     group by
36         supp_nation,
37         cust_nation,
38         l_year
39     order by
40         supp_nation,
41         cust_nation,
42         l_year;

```

## 8. Consulta de Quota de Mercado Nacional

Determina quanto a quota de mercado de uma dada nação em uma dada região mudou em dois anos para um determinado tipo de peça.

Parâmetro de Substituição	Valor
NATION	BRAZIL
REGION	AMERICA
TYPE	ECONOMY ANODIZED STEEL

```

1  select
2      o_year,
3      sum(case
4          when nation = '[NATION]'
5              then volume
6              else 0
7      end) / sum(volume) as mkt_share
8  from
9      (
10         select
11             extract(year from o_orderdate) as o_year,
12             l_extendedprice * (1 - l_discount) as volume,
13             n2.n_name as nation
14         from
15             part,
16             supplier,
17             lineitem,
18             orders,
19             customer,
20             nation n1,
21             nation n2,
22             region
23         where
24             p_partkey = l_partkey
25             and s_suppkey = l_suppkey
26             and l_orderkey = o_orderkey
27             and o_custkey = c_custkey
28             and c_nationkey = n1.n_nationkey
29             and n1.n_regionkey = r_regionkey
30             and r_name = '[REGION]'
31             and s_nationkey = n2.n_nationkey
32             and o_orderdate between date '1995-01-01' and date '↵
33                                     1996-12-31'
34                                     and p_type = '[TYPE]'
35         ) as all_nations
36  group by
37      o_year
38  order by
39      o_year;

```

## 9. Consulta ao Lucro de um Tipo de Produto

Encontra para cada nação em cada ano o lucro de todas as peças pedidas naquele ano contendo uma substring específica em seus nomes que foram preenchidos por um fornecedor naquela nação.

Parâmetro de Substituição	Valor
COLOR	green

```

1  select
2      nation,
3      o_year,
4      sum(amount) as sum_profit
5  from
6      (
7          select
8              n_name as nation,
9              extract(year from o_orderdate) as o_year,
10             l_extendedprice * (1 - l_discount) - ps_supplycost * l_
11             quantity as a
12         from
13             part,
14             supplier,
15             lineitem,
16             partsupp,
17             orders,
18             nation
19         where
20             s_suppkey = l_suppkey
21             and ps_suppkey = l_suppkey
22             and ps_partkey = l_partkey
23             and p_partkey = l_partkey
24             and o_orderkey = l_orderkey
25             and s_nationkey = n_nationkey
26             and p_name like '%[COLOR]%'
27         ) as profit
28  group by
29      nation,
30      o_year
31  order by
32      nation,
33      o_year desc ;

```

## 10. Consulta de Relatório de Itens Retornados

Retorna os 20 principais clientes que podem estar tendo problemas com peças que foram enviadas a eles em um dado trimestre.

Parâmetro de Substituição	Valor
DATE	1993-10-01

```

1  select
2      c_custkey,
3      c_name,
4      sum(l_extendedprice * (1 - l_discount)) as revenue,
5      c_acctbal,
6      n_name,
7      c_address,
8      c_phone,
9      c_comment
10 from
11     customer,
12     orders,
13     lineitem,
14     nation
15 where
16     c_custkey = o_custkey
17     and l_orderkey = o_orderkey
18     and o_orderdate >= date '[DATE]'
19     and o_orderdate < date '[DATE]' + interval '3' month
20     and l_returnflag = 'R'
21     and c_nationkey = n_nationkey
22 group by
23     c_custkey,
24     c_name,
25     c_acctbal,
26     c_phone,
27     n_name,
28     c_address,
29     c_comment
30 order by
31     revenue desc;
32 set rowcount 20
33 go

```

## 11. Consulta a Identificação de Estoques Importantes

Avalia de todos os estoques de fornecedores disponíveis em uma dada nação as peças com uma percentagem significativa do total de peças disponíveis.

Parâmetro de Substituição	Valor
NATION	GERMANY
FRACTION	.0001

```

1  select
2      ps_partkey,
3      sum(ps_supplycost * ps_availqty) as value
4  from
5      partsupp,
6      supplier,
7      nation
8  where
9      ps_suppkey = s_suppkey
10     and s_nationkey = n_nationkey
11     and n_name = '[NATION]'
12  group by
13      ps_partkey having
14          sum(ps_supplycost * ps_availqty) > (
15              select
16                  sum(ps_supplycost * ps_availqty) * [FRACTION]
17              from
18                  partsupp,
19                  supplier,
20                  nation
21              where
22                  ps_suppkey = s_suppkey
23                  and s_nationkey = n_nationkey
24                  and n_name = '[NATION]'
25          )
26  order by
27      value desc;

```

## 12. Consulta a Modos de Envio e Prioridade de Pedidos

Determina quando selecionar modos de envio mais baratos afeta negativamente pedidos com alta prioridade.

Parâmetro de Substituição	Valor
SHIPMODE1	MAIL
SHIPMODE2	SHIP
DATE	1994-01-01

```

1  select
2      l_shipmode,
3      sum(case
4          when o_orderpriority = '1-URGENT'
5              or o_orderpriority = '2-HIGH'
6              then 1
7          else 0
8      end) as high_line_count,
9      sum(case

```

```

10         when o_orderpriority <> '1-URGENT'
11             and o_orderpriority <> '2-HIGH'
12             then 1
13             else 0
14         end) as low_line_count
15 from
16     orders,
17     lineitem
18 where
19     o_orderkey = l_orderkey
20     and l_shipmode in ( '[SHIPMODE1]', '[SHIPMODE2]' )
21     and l_commitdate < l_receiptdate
22     and l_shipdate < l_commitdate
23     and l_receiptdate >= date '[DATE]'
24     and l_receiptdate < date '[DATE]' + interval '1' year
25 group by
26     l_shipmode
27 order by
28     l_shipmode;

```

### 13. Consulta à Distribuição de Clientes

Determina a distribuição de clientes pelo número de pedidos que eles fizeram.

Parâmetro de Substituição	Valor
WORD1	special
WORD2	requests

```

1  select
2      c_count,
3      count(*) as custdist
4  from
5      (
6          select
7              c_custkey,
8              count(o_orderkey)
9          from
10             customer left outer join orders on
11                 c_custkey = o_custkey
12                 and o_comment not like '%[WORD1] %[WORD2]%'
13          group by
14              c_custkey
15      ) as c_orders (c_custkey, c_count)
16 group by
17     c_count
18 order by
19     custdist desc,
20     c_count desc;

```



#### 14. Consulta aos Efeitos de Promoção

Informa o retorno de mercado a uma propaganda, como um comercial de televisão ou uma campanha especial.

Parâmetro de Substituição	Valor
DATE	1995-09-01

```
1 select
2     100.00 * sum(case
3         when p_type like 'PROMO%'
4             then l_extendedprice * (1 - l_discount)
5         else 0
6     end) / sum(l_extendedprice * (1 - l_discount)) as ↵
7     promo_revenue
8 from
9     lineitem,
10    part
11 where
12     l_partkey = p_partkey
13     and l_shipdate >= date '[DATE]'
14     and l_shipdate < date '[DATE]' + interval '1' month;
```

#### 15. Consulta de Fornecedor Principal

Encontra o fornecedor que mais contribuiu com a receita total de peças enviadas durante um dado trimestre de um ano.

Parâmetro de Substituição	Valor
DATE	1996-01-01

```
1 create view revenue[STREAM_ID] (supplier_no, total_revenue) as
2     select
3         l_suppkey,
4         sum(l_extendedprice * (1 - l_discount))
5     from
6         lineitem
7     where
8         l_shipdate >= date '[DATE]'
9         and l_shipdate < date '[DATE]' + interval '3' month
10    group by
11        l_suppkey;
12
13 select
14     s_suppkey,
15     s_name,
```

```

16     s_address,
17     s_phone,
18     total_revenue
19 from
20     supplier,
21     revenue[STREAM_ID]
22 where
23     s_suppkey = supplier_no
24     and total_revenue = (
25         select
26             max(total_revenue)
27         from
28             revenue[STREAM_ID]
29     )
30 order by
31     s_suppkey;
32
33 drop view revenue[STREAM_ID];

```

## 16. Consulta à Relação *Part/Supplier*

Descobre quantos fornecedores podem fornecer peças com determinados atributos requeridos por um cliente.

Parâmetro de Substituição	Valor
BRAND	Brand#45
TYPE	MEDIUM POLISHED
SIZE1	49
SIZE2	14
SIZE3	23
SIZE4	45
SIZE5	19
SIZE6	3
SIZE7	36
SIZE8	9

```

1  select
2      p_brand,
3      p_type,
4      p_size,
5      count(distinct ps_suppkey) as supplier_cnt
6  from
7      partsupp,
8      part
9  where
10     p_partkey = ps_partkey

```

```

11  and p_brand <> '[BRAND] '
12  and p_type not like '[TYPE]%'
13  and p_size in ([SIZE1], [SIZE2], [SIZE3], [SIZE4], [SIZE5], ↵
    [SIZE6], [SIZE7], [SIZE8])
14  and ps_suppkey not in (
15      select
16          s_suppkey
17      from
18          supplier
19      where
20          s_comment like '%Customer%Complaints%'
21  )
22  group by
23      p_brand,
24      p_type,
25      p_size
26  order by
27      supplier_cnt desc ,
28      p_brand,
29      p_type,
30      p_size;

```

## 17. Consulta a Receita de Pedidos de Pequenas Quantidades

Determina quanto de receita seria perdido se não fossem mais feitos pedidos para quantidades pequenas de certas peças.

Parâmetro de Substituição	Valor
BRAND	Brand#23
TYPE	MEDIUM POLISHED
CONTAINER	MED BOX

```

1  select
2      sum(l_extendedprice) / 7.0 as avg_yearly
3  from
4      lineitem,
5      part
6  where
7      p_partkey = l_partkey
8      and p_brand = '[BRAND] '
9      and p_container = '[CONTAINER] '
10     and l_quantity < (
11         select
12             0.2 * avg(l_quantity)
13         from
14             lineitem
15         where

```

```

16         l_partkey = p_partkey
17     );

```

## 18. Consulta a Grandes Volumes de Clientes

Classifica os 100 principais clientes que já realizaram grandes quantidades de pedidos.

Parâmetro de Substituição	Valor
QUANTITY	300

```

1  select
2      c_name,
3      c_custkey,
4      o_orderkey,
5      o_orderdate,
6      o_totalprice,
7      sum(l_quantity)
8  from
9      customer,
10     orders,
11     lineitem
12  where
13      o_orderkey in (
14          select
15              l_orderkey
16          from
17              lineitem
18          group by
19              l_orderkey having
20                  sum(l_quantity) > [QUANTITY]
21      )
22  and c_custkey = o_custkey
23  and o_orderkey = l_orderkey
24  group by
25      c_name,
26      c_custkey,
27      o_orderkey,
28      o_orderdate,
29      o_totalprice
30  order by
31      o_totalprice desc,
32      o_orderdate;

```

## 19. Consulta a Desconto de Receita

Encontra o desconto bruto de todos os pedidos para três diferentes tipos de peças que foram enviadas por via aérea e entregues pessoalmente.

Parâmetro de Substituição	Valor
QUANTITY1	300
BRAND1	Brand#12
QUANTITY2	10
BRAND2	Brand#23
QUANTITY3	20
BRAND3	Brand#34

```

1  select
2      sum(l_extendedprice* (1 - l_discount)) as revenue
3  from
4      lineitem,
5      part
6  where
7      (
8          p_partkey = l_partkey
9          and p_brand = '[BRAND1]'
10         and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM ←
11             PKG')
12         and l_quantity >= [QUANTITY1] and l_quantity <= [←
13             QUANTITY1] + 10
14         and p_size between 1 and 5
15         and l_shipmode in ('AIR', 'AIR REG')
16         and l_shipinstruct = 'DELIVER IN PERSON'
17     )
18     or
19     (
20         p_partkey = l_partkey
21         and p_brand = '[BRAND2]'
22         and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED←
23             PACK')
24         and l_quantity >= [QUANTITY2] and l_quantity <= [←
25             QUANTITY2] + 10
26         and p_size between 1 and 10
27         and l_shipmode in ('AIR', 'AIR REG')
28         and l_shipinstruct = 'DELIVER IN PERSON'
29     )
30     or
31     (
32         p_partkey = l_partkey
33         and p_brand = '[BRAND3]'
34         and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG ←
35             PKG')
36         and l_quantity >= [QUANTITY3] and l_quantity <= [←
37             QUANTITY3] + 10
38         and p_size between 1 and 15
39         and l_shipmode in ('AIR', 'AIR REG')

```

```

34         and l_shipinstruct = 'DELIVER IN PERSON'
35     );

```

## 20. Consulta a Potenciais Promoções de Partes

Identifica os fornecedores de uma nação que possuem uma dada peça em excesso, que podem ser candidatas a uma oferta promocional.

Parâmetro de Substituição	Valor
COLOR	forest
DATE	1994-01-01
NATION	CANADA

```

1  select
2      s_name,
3      s_address
4  from
5      supplier,
6      nation
7  where
8      s_suppkey in (
9          select
10             ps_suppkey
11         from
12             partsupp
13         where
14             ps_partkey in (
15                 select
16                     p_partkey
17                 from
18                     part
19                 where
20                     p_name like '[COLOR]%'
21             )
22         and ps_availqty > (
23             select
24                 0.5 * sum(l_quantity)
25             from
26                 lineitem
27             where
28                 l_partkey = ps_partkey
29                 and l_suppkey = ps_suppkey
30                 and l_shipdate >= date '[DATE]'
31                 and l_shipdate < date '[DATE]' + interval '1' year
32         )
33     )
34     and s_nationkey = n_nationkey

```

```

35     and n_name = '[NATION]'
36 order by
37     s_name;

```

## 21. Consulta a Fornecedores que Mantiveram Pedidos em Espera

Identifica fornecedores que não puderam enviar peças pedidas em um tempo hábil.

Parâmetro de Substituição	Valor
NATION	SAUDI ARABIA

```

1  select
2      s_name,
3      count(*) as numwait
4  from
5      supplier,
6      lineitem l1,
7      orders,
8      nation
9  where
10     s_suppkey = l1.l_suppkey
11     and o_orderkey = l1.l_orderkey
12     and o_orderstatus = 'F'
13     and l1.l_receiptdate > l1.l_commitdate
14     and exists (
15         select
16             *
17         from
18             lineitem l2
19         where
20             l2.l_orderkey = l1.l_orderkey
21             and l2.l_suppkey <> l1.l_suppkey
22     )
23     and not exists (
24         select
25             *
26         from
27             lineitem l3
28         where
29             l3.l_orderkey = l1.l_orderkey
30             and l3.l_suppkey <> l1.l_suppkey
31             and l3.l_receiptdate > l3.l_commitdate
32     )
33     and s_nationkey = n_nationkey
34     and n_name = '[NATION]'
35 group by
36     s_name
37 order by

```

```

38     numwait desc ,
39     s_name;
40 set rowcount 100
41 go

```

## 22. Consulta a Oportunidades de Vendas Globais

Conta quantos clientes de determinados países não realizaram pedidos durante sete anos, porém têm chances de realizar um pedido.

Parâmetro de Substituição	Valor
I1	13
I2	31
I3	23
I4	29
I5	30
I6	18
I7	17

```

1  select
2      cntrycode,
3      count(*) as numcust,
4      sum(c_acctbal) as totacctbal
5  from
6      (
7          select
8              substring(c_phone from 1 for 2) as cntrycode,
9              c_acctbal
10         from
11             customer
12         where
13             substring(c_phone from 1 for 2) in
14                 ('[I1]', '[I2]', '[I3]', '[I4]', '[I5]', '[I6]', '[I7]')
15         and c_acctbal > (
16             select
17                 avg(c_acctbal)
18             from
19                 customer
20             where
21                 c_acctbal > 0.00
22                 and substring(c_phone from 1 for 2) in
23                     ('[I1]', '[I2]', '[I3]', '[I4]', '[I5]', '[I6]
24                     ', '[I7]')
25         )
26         and not exists (
27             select

```



```
27             *
28         from
29             orders
30         where
31             o_custkey = c_custkey
32     )
33 ) as custsale
34 group by
35     cntrycode
36 order by
37     cntrycode;
```

# Apêndice B

## Consultas do Ambiente Desnormalizado

São apresentadas aqui as 22 consultas realizadas no Ambiente Desnormalizado adaptado da proposta original do TPC-H. As questões de negócio são as mesmas que as apresentadas no Apêndice A.

### 1. Consulta de Relatório de Resumo de Preços

Parâmetro de Substituição	Valor
DELTA	90

```
1  select
2      i_returnflag,
3      i_linestatus,
4      sum(i_quantity) as sum_qty,
5      sum(i_extendedprice) as sum_base_price,
6      sum(i_extendedprice * (1 - i_discount)) as sum_disc_price,
7      sum(i_extendedprice * (1 - i_discount) * (1 + i_tax)) as ↵
          sum_charge,
8      avg(i_quantity) as avg_qty,
9      avg(i_extendedprice) as avg_price,
10     avg(i_discount) as avg_disc,
11     count(*) as count_order
12 from
13     item
14 where
15     i_shipdate <= date '1998-12-01' - interval '[DELTA]' day ↵
          (3)
16 group by
17     i_returnflag,
18     i_linestatus
19 order by
20     i_returnflag,
21     i_linestatus;
```

## 2. Consulta de Fornecedor de Custo Mínimo

Parâmetro de Substituição	Valor
SIZE	15
TYPE	BRASS
REGION	EUROPE

```
1  select
2      s_acctbal,
3      s_name,
4      s_nation_name,
5      p_partkey,
6      p_mfgr,
7      s_address,
8      s_phone,
9      s_comment
10 from
11     part,
12     supplier
13 where
14     p_partkey = i_partkey
15     and s_suppkey = i_suppkey
16     and p_size = [SIZE]
17     and p_type like '[TYPE] '
18     and s_region_name = '[REGION] '
19     and i_partsupp_supplycost = (
20         select
21             min(i_partsupp_supplycost)
22         from
23             item,
24             supplier
25         where
26             p_partkey = i_partkey
27             and s_suppkey = i_suppkey
28             and s_region_name = '[REGION] '
29     )
30 order by
31     s_acctbal desc,
32     s_nation_name,
33     s_name,
34     p_partkey;
35 set rowcount 100
36 go
```

## 3. Consulta de Prioridade de Envio

Parâmetro de Substituição	Valor
SEGMENT	BUILDING
DATE	1995-03-15

```

1  select
2      i_itemkey,
3      sum(i_extendedprice * (1 - i_discount)) as revenue,
4      i_order_orderdate,
5      i_order_shippriority
6  from
7      customer,
8      item
9  where
10     c_mktsegment = '[SEGMENT]'
11     and c_custkey = i_custkey
12     and i_order_orderdate < date '[DATE]'
13     and i_shipdate > date '[DATE]'
14  group by
15     i_itemkey,
16     i_order_orderdate,
17     i_order_shippriority
18  order by
19     revenue desc,
20     o_orderdate
21  set rowcount 10
22  go

```

#### 4. Consulta de Verificação de Prioridade de Pedido

Parâmetro de Substituição	Valor
DATE	1993-07-01

```

1  select
2      i_order_orderpriority,
3      count(*) as order_count
4  from
5      item
6  where
7      i_order_orderdate >= date '[DATE]'
8      and i_order_orderdate < date '[DATE]' + interval '3' month
9      and exists (
10         select
11             *
12         from
13             item
14         where
15             i_commitdate < i_receiptdate
16     )
17  group by
18     i_order_orderpriority
19  order by
20     i_order_orderpriority;

```

## 5. Consulta de Volume do Fornecedor Local

Parâmetro de Substituição	Valor
REGION	ASIA
DATE	1994-01-01

```
1 select
2     s_nation_name,
3     sum(i_extendedprice * (1 - i_discount)) as revenue
4 from
5     customer,
6     item
7     supplier,
8 where
9     c_custkey = i_custkey
10    and i_suppkey = s_suppkey
11    and c_nation_name = s_nation_name
12    and s_region_name = '[REGION]'
13    and i_order_orderdate >= date '[DATE]'
14    and i_order_orderdate < date '[DATE]' + interval '1' year
15 group by
16     s_nation_name
17 order by
18     revenue desc;
```

## 6. Previsão de Mudança de Receita

Parâmetro de Substituição	Valor
DATE	1994-01-01
DISCOUNT	.06
QUANTITY	24

```
1 select
2     sum(i_extendedprice * i_discount) as revenue
3 from
4     item
5 where
6     i_shipdate >= date '[DATE]'
7     and i_shipdate < date '[DATE]' + interval '1' year
8     and i_discount between [DISCOUNT] - 0.01 and [DISCOUNT] + ↵
9         0.01
10    and i_quantity < [QUANTITY];
```

## 7. Consulta de Quantidade de Envio

Parâmetro de Substituição	Valor
NATION1	FRANCE
NATION2	GERMANY

```

1  select
2      supp_nation,
3      cust_nation,
4      i_year,
5      sum(volume) as revenue
6  from
7      (
8          select
9              s_nation_name as supp_nation,
10             c_nation_name as cust_nation,
11             extract(year from i_shipdate) as i_year,
12             i_extendedprice * (1 - i_discount) as volume
13         from
14             supplier,
15             item,
16             customer,
17         where
18             s_suppkey = i_suppkey
19             and c_custkey = i_custkey
20             and (
21                 (s_nation_name = '[NATION1]' and c_nation_name = '[NATION2]')
22                 or (s_nation_name = '[NATION2]' and c_nation_name = '[NATION1]')
23             )
24             and i_shipdate between date '1995-01-01' and date '1996-12-31'
25     ) as shipping
26  group by
27      supp_nation,
28      cust_nation,
29      i_year
30  order by
31      supp_nation,
32      cust_nation,
33      i_year;

```

## 8. Consulta de Quota de Mercado Nacional

Parâmetro de Substituição	Valor
NATION	BRAZIL
REGION	AMERICA
TYPE	ECONOMY ANODIZED STEEL

```

1  select
2      order_year,
3      sum(case
4          when nation = '[NATION]' then volume

```

```

5         else 0
6     end) / sum(volume) as mkt_share
7 from
8     (
9         select
10            extract(year from i_order_orderdate) as order_year,
11            i_extendedprice * (1 - i_discount) as volume,
12            s_nation_name as nation
13        from
14            part,
15            supplier,
16            item,
17            customer
18        where
19            p_partkey = i_partkey
20            and s_suppkey = i_suppkey
21            and i_custkey = c_custkey
22            and c_region_name = '[REGION]'
23            and i_order_orderdate between date '1995-01-01' and ←
24                date '1996-12-31'
25            and p_type = '[TYPE]'
26        ) as all_nations
27 group by
28     order_year
29 order by
30     order_year;

```

## 9. Consulta ao Lucro de um Tipo de Produto

Parâmetro de Substituição	Valor
COLOR	green

```

1 select
2     nation,
3     order_year,
4     sum(amount) as sum_profit
5 from
6     (
7         select
8            s_nation_name as nation,
9            extract(year from i_order_orderdate) as order_year,
10            i_extendedprice * (1 - i_discount) - ←
11                i_partsupp_supplycost * i_quantity as amount
12        from
13            part,
14            supplier,
15            item
16        where

```

```

16         s_suppkey = i_suppkey
17         and p_partkey = i_partkey
18         and p_name like '%[COLOR]%'
19     ) as profit
20 group by
21     nation,
22     order_year
23 order by
24     nation,
25     order_year desc;

```

## 10. Consulta de Relatório de Itens Retornados

Parâmetro de Substituição	Valor
DATE	1993-10-01

```

1  select
2      c_custkey,
3      c_name,
4      sum(i_extendedprice * (1 - i_discount)) as revenue,
5      c_acctbal,
6      c_nation_name,
7      c_address,
8      c_phone,
9      c_comment
10 from
11     customer,
12     item
13 where
14     c_custkey = i_custkey
15     and i_order_orderdate >= date '[DATE]'
16     and i_order_orderdate < date '[DATE]' + interval '3' month
17     and i_returnflag = 'R'
18 group by
19     c_custkey,
20     c_name,
21     c_acctbal,
22     c_phone,
23     c_nation_name,
24     c_address,
25     c_comment
26 order by
27     revenue desc;
28
29 set rowcount 20
30 go

```



## 11. Consulta a Identificação de Estoques Importantes

Parâmetro de Substituição	Valor
NATION	GERMANY
FRACTION	.0001

```
1  select
2      i_partkey,
3      sum(i_partsupp_supplycost * i_partsupp_availqty) as value
4  from
5      item,
6      supplier
7  where
8      i_suppkey = s_suppkey
9      and s_nation_name = '[NATION] '
10 group by
11     i_partkey having
12     sum(i_partsupp_supplycost * i_partsupp_availqty) > (
13         select
14             sum(i_partsupp_supplycost * i_partsupp_availqty) * ←
15             [FRACTION]
16         from
17             item,
18             supplier
19         where
20             i_suppkey = s_suppkey
21             and s_nation_name = '[NATION] '
22     )
23 order by
24     value desc;
```

## 12. Consulta a Modos de Envio e Prioridade de Pedidos

Parâmetro de Substituição	Valor
SHIPMODE1	MAIL
SHIPMODE2	SHIP
DATE	1994-01-01

```
1  select
2      i_shipmode,
3      sum(case
4          when i_order_orderpriority = '1-URGENT'
5          or i_order_orderpriority = '2-HIGH'
6          then 1
7          else 0
8      end) as high_line_count,
9      sum(case
```

```

10         when i_order_orderpriority <> '1-URGENT'
11             and i_order_orderpriority <> '2-HIGH'
12             then 1
13             else 0
14         end) as low_line_count
15 from
16     item
17 where
18     and i_shipmode in ('[SHIPMODE1]', '[SHIPMODE2]')
19     and i_commitdate < i_receiptdate
20     and i_shipdate < i_commitdate
21     and i_receiptdate >= date '[DATE]'
22     and i_receiptdate < date '[DATE]' + interval '1' year
23 group by
24     i_shipmode
25 order by
26     i_shipmode;

```

### 13. Consulta à Distribuição de Clientes

Parâmetro de Substituição	Valor
WORD1	special
WORD2	requests

```

1  select
2      c_count,
3      count(*) as custdist
4  from
5      (
6          select
7              c_custkey,
8              count(i_itemkey)
9          from
10             customer left outer join item on
11                 c_custkey = i_custkey
12                 and i_order_comment not like '%[WORD1]%' [WORD2]%'↵
13             group by
14                 c_custkey
15         ) as c_orders (c_custkey, c_count)
16 group by
17     c_count
18 order by
19     custdist desc,
20     c_count desc;

```

### 14. Consulta aos Efeitos de Promoção

Parâmetro de Substituição	Valor
DATE	1995-09-01

```

1 select
2     100.00 * sum(case
3         when p_type like 'PROMO%'
4             then i_extendedprice * (1 - i_discount)
5         else 0
6     end) / sum(i_extendedprice * (1 - i_discount)) as ←
        promo_revenue
7 from
8     item,
9     part
10 where
11     i_partkey = p_partkey
12     and i_shipdate >= date '[DATE]'
13     and i_shipdate < date '[DATE]' + interval '1' month;

```

#### 15. Consulta de Fornecedor Principal

Parâmetro de Substituição	Valor
DATE	1996-01-01

```

1 create view revenue[STREAM_ID] (supplier_no, total_revenue) as
2     select
3         i_suppkey,
4         sum(i_extendedprice * (1 - i_discount))
5     from
6         item
7     where
8         i_shipdate >= date '[DATE]'
9         and i_shipdate < date '[DATE]' + interval '3' month
10    group by
11        i_suppkey;
12
13 select
14     s_suppkey,
15     s_name,
16     s_address,
17     s_phone,
18     total_revenue
19 from
20     supplier,
21     revenue[STREAM_ID]
22 where
23     s_suppkey = supplier_no
24     and total_revenue = (

```

```

25         select
26             max(total_revenue)
27         from
28             revenue[STREAM_ID]
29     )
30 order by
31     s_suppkey;
32
33 drop view revenue[STREAM_ID];

```

#### 16. Consulta à Relação *Part/Supplier*

Parâmetro de Substituição	Valor
BRAND	Brand#45
TYPE	MEDIUM POLISHED
SIZE1	49
SIZE2	14
SIZE3	23
SIZE4	45
SIZE5	19
SIZE6	3
SIZE7	36
SIZE8	9

```

1  select
2      p_brand,
3      p_type,
4      p_size,
5      count(distinct i_suppkey) as supplier_cnt
6  from
7      item,
8      part
9  where
10     p_partkey = i_partkey
11     and p_brand <> '[BRAND] '
12     and p_type not like '[TYPE]%'
13     and p_size in ([SIZE1], [SIZE2], [SIZE3], [SIZE4], [SIZE5], ↵
14                   [SIZE6], [SIZE7], [SIZE8])
15     and i_suppkey not in (
16         select
17             s_suppkey
18         from
19             supplier
20         where
21             s_comment like '%Customer%Complaints%'
22     )
23 group by

```

```

23     p_brand,
24     p_type,
25     p_size
26 order by
27     supplier_cnt desc ,
28     p_brand,
29     p_type,
30     p_size;
31 go

```

#### 17. Consulta a Receita de Pedidos de Pequenas Quantidades

Parâmetro de Substituição	Valor
BRAND	Brand#23
TYPE	MEDIUM POLISHED
CONTAINER	MED BOX

```

1  select
2      sum(i_extendedprice) / 7.0 as avg_yearly
3  from
4      item,
5      part
6  where
7      p_partkey = i_partkey
8      and p_brand = '[BRAND] '
9      and p_container = '[CONTAINER] '
10     and i_quantity < (
11         select
12             0.2 * avg(i_quantity)
13         from
14             item
15         where
16             i_partkey = p_partkey
17     );

```

#### 18. Consulta a Grandes Volumes de Clientes

Parâmetro de Substituição	Valor
QUANTITY	300

```

1  select
2      c_name,
3      c_custkey,
4      i_itemkey,
5      i_order_orderdate,
6      i_order_totalprice,
7      sum(i_quantity)

```

```

8  from
9      customer,
10     item
11  where
12      i_itemkey in (
13          select
14              i_itemkey
15          from
16              item
17          group by
18              i_itemkey having
19                  sum(i_quantity) > [QUANTITY]
20      )
21  and c_custkey = i_custkey
22  group by
23      c_name,
24      c_custkey,
25      i_itemkey,
26      i_order_orderdate,
27      i_order_totalprice
28  order by
29      i_order_totalprice desc ,
30      i_order_orderdate;
31  set rowcount 100
32  go

```

#### 19. Consulta a Desconto de Receita

Parâmetro de Substituição	Valor
QUANTITY1	300
BRAND1	Brand#12
QUANTITY2	10
BRAND2	Brand#23
QUANTITY3	20
BRAND3	Brand#34

```

1  select
2      sum(i_extendedprice* (1 - i_discount)) as revenue
3  from
4      item,
5      part
6  where
7      (
8          p_partkey = i_partkey
9          and p_brand = '[BRAND1]'
10         and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM←
                                PKG')

```

```

11         and i_quantity >= [QUANTITY1] and i_quantity <= [↵
           QUANTITY1] + 10
12         and p_size between 1 and 5
13         and i_shipmode in ('AIR', 'AIR REG')
14         and i_shipinstruct = 'DELIVER IN PERSON'
15     )
16     or
17     (
18         p_partkey = i_partkey
19         and p_brand = '[BRAND2]'
20         and p_container in ('MED BAG', 'MED BOX', 'MED PKG', '↵
           MED PACK')
21         and i_quantity >= [QUANTITY2] and i_quantity <= [↵
           QUANTITY2] + 10
22         and p_size between 1 and 10
23         and i_shipmode in ('AIR', 'AIR REG')
24         and i_shipinstruct = 'DELIVER IN PERSON'
25     )
26     or
27     (
28         p_partkey = i_partkey
29         and p_brand = '[BRAND3]'
30         and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG↵
           PKG')
31         and i_quantity >= [QUANTITY3] and i_quantity <= [↵
           QUANTITY3] + 10
32         and p_size between 1 and 15
33         and i_shipmode in ('AIR', 'AIR REG')
34         and i_shipinstruct = 'DELIVER IN PERSON'
35     );

```

## 20. Consulta a Potenciais Promoções de Partes

Parâmetro de Substituição	Valor
COLOR	forest
DATE	1994-01-01
NATION	CANADA

```

1  select
2      s_name,
3      s_address
4  from
5      supplier
6  where
7      s_suppkey in (
8          select
9              i_suppkey
10         from

```

```

11         item
12     where
13         i_partkey in (
14             select
15                 p_partkey
16             from
17                 part
18             where
19                 p_name like '[COLOR]%'
20         )
21     and i_partsupp_availqty > (
22         select
23             0.5 * sum(i_quantity)
24         from
25             item
26         where
27             and i_shipdate >= date '[DATE]'
28             and i_shipdate < date '[DATE]' + interval '↵
29                 1' year
30     )
31     and s_nation_name = '[NATION]'
32 order by
33     s_name;

```

## 21. Consulta a Fornecedores que Mantiveram Pedidos em Espera

Parâmetro de Substituição	Valor
NATION	SAUDI ARABIA

```

1  select
2      s_name,
3      count(*) as numwait
4  from
5      supplier,
6      item i1
7  where
8      s_suppkey = i1.i_suppkey
9      and i1.i_order_orderstatus = 'F'
10     and i1.i_receiptdate > i1.i_commitdate
11     and exists (
12         select
13             *
14         from
15             item i2
16         where
17             i2.i_itemkey = i1.i_itemkey
18         and i2.i_suppkey <> i1.i_suppkey

```



```

19     )
20     and not exists (
21         select
22             *
23         from
24             item i3
25         where
26             i3.i_itemkey = i1.i_itemkey
27             and i3.i_suppkey <> i1.i_suppkey
28             and i3.i_receiptdate > i3.i_commitdate
29     )
30     and s_nation_name = '[NATION]'
31 group by
32     s_name
33 order by
34     numwait desc ,
35     s_name;
36 set rowcount 100
37 go

```

## 22. Consulta a Oportunidades de Vendas Globais

Parâmetro de Substituição	Valor
I1	13
I2	31
I3	23
I4	29
I5	30
I6	18
I7	17

```

1  select
2      cntrycode,
3      count(*) as numcust,
4      sum(c_acctbal) as totacctbal
5  from
6      (
7          select
8              substring(c_phone from 1 for 2) as cntrycode,
9              c_acctbal
10         from
11             customer
12         where
13             substring(c_phone from 1 for 2) in
14                 ('[I1]', '[I2]', '[I3]', '[I4]', '[I5]', '[I6]'↵
15                 , '[I7]')
16         and c_acctbal > (

```

```

16         select
17             avg(c_acctbal)
18         from
19             customer
20         where
21             c_acctbal > 0.00
22             and substring(c_phone from 1 for 2) in
23                 ('[I1]', '[I2]', '[I3]', '[I4]', '[I5]', '[I6]', '[I7]')
24     )
25     and not exists (
26         select
27             *
28         from
29             item
30         where
31             i_custkey = c_custkey
32     )
33 ) as custsale
34 group by
35     cntrycode
36 order by
37     cntrycode;

```

# Referências Bibliográficas

- [1] KIMBALL, R.; ROSS, M. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. 2. ed. [S.l.]: Wiley Computer Publishing, 2002.
- [2] MATEI, G.; BANK, R. C. Column-oriented databases, an alternative for analytical environment. *Database Systems Journal*, Academy of Economic Studies-Bucharest, Romania, v. 1, n. 2, p. 3–16, 2010.
- [3] WREMBEL, R.; KONCILIA, C. *Data warehouses and OLAP: concepts, architectures, and solutions*. [S.l.]: Igi Global, 2007.
- [4] CODD, E.; CODD, S.; SALLEY, C. Providing olap (on-line analytical processing) to user-analysis: An it mandate. *White paper*, 1993.
- [5] CHAUDHURI, S.; DAYAL, U. An overview of data warehousing and olap technology. *ACM Sigmod record*, ACM, v. 26, n. 1, p. 65–74, 1997.
- [6] ELMASRI, R.; NAVATHE, S. B. *Sistemas de Banco de Dados*. 6. ed. [S.l.]: Pearson Education do Brasil, 2011.
- [7] Good Data Help. *Column Storage and Compression in Data Warehouse*. 2017. Acessado em: 12 de maio de 2017. Disponível em: <<https://goo.gl/f1yeh8>>.
- [8] BOUCKAERT, S. et al. Benchmarking computers and computer networks. *EU FIRE White Paper*, 2010.
- [9] TRANSACTION PROCESSING PERFORMANCE COUNCIL. *TPC Homepage*. 2017. Acessado em: 12 de agosto de 2017. Disponível em: <<http://www.tpc.org>>.

- [10] TRANSACTION PROCESSING PERFORMANCE COUNCIL. *TPC-H Homepage*. 2017. Acessado em: 19 de maio de 2017. Disponível em: <<http://www.tpc.org/tpch/>>.
- [11] BAX, M. P.; SOUZA, R. Modelagem estratégica de dados: Normalização versus "dimensionalização". *KMBRASIL, Anais... São Paulo: SBGC*, 2003.
- [12] INMON, W. H. *Building the data warehouse*. [S.l.]: John wiley & sons, 2005.
- [13] SHIM, J. P. et al. Past, present, and future of decision support technology. *Decision support systems*, Elsevier, v. 33, n. 2, p. 111–126, 2002.
- [14] SEN, A.; SINHA, A. P. A comparison of data warehousing methodologies. *Communications of the ACM*, ACM, v. 48, n. 3, p. 79–84, 2005.
- [15] LEVENE, M.; LOIZOU, G. Why is the snowflake schema a good data warehouse design? *Information Systems*, Elsevier, v. 28, n. 3, p. 225–240, 2003.
- [16] KHOSHAFIAN, S. et al. A query processing strategy for the decomposed storage model. In: IEEE. *Data Engineering, 1987 IEEE Third International Conference on*. [S.l.], 1987. p. 636–643.
- [17] ABADI, D.; MADDEN, S.; FERREIRA, M. Integrating compression and execution in column-oriented database systems. In: ACM. *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. [S.l.], 2006. p. 671–682.
- [18] MACNICOL, R.; FRENCH, B. Sybase iq multiplex-designed for analytics. In: VLDB ENDOWMENT. *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. [S.l.], 2004. p. 1227–1230.
- [19] MONETDB – The column-store pioneer. 2017. Acessado em: 20 de agosto de 2017. Disponível em: <<https://www.monetdb.org/Home>>.
- [20] VERTICA. 2017. Acessado em: 20 de agosto de 2017. Disponível em: <<https://www.vertica.com/>>.

- [21] STONEBRAKER, M. et al. C-store: a column-oriented dbms. In: VLDB ENDOWMENT. *Proceedings of the 31st international conference on Very large data bases*. [S.l.], 2005. p. 553–564.
- [22] BHUTTA, K. S.; HUQ, F. Benchmarking–best practices: an integrated approach. *Benchmarking: An International Journal*, MCB UP Ltd, v. 6, n. 3, p. 254–268, 1999.
- [23] KYRÖ, P. Revising the concept and forms of benchmarking. *Benchmarking: An International Journal*, MCB UP Ltd, v. 10, n. 3, p. 210–225, 2003.
- [24] NGAMSURIYAROJ, S.; PORNPATTANA, R. Performance evaluation of tpc-h queries on mysql cluster. In: IEEE. *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on*. [S.l.], 2010. p. 1035–1040.
- [25] NADEE, W.; NGAMSURIYAROJ, S. Performance evaluation of tpc-h queries on java ee cluster. In: IEEE. *Computer Science and Software Engineering (JCSSE), 2012 International Joint Conference on*. [S.l.], 2012. p. 385–390.
- [26] THANOPOULOU, A.; CARREIRA, P.; GALHARDAS, H. Benchmarking with tpc-h on off-the-shelf hardware. *ICEIS (I)*, p. 205–208, 2012.
- [27] BARATA, M.; BERNARDINO, J.; FURTADO, P. Ycsb and tpc-h: Big data and decision support benchmarks. In: IEEE. *Big Data (BigData Congress), 2014 IEEE International Congress on*. [S.l.], 2014. p. 800–801.
- [28] RUTISHAUSER, N.; NOURELDIN, A. Tpc-h applied to mongodb: How a nosql database performs. Feb, 2012.
- [29] TRANSACTION PROCESSING PERFORMANCE COUNCIL. *TPC BENCHMARK H Standard Specification. Revision 2.17.2*. San Francisco, 2017. Acessado em: 12 de maio de 2017. Disponível em: <<https://goo.gl/uaRRcv>>.