

# **Banco de Dados Oracle e Firebird: Análise de Desempenho através de uma Estrutura de Data Mart**

**Marcelo Henrique S. Amaro, Mateus Henrique G. Venancio, Patricia B. Ribeiro, Anderson F. Talon**

Curso de Tecnologia em Banco de dados - Faculdade de Tecnologia de Bauru (FATEC)

Rua Manoel Bento da Cruz, nº 30 Quadra 3 - Centro - 17.015-171 - Bauru, SP - Brasil

souzamarcelo793@gmail.com, mateushgv1996@gmail.com,  
patriciabellin@yahoo.com.br, anderson.talon@fatec.sp.gov.br

***Abstract.** In order for the large volume of data currently generated to be stored and queried, several technologies are used, among them: Business Intelligence (BI), Data Warehouse (DW) and Data Mart (DM). This research aims to contextualize such environments, as well as create a DM structure in two distinct databases and finally make a comparison between the banks used - Oracle and Firebird. It was concluded through the tests carried out, that the Oracle bank uses a higher consumption of the system when compared to its opponent; but Firebird has a much longer response time.*

***Resumo.** Para que o grande volume de dados gerados atualmente possa ser armazenado e consultado, diversas tecnologias são utilizadas, dentre elas: Business Intelligence (BI), Data Warehouse (DW) e Data Mart (DM). Esta pesquisa tem por objetivo contextualizar tais ambientes, além de criar uma estrutura DM em dois bancos de dados distintos e por fim fazer um comparativo entre os bancos utilizados – Oracle e Firebird. Concluiu-se através dos testes realizados, que o banco Oracle se utiliza de um maior consumo do sistema quando se comparado ao seu adversário; porém o Firebird possui um tempo de resposta muito maior.*

## **1. Introdução**

Com as constantes inovações das ferramentas de Tecnologia da informação (TI), a evolução dos *hardwares* e do grande aumento na capacidade de armazenamento de dados, criou-se a necessidade de novas aplicações. A partir disto surgiram os conceitos de *Data Warehouse (DW)*, *Data Mart (DM)*, e *Business Intelligence (BI)*, entre outras tecnologias que auxiliam a tomadas de decisões menos intuitivas e mais focadas em informação, sejam estas decisões estratégicas, táticas ou operacionais. Os conceitos descritos anteriormente têm como ponto em comum uma nova maneira de lidar com volumes gigantescos de dados, estruturados ou não, para tomada de decisões estratégicas, objetivando a organização ágil e coerente da informação, para que os gestores e líderes decidam de forma acertada e rápida.

Sendo assim, Elias (2013) afirma em seu artigo que pelo fato do *BI* ser um conceito abstrato e não possuir apenas uma forma de elaboração, o *DW* é a ferramenta

normalmente escolhida para a sua implementação já que possui a capacidade de armazenar grandes massas de dados consolidados de diversas fontes, sendo um grande armazém de informação, tendo como objetivo principal a precisão e qualidade no suporte as decisões. Machado (2010) relata que seria como construir armazéns de dados onde neles estariam inseridos a história da empresa, dos clientes, dos fornecedores e as operações que se mantiveram disponíveis e acessíveis para as consultas e análises. Já o *Data Mart* tem a mesma estrutura de um *Data Warehouse*, mas utilizando um foco mais específico na consulta e visualização da informação, podendo ser subconjunto de um *DW*. *BI* é um afunilamento maior dos dados coletados, referindo-se ao processo de coleta, compartilhamento e monitoramento das informações, transformando dados brutos em úteis para análise e decisão.

No Brasil temos uma grande carência de tecnologias voltadas ao armazenamento, organização e mineração de dados, para que os gestores de políticas públicas consigam decidir de forma rápida e acertada as ações necessárias ao bem-estar da população. Por ser um país de dimensões continentais, há uma grande diversidade populacional e geográfica, criando necessidades específicas à cada região.

Então, o objetivo desta pesquisa, é criar um *Data Mart* que possa receber e armazenar grandes volumes de dados. Gerar e organizar os dados de forma consistente, para testar o desempenho do *DM* em dois Sistemas de Gerenciamento de Banco de Dados (SGBD), ou seja, serão criados dois ambientes idênticos, um SGBD Oracle e outro SGBD Firebird, a fim de obter uma comparação qualitativa e quantitativa entre ambos e também será realizada a geração de gráficos e métricas, que facilitem a compreensão das informações obtidas. Além de contextualizar os ambientes de *BI*, *DW* e *DM*, desde suas definições, teorias, arquiteturas, formas de implementação e funcionamento.

Este artigo está estruturado da seguinte forma: na seção 2 foi descrito os conceitos de *Business Intelligence (BI)*; na seção 3, apresentado o tema referente à *Data Warehouse (DW)*; na seção 4 realiza-se uma descrição sobre *Data Mart (DM)*; na seção 5 explica-se os materiais e métodos utilizados para realização deste trabalho e na seção 6, mostrado os resultados obtidos.

## **2. *Business Intelligence (BI)***

Com toda a massa de dados que se apresenta nos dias atuais e tendo que ser absorvida e administrada pelas organizações, as mesmas se sentem pressionadas e forçadas muitas vezes a mudarem suas estruturas tecnológicas, para conseguirem acompanhar as constantes mudanças do mercado, e assim garantir a sua sobrevivência. Sendo assim, é necessário que as decisões a serem tomadas pelas organizações sejam muito bem pensadas e planejadas, para que se obtenha êxito e sucesso, aí entra a função do *BI*, ajudar na tomada de decisões. A empresa Microsoft (2011) afirma que *BI* cumpre uma simples promessa, melhorar o desempenho dos negócios e aperfeiçoar a tomada de decisões em toda a organização. Conclui ainda que, quando há certeza na visão dos dados e estes são sólidos, bem fundamentados e completos, pode-se esperar que suas decisões o ajudem a alcançar seus objetivos corporativos.

Porém, definir um conceito para o termo *BI* não é uma tarefa muito fácil. São encontradas várias definições sobre o assunto, em diversas literaturas e na internet. A

empresa de consultoria e pesquisa *Gartner* (s/d), por exemplo, o define como um assunto extenso, que inclui aplicativos, infraestrutura, ferramentas e também melhores práticas que permitem assim o acesso e análise às informações, para melhorar e otimizar o desempenho e as decisões das organizações. Já para Birman [2003 apud Gomes, 2010], no conceito de BI está incluso um conjunto de soluções de TI que atendem as necessidades da organização, ele faz a assimilação de forma rápida dos dados que são relevantes e os converte em relatórios dinâmicos, realiza simulações gráficas de cenário e gera informações estratégicas, que encontram tendências de mercado e aperfeiçoam o potencial competitivo.

Sendo assim, é possível concluir que mesmo tendo diferenças entre as definições de *Business Intelligence*, seu foco é sempre o auxílio na tomada de decisões das organizações. A Figura 1 ilustra uma arquitetura tradicional de *BI* com os seus principais componentes.

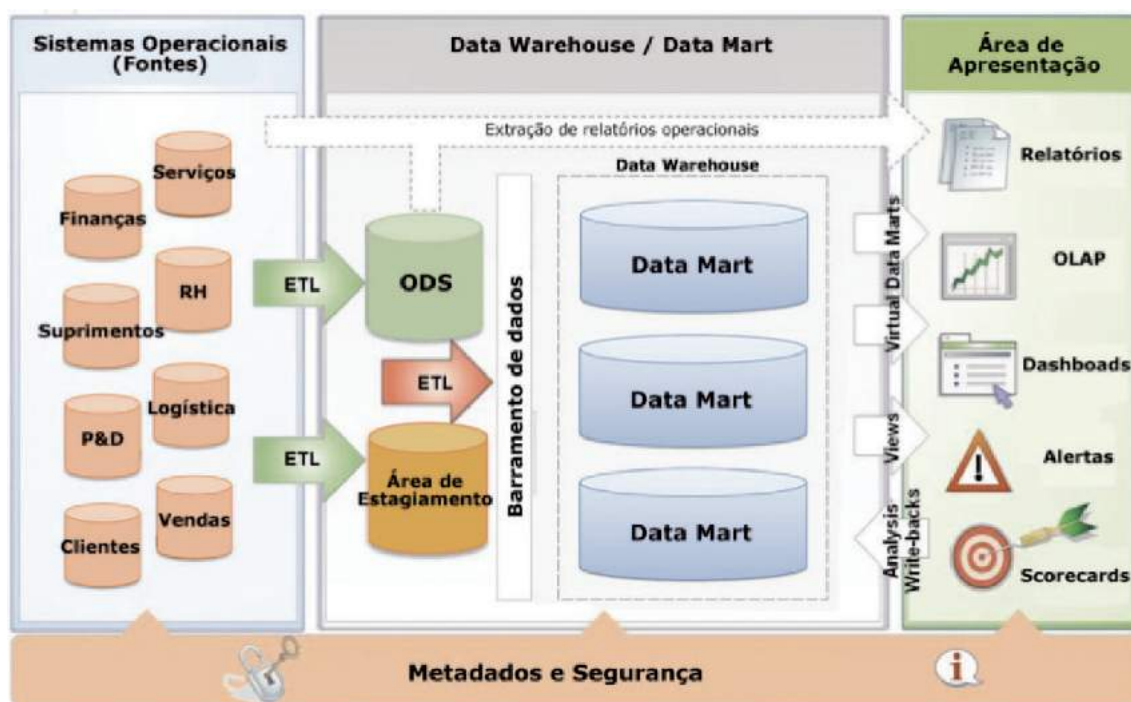


Figura 1. Arquitetura tradicional de BI

Fonte: Silva (2011 apud Ceci, 2012)

Na primeira camada da imagem chamada de “Sistemas operacionais”, temos as fontes de dados operacionais (transacionais) da organização. Como é possível observar, as bases estão separadas por setores: serviços, finanças, RH, suprimentos, P&D, logística, clientes e vendas. Após, inicia-se o processo denominado *extract transform load* (ETL – extração, transformação e carregamento), em que Gomes (2010) define como sendo o processo responsável pela extração dos dados contidos nos arquivos de origem, transformando-os em dados analíticos e por fim, fazer o envio ao *Data Warehouse* (DW) da organização, que na imagem acima está referenciado na segunda camada “*Data Warehouse* (mais detalhes na seção 3) / *Data Mart* (mais detalhes na

seção 4)”. Por fim, na última camada – “Área de apresentação” - os dados que foram organizados no repositório (*DW*) serão consumidos nesta área através das ferramentas de relatórios; *OLAP* (que tem por finalidade de manipular os dados contidos no *DW*, ou seja, acessa os dados brutos e compartilha-os de forma clara e intuitiva facilitando a análise); *Dashboards* (que são painéis visuais que irão apresentar de forma dinâmica, um conjunto de informações (indicadores e métricas) com o intuito de auxiliar nas tomadas de decisões dos gestores); alertas e *Scorecards* ou *Balanced Scorecard* (*BSC* – Indicadores Balanceados de Desempenho, em português) é uma ferramenta de planejamento estratégico que visa medir o desempenho empresarial quando a organização tem suas metas e estratégias definidas, é feito o balanceamento entre os quatro indicadores da empresa: financeiro, clientes, processos internos e aprendizado e crescimento).

Ceci (2012) conclui explicando que não existe uma solução genérica de *BI*, ou seja, cada organização possui um cenário distinto, procurando respostas as suas perguntas específicas, e que a única coisa em comum é a arquitetura geral tradicional de *BI*.

Portanto, com o atual cenário do mercado, é mais do que fundamental que o conceito de *Business Intelligence* esteja inserido nas organizações. Ele será o divisor entre o sucesso e o fracasso dos empreendimentos empresariais, ou seja, poderá haver grandes catástrofes oriundas de más decisões, ou um grande êxito na decisão quando tomada de forma bem pensada.

Porém, um problema já é observado no modo de pensar e solucionar problemas nas organizações. Muitas vezes ocorre uma inversão, e é dito que, a tecnologia veio para resolver o problema de gestão e não apoiar. Elias (2013) relata que a contribuição que se espera de um profissional de *BI* é no aperfeiçoamento da sua gestão empresarial, não priorizando assim o uso da tecnologia em relação à necessidade do negócio. Ele conclui dizendo que o negócio que se espera do *BI*, é uma visão que vai além de usar apenas uma boa ferramenta, e sim, de se ter uma visão de negócio, analisando falhas nos processos decisórios, nos sistemas de informação, a fim de evitar um ofuscamento no momento em que os gestores forem avaliar as informações.

### **3. Data Warehouse (DW)**

Segundo Primak (2008), um *DW* é uma coleção de dados derivados de bases operacionais de suporte à decisão. Estas informações derivadas são referenciadas como dados “gerenciais”, “informativos” ou “analíticos”. O mesmo autor afirma que *DW* são construídos para armazenar e dar acesso às informações de forma que não sejam limitados por tabelas e linhas relacionais. Tais dados são oriundos de múltiplos sistemas, utilizados há vários anos e que continuam a ser utilizados.

Para Machado (2010), um *DW* é um armazém de dados históricos que tem como objetivo apresentar informações que identifiquem indicadores e evolução de valores, dentro de uma grande janela de tempo. Um *DW* normalmente armazena elementos de um período histórico de no mínimo três anos, pois seu objetivo é determinar padrões de comportamento, indicadores de crescimento por tempo, etc.

Sendo assim, *Data Warehouse* para Inmon [2005 apud Costa, 2012] é um conjunto de dados organizados, integrados e orientados sobre um determinado assunto que posteriormente ajudará nas tomadas de decisões.

Portanto o *DW* se preocupa em juntar e solidificar as informações de fontes internas da empresa, que muitas vezes são heterogêneas, e também de fontes externas. Ele sintetiza, limpa e filtra os dados para que possam ser analisados e dar suporte as tomadas de decisões.

### 3.1 Características do *Data Warehouse*

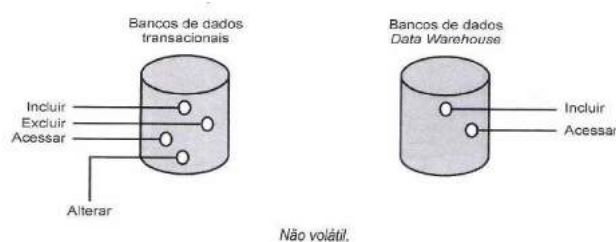
Machado (2010), explica a seguir as quatro características presentes no *Data Warehouse*, que são elas: orientado por assunto/tema; variação de tempo; não volátil; e por fim, integração.

Organizado e orientado por assunto/tema: Isto significa que os dados serão armazenados por assuntos (temas chaves) específicos da organização. Esses assuntos geralmente estão ligados aos processos principais da organização, como as entradas e saídas com agregação de valores. Os dados são mostrados de maneira simples, excluindo assim os elementos que não irão ser utilizados e nem serão relevantes nas tomadas de decisões;

Variação de tempo: Os dados de um *Data Warehouse* são muito precisos em relação ao tempo, os resultados mostrados refletem o momento exato em que as informações foram capturadas. Os dados são uma forma de *snapshot*, ou seja, um conjunto de registros estáticos; o que quer dizer que os elementos de um *DW* não podem ser atualizados. Outro elemento importante é a questão da data, já que estamos falando sobre uma estrutura de janelas de tempo;

Não volátil: Quando dizemos que um *Data Warehouse* não é volátil quer dizer que, os dados contidos nele, não serão atualizados registro a registro. Sendo assim o *DW* possui apenas duas operações, a inclusão de novos elementos e o acesso aos elementos em modo leitura. A Figura 2 representa bem essa não volatilidade do *DW*, enquanto no banco de dados transacionais temos várias opções: incluir, excluir, acessar e alterar, no banco de dados dimensional *DW* temos apenas as duas opções já citadas: incluir e acessar.

Integração: Como serão encontrados vários dados sobre tal assunto e em diversas fontes, tais dados terão que ser filtrados e armazenados no *DW* para que este se torne a única fonte de consulta. Sendo assim, esta é a característica mais importante de um *Data Warehouse*.



**Figura 2. Data Warehouse não volátil**

**Fonte: Machado, 2010**

### 3.2 Arquitetura do *Data Warehouse*

Neste tópico aparecerá o termo *Data Mart (DM)*, no qual teremos maiores informações na seção 4, e aqui estará sendo utilizado com a definição de ser um subconjunto do *Data Warehouse*.

Machado (2010) cita três arquiteturas possíveis em um *DW* e afirma que a escolha é uma decisão gerencial do projeto, e está baseada em fatores como infraestrutura disponível, ambiente de negócios, abrangência desejada e capacitação dos empregados das empresas. Muitas variáveis afetam a escolha da arquitetura, entre elas, o tempo de execução, o retorno do investimento, a velocidade dos benefícios, a satisfação do usuário e os recursos necessários para implantação e implementação. As estruturas possíveis citadas são: global, independente e integrada.

Arquitetura global: O autor cita esta arquitetura como sendo a que suporta a maior parte dos requerimentos ou necessidades de um *DW* com grande volume de requisições de informação dos vários departamentos da organização.

Arquitetura de *DM* independente: Para o autor esta arquitetura de *Data Marts stand alone* (estar sozinho, sem conexão com outro) atende as necessidades de departamentos específicos da organização, sem foco corporativo algum. Sendo a arquitetura preferida dos desenvolvedores para consulta dos *DW*, pois é isolada e encanta os olhos dos usuários.

Arquitetura de *DM* integrado: Os *DM* são criados separadamente por departamentos ou grupos, mas são integrados e conectados, oferecendo uma visão corporativa das informações. Este alto nível de integração torna esta arquitetura similar a global, mas com a vantagem de os dados estarem implementados por departamentos, e podendo ser compartilhados entre si. Como consequência, há um aumento no nível de complexidade dos requisitos.

Bonomo (2009), descreve que existem duas abordagens na implementação deste sistema, *top-down* e *bottom-up*. A implementação *top-down* se caracteriza quando a organização cria primeiro o *DW* e depois o segmenta, divide em áreas menores, gerando os *DM*, orientados por assuntos aos departamentos. E a implementação *bottom-up* sendo a situação inversa, quando a organização cria primeiro os *DM* segmentados, com custos e complexidades menores de execução e manutenção, e por fim une todos criando o *DW*.

### 4. *Data Mart (DM)*

Silva (2016) descreve *Data Mart (DM)* como um pequeno *Data Warehouse (DW)* especializado em um assunto ou setor da organização.

Machado (2010) afirma que um *Data Mart* permite acesso descentralizado às informações, criando a possibilidade de retorno rápido para avaliar os benefícios extraídos do investimento e um maior envolvimento do usuário final. Ainda dizendo que *DW* funciona como um grande conjunto de dados, enquanto os *Data Marts* seriam pequenas porções de dados, usando a mesma definição dita anteriormente, o *DW* seria o armazém (de dados) e os *DM* as prateleiras.

A Figura 3 a seguir, ilustra exatamente esta ideia. O funcionário possui uma série de papéis (podemos fazer uma analogia, aos dados de uma organização), e seu

objetivo é organizá-los conforme os respectivos assuntos. Sendo assim o *DW* é o de armazém de dados englobando todos os elementos, e o *DM* as estantes que separam as informações por assunto.

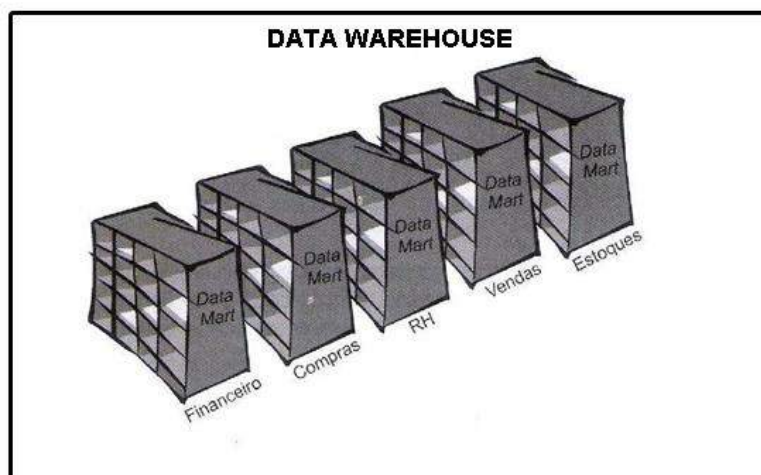


Figura 3. Data Warehouse - Armazém de dados

Fonte: Machado, 2010

Isto significa que, podemos utilizar as mesmas técnicas, a mesma arquitetura e métodos de implementação no *DM*, que utilizamos no *DW*, sempre com foco nas necessidades e características da organização.

#### 4.1 Modelagem dimensional

Para concepção de um *DM* é utilizada a técnica de modelagem dimensional. Segundo Kimball e Ross (2013), a modelagem dimensional “é uma técnica de projeto lógico que busca apresentar os dados em uma estrutura padrão e intuitiva que permite um acesso de alta performance”. Ou seja, quando utilizamos um modelo multidimensional, deixamos de ter o foco na coleta de dados, para nos preocuparmos com a consulta aos dados e a velocidade de retorno da informação.

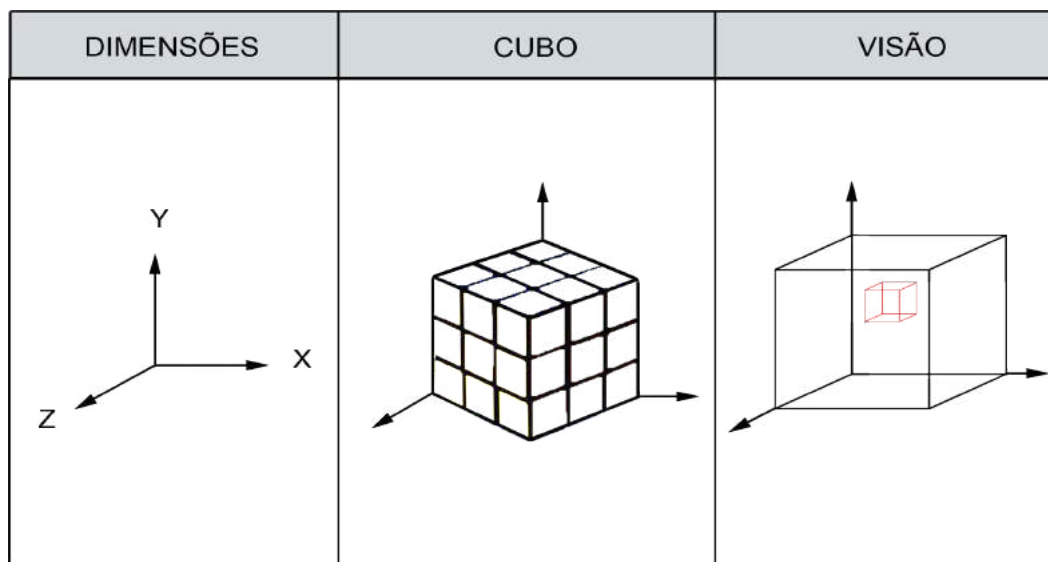
Machado (2010), afirma que um modelo multidimensional é formado por 3 unidades básicas: fatos, dimensões e métricas (variáveis).

E declara que **fatos** é tudo aquilo que representa o avanço dos negócios, do dia a dia de uma organização. Singh (2001), diz que os dados numéricos coletados ficam armazenados na tabela de fatos, por exemplo, os dados de vendas, estoque, assinaturas de periódicos, despesas, margem bruta, etc.

Machado (2010) descreve **dimensão** como as formas de figurar os dados, “são os ‘por’ dos dados: ‘por mês’, ‘por país’, ‘por região’” etc. Afirma ainda que dimensão pode conter muitos componentes, por exemplo, todas as ocorrências com datas fazem parte da dimensão tempo, e todas as cidades e regiões fazem parte da dimensão geográfica.

Imhoff, Gammemo e Geiger (2003) descrevem as **métricas** como um termo da modelagem dimensional que se refere a valores, geralmente numéricos, que medem algum aspecto do negócio. As medidas residem em tabelas fato.

O caminho mais fácil de visualizar um modelo dimensional é utilizando a figura de um cubo tridimensional, mas usualmente um modelo multidimensional tem mais de três superfícies conforme cita Machado (2010). A Figura 4 ilustra este conceito.



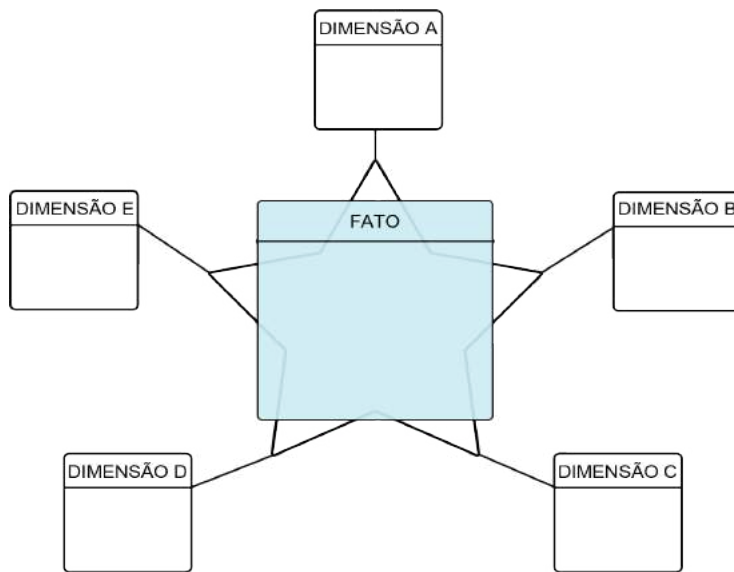
**Figura 4. Cubo de dados multidimensional**

**Elaborado: Os autores**

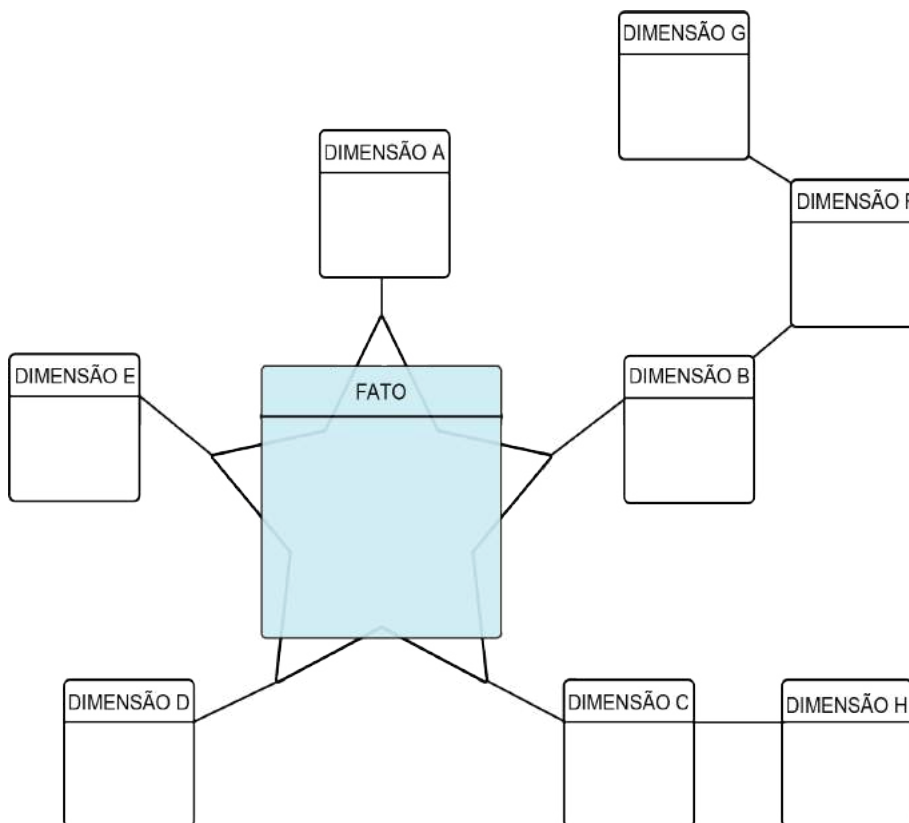
Kimball e Ross [2013 apud Floriano, Lemes, Heofacker, 2016] descrevem o conceito de modelo dimensional baseando-se na definição de bases multidimensionais. Afirma ainda que existem várias estruturas de bases de dados dimensionais, e que as mais utilizadas são os modelos estrela (*star schema*), e o modelo floco de neve (*snowflake*).

Machado (2010), explica que na composição do modelo estrela, há uma grande entidade central denominada fato (*fact table*) e um grupamento de entidades menores chamadas dimensões (*dimension tables*) organizadas ao redor da entidade central, formando uma estrela. O mesmo autor descreve o modelo *snowflake* como o resultado da decomposição de uma ou mais dimensões, que possuem hierarquia entre seus membros. Cita ainda que este modelo é o resultado da aplicação da terceira forma de normalização (que é responsável pela não redundância dos dados e aumentar a independência entre as relações) sobre as entidades dimensão. Sendo frequentemente utilizada pelos desenvolvedores, pois preserva a utilização de meios de armazenamento e evita redundância de valores textuais em uma tabela. As Figuras 5 e 6 representam respectivamente o modelo estrela e o modelo *snowflake* explicado pelo autor.





**Figura 5. Esquema estrela**  
**Elaborado: Os autores**



**Figura 6. Esquema floco de neve**  
**Elaborado: Os autores**

## 5. Materiais e métodos

Foi utilizada a pesquisa descritiva para coleta de informações, com uma abordagem qualitativa e um método indutivo, com o objetivo de analisar os dados para encontrar conceitos, princípios, significados entre as variáveis e seus resultados valorativos, desta forma foi determinado o melhor método de armazenamento, para analisar os dados do *Data Mart* e extrair as estatísticas de desempenho em dois Sistemas de Gerenciamento de Banco de Dados (*SGBDs*).

Para definição, criação do ambiente e realização dos testes de desempenho, foi necessário obter uma base de dados com um grande volume de informações, e que possuísse cronologia de mês e ano. Após pesquisa de uma base pública disponível na internet, definiu-se que os dados do Departamento de Polícia Rodoviária Federal – Ministério da Justiça e Cidadania<sup>1</sup> compreendia todos os fundamentos necessários para a realização do estudo.

Após o acesso ao site foi necessário acessar a aba Dados Abertos, onde encontram-se informações referentes as infrações de trânsito e aos acidentes (que foi utilizada neste trabalho) além do dicionário de dados de ambos; dentro da seção acidentes estão os registros separados por ano e em subcategorias: agrupados por ocorrência e por pessoa, sendo a primeira opção a escolhida. Foi realizado o *download* dos arquivos, ano a ano, desde 2013 até 2017 (que foram os anos escolhidos e que seriam necessários para o desenvolvimento deste projeto). Os arquivos que vem em formato zipado foram posteriormente descompactados.

Após a obtenção da base de dados, foi necessário realizar a extração dos dados para carregá-los no *Data Mart* desenvolvido, etapa esta que foi explicada anteriormente como *extract transform load (ETL – extração, transformação e carregamento)*. Sendo assim, com a utilização do software Excel da Microsoft, diversas informações (colunas) que existiam e que não seriam importantes para o nosso trabalho, como por exemplo: *id*, *data\_inversa* (que foi alterada para aparecer somente o mês e o ano), *dia\_semana*, *horário*, *km*, *classificação*, *sentido\_via*, *condição\_meteorologica*, *tracado\_via*, *pessoas*, *ignorados*, *feridos* e *veículos*; foram excluídas, sobrando assim, somente às seguintes colunas: *tipoacidente*; *feridosgraves*; *feridosleves*; *mortos*; *ileso*; *mês*; *ano*; *cidade*; *uf*; *br*; *causaacidente*; *fasedia*; *tipopista* e *usosolo*. Vale ressaltar que este procedimento foi realizado para todos os anos (2013 a 2017).

Após uma nova planilha do Excel foi aberta e todos os dados, referentes aos 5 anos, foram reunidos neste novo documento. O intuito desta etapa é deixar os dados ainda mais precisos, ou seja, com menos variações possíveis; sendo assim, todos os acentos foram removidos. Com esta etapa finalizada a planilha foi salva com o nome *Basedados* e extensão *.csv*.

O próximo passo foi organizar os dados que posteriormente vão ser carregados nas dimensões do *Data Mart*. Ainda na ferramenta *Excel*, esta etapa consistiu em separar individualmente as informações que se repetiam na tabela *Basedados*, ou seja, todas as informações sobre um mesmo assunto (*tipoacidente*, *mês/ano*, *cidade/uf* e *causaacidente*) foi copiado para uma nova planilha em branco e logo em seguida, na

---

<sup>1</sup> Link do site: <https://www.pr.f.gov.br/portal>

guia Dados, na aba Ferramentas de Dados foi utilizado a opção remover duplicatas para que apenas os dados distintos entre si ficassem e pudessem assim receber numeração (ID's); a Figura 7 ilustra um resumo de como esta etapa ficou concluída. Ao término desta etapa teremos um documento (planilha do *Excel*) para cada dimensão em formato .csv com seus dados e id's pré-estabelecidos.

IdTempo	Mês	Ano
1	1	2013
2	2	2013
⋮	⋮	⋮
15	3	2014
16	4	2014
⋮	⋮	⋮
29	5	2015
30	6	2015
31	7	2015
⋮	⋮	⋮
37	1	2016
⋮	⋮	⋮
56	8	2017
57	9	2017

IdCausa	causAcidente
1	Animais na Pista
2	Avaria no Pneu
3	Carga Mal Acondicionada
4	Condutor Dormindo
⋮	⋮
23	Ultrapassagem indevida
24	Velocidade incompatível

IdTipoAcidente	tipoAcidente
1	Atropelamento de animal
2	Atropelamento de pedestre
3	Atropelamento de pessoa
4	Capotamento
⋮	⋮
21	Saida de pista
22	Tombamento

IdBr	numBr
1	0
2	1
3	2
4	10
5	20
⋮	⋮
141	931
142	

IdCidade	cidade	Estado
1	ACRELANDIA	AC
2	ASSIS BRASIL	AC
3	BRASILEIA	AC
4	BUJARI	AC
⋮	⋮	⋮
2088	WANDERLANDIA	TO
2089	XAMBIOA	TO

Figura 7. Separação dos dados por assunto

Elaborado: Os autores

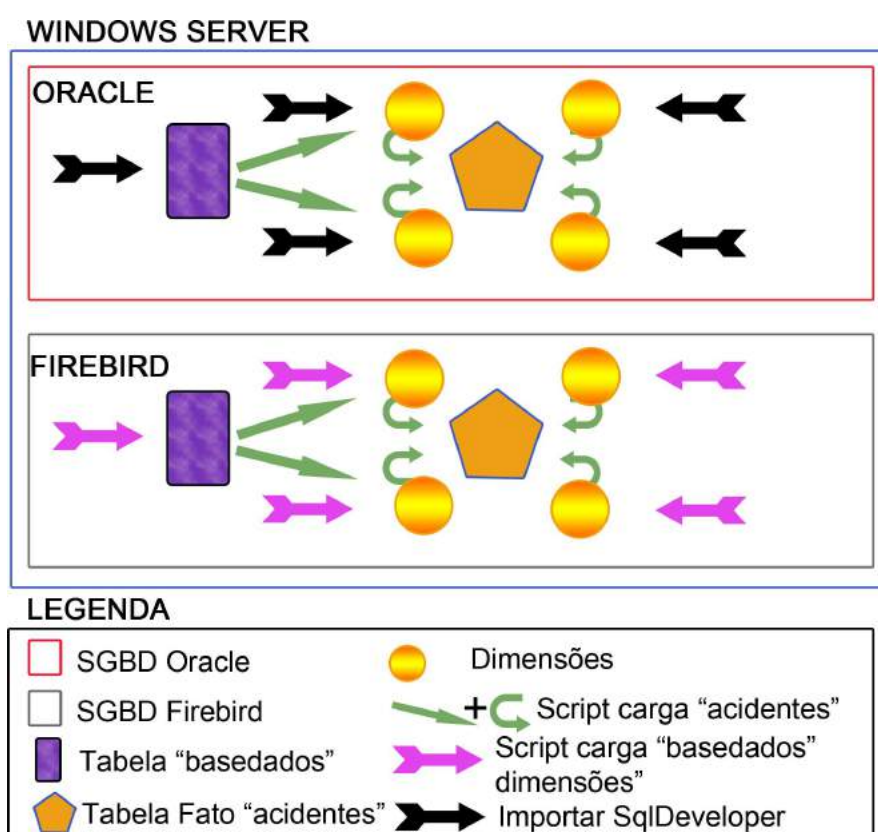
Após finalizar os processos de extração e transformação de dados, o passo seguinte é fazer o carregamento nas dimensões e basedados. Para isto, em um desktop com sistema operacional Windows 10 de 64 bits, memória RAM de 8,00 GB e processador Intel Core i5, foi instalado o Oracle VirtualBox 5.1.22 Edition, que é um aplicativo de virtualização multi-plataforma disponibilizado pela Oracle, no qual foi criado um ambiente virtual com *Windows Server 2008 R2*, memória RAM de 4GB e processador de dois núcleos. Foram instalados na virtualização dois SGBDs, o Oracle 12C com suporte a BI e o SGBD Firebird 2.5 sem suporte a BI.

No SGBD Oracle foi feita a criação de todas as dimensões e da basedados. Para melhor entendimento do leitor, usaremos a dimensão **causa** como exemplo de criação. Para esta tabela foi criado dois campos: IdCausa e causAcidente; o primeiro campo (IdCausa) recebeu como tipo number (4) (que é usado para receber valores numéricos) e causAcidente recebeu varchar2 (120) (usado para receber texto). Depois de criada, é necessário fazer o carregamento dos dados, porém, antes disso, no documento referente a esta dimensão (que já está salva com a extensão .csv) é preciso substituir os “;” (que o formato .csv do *Excel* atribui como forma de separar os dados como se fossem colunas) por “,” essa mudança é feita a partir da edição da planilha pelo bloco de notas.

Com esta alteração já finalizada, dentro do SGBD do Oracle, clique com o botão direito em cima da tabela desejada (neste exemplo, tabela causa), escolha a opção “importar dados”; na tela seguinte, vá até a pasta onde o arquivo .csv foi salvo, selecione-o e clique em “abrir”; na próxima tela – “Método de importação”, selecione a opção “inserir” e clique em “próximo”; para o próximo passo, no campo “Estabelecer correspondência por” selecione a opção “posição”, nesta tela já será possível ver os nomes das colunas que serão importadas, após clique em “próximo” e por fim em “finalizar”. Vale ressaltar que este procedimento, de criação e carregamento, foi feito em quase todas as dimensões e inclusive com a basedados; a única dimensão que teve

um carregamento diferente foi o fato principal (acidentes), que para tal um *script* na linguagem SQL foi criado.

No SGBD Firebird as tabelas foram criadas da mesma forma que no Oracle, porém, a troca dos tipos dos campos precisou ser realizada: de *varchar2* para *varchar* e de *number* para *numeric*. Para fazer o carregamento das dimensões (exceto a tabela acidentes) e basedados foram usados os mesmos arquivos csv, entretanto, um *script* SQL foi gerado para o carregamento; este comando SQL faz o insert nas tabelas buscando o caminho e o nome do arquivo (em csv) que estão salvos na máquina virtual. Para o carregamento da dimensão acidentes, assim como no Oracle, um script de carregamento foi utilizado, mas, com as devidas modificações realizadas para atender as necessidades do banco de dados Firebird. A Figura 8 ilustra o fluxo do carregamento dos dados nos SGBD'S.

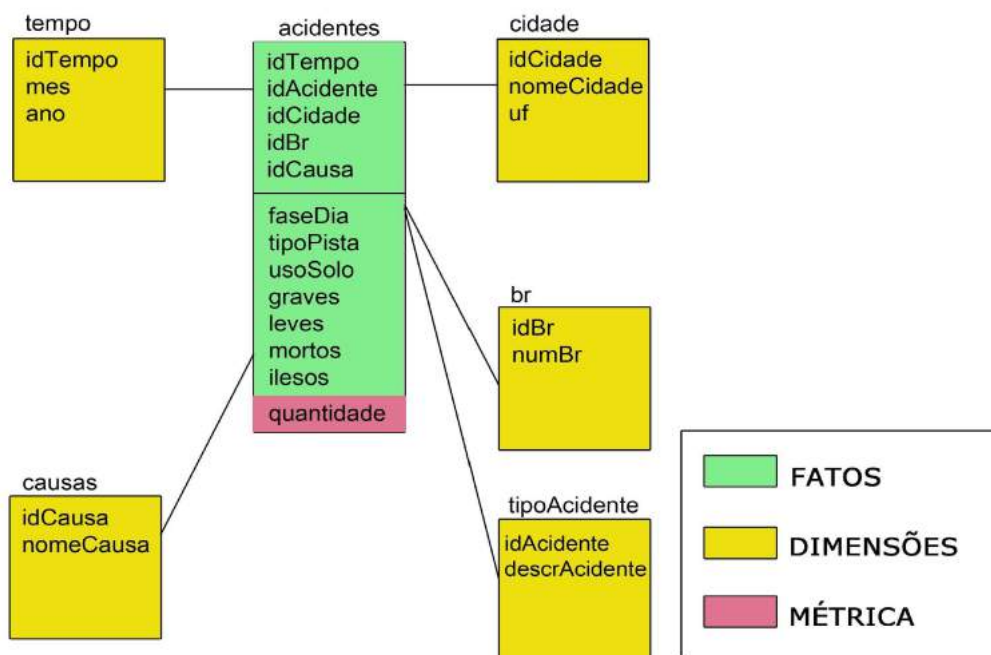


**Figura 8. Fluxo carregamento dados**

Elaborado: Os autores

## 6. Resultados

Após a avaliação das premissas identificadas na pesquisa, foi definido que a concepção de um *Data Mart* com modelagem dimensional e seus conceitos de fatos, dimensões e métricas, utilizando o esquema estrela (*star schema*), como o ideal para guardar e organizar o grande volume de informações da base de dados de acidentes nas rodovias federais, no período de 2013 a 2017, total de 640.815 linhas. A Figura 9 ilustra o *Data Mart* criado para servir de armazém, para as informações serem organizadas e analisadas.



**Figura 9. Modelo dimensional “estrela”**

**Elaborado: Os autores**

Para realizar o comparativo entre os bancos de dados Oracle e Firebird, uma consulta *select* foi desenvolvida e executada via *prompt* utilizando o comando *set timing on* no Oracle e *set stats* no Firebird para gerar os *benchmarks* (referências). O *script SQL select* faz a consulta nas dimensões “cidade”, “tempo”, “acidentes” e soma as quantidades de “feridos graves”, “feridos leves”, “mortos” e “ilesos” que constam na dimensão “acidentes”. O banco de dados Oracle por ter suporte a *Business Intelligence (BI)*, possui a cláusula *Cube* específica para a função *Group By*, sendo necessário somente um *select* para obter os resultados com subtotais para cidades, estados, meses e anos. O banco de dados *Firebird* não tem suporte a *BI* por isto não aceita a cláusula *Cube*, para que o resultado possa ser o mesmo, três consultas *selects* são necessárias para que o comando *Group By* agrupe separadamente por estados, meses e anos e depois seja feita a somatória dos subtotais. Para exemplificar de forma gráfica foram feitos testes com o *script select* nas interfaces *sqldeveloper* para o Oracle e no *IB Expert* para o Firebird. Os testes de performance foram feitos via *prompt* de comando, utilizando o Monitor de Desempenho do Windows para monitorar o desempenho dos processos dos bancos, durante a execução dos *scripts*.

A tabela 1 contém a instrução *select* utilizando o *group by Cube*. A figura 10 mostra através da interface *Sqldeveloper* como o banco de dados Oracle organiza as informações utilizando a cláusula *Cube*.

**Tabela 1. Select com cláusula Cube**

```
select ci.nomecidade, ci.uf, te.mes, te.ano,
sum(ac.mortos) as "SOMA MORTOS",
sum(ac.leves) as "SOMA FERIDOS LEVES",
sum(ac.graves) as "SOMA FERIDOS GRAVES",
sum(ac.ilesos) as "SOMA ILESOS"
from cidade ci, tempo te, acidentes ac
where ci.idcidade = ac.idcidade and te.idtempo = ac.idtempo
group by cube (ci.nomecidade, ci.uf, te.mes, te.ano);
```

	NOMECIDADE	UF	MES	ANO	SOMA MORTOS	SOMA FERIDOS LEVES	SOMA FERIDOS GRAVES	SOMA ILESOS
1	(null)	(null)	(null)	(null)	34576	332800	110795	867797
2	(null)	(null)	(null)	2013	8422	76773	26924	276147
3	(null)	(null)	(null)	2014	8234	74596	26236	243275
4	(null)	(null)	(null)	2015	6867	67743	22508	159317
5	(null)	(null)	(null)	2016	6398	65249	21423	112305
6	(null)	(null)	(null)	2017	4655	48439	13704	76753
7	(null)	(null)	1	(null)	2999	31590	9803	85114
8	(null)	(null)	1	2013	704	6645	2215	23630
9	(null)	(null)	1	2014	641	6669	2162	23104
10	(null)	(null)	1	2015	607	6168	1977	17902
11	(null)	(null)	1	2016	512	6321	1985	11187
12	(null)	(null)	1	2017	535	5787	1464	9291

**Figura 10. Sql Developer "Order by Cube"**

**Elaborado: Os autores**

Na primeira linha é demonstrada a somatória total dos mortos, feridos leves, feridos graves e ilesos durante todo o período da pesquisa. Campos sem dados como nomecidade, uf, mês e ano são preenchidos com *null*, pois o intuito da primeira linha é mostrar a somatória total de vítimas. Nas próximas linhas são feitas as somatórias por anos, depois por meses, por cidade e por estado, por meses e cidades, por meses e anos, por meses e cidades, etc... Conforme solicitado na instrução *select*.

As tabelas 2, 3 e 4 contém as instruções select utilizadas para as consultas. As figuras 11, 12 e 13 demonstram através da interface IBExpert como o banco de dados Firebird organiza as informações utilizando os comandos *group by* e *order by* sem a cláusula *Cube*. As informações são agrupadas, mas não organizadas com subtotais, necessitando de somatória após a execução do *select*.

**Tabela 2. Select 1 Firebird sem cláusula Cube**

```
select ci.nomecidade, ci.uf, te.mes, te.ano,
sum(ac.mortosdata) as "SOMA MORTOS",
sum(ac.feridoslevesdata) as "SOMA FERIDOS LEVES",
sum(ac.feridosgravesdata) as "SOMA FERIDOS GRAVES",
sum(ac.ilesosdata) as "SOMA ILESOS"
from cidade ci, tempo te, acidentes ac
where ci.idcidade = ac.idcidade and te.idtempo = ac.idtempo
group by ci.nomecidade, ci.uf, te.mes, te.ano
order by te.mes;
```

NOMECIDADE	UF	MES	ANO	SOMA MORTOS	SOMA FERIDOS LEVES	SOMA FERIDOS GRAVES	SOMA ILESOS
LAPA	PR	1	2013	1	5	0	18
LAPA	PR	1	2014	0	6	1	13
LAPA	PR	1	2015	0	5	0	16
LAPA	PR	1	2016	1	5	1	6
LAPA	PR	1	2017	2	9	0	7
LAJES	RN	1	2013	0	2	1	1
LAJES	RN	1	2014	1	1	0	5
LAJES	RN	1	2017	0	0	1	2
LAJEDO	PE	1	2013	0	0	0	2
LAJEDO	PE	1	2014	0	1	1	6
LAJEDO	PE	1	2015	0	1	0	2
LAJEDO	PE	1	2017	0	1	5	0
LAVRAS	MG	1	2013	0	20	4	36
LAVRAS	MG	1	2014	0	8	0	21
LAVRAS	MG	1	2015	0	12	1	14
LAVRAS	MG	1	2016	0	8	0	19
LAVRAS	MG	1	2017	0	12	0	21
LAMARAO	BA	1	2015	0	0	0	2
LARANJAL	MG	1	2013	0	5	0	7
LARANJAL	MG	1	2014	0	1	1	7

**Figura 11. IBExpert select 1**

**Elaborado: Os autores**

Nesta primeira instrução select executada no Firebird os dados foram agrupados por nomecidade, uf, mês, ano e ordenados por mês. É possível ver nas primeiras linhas a cidade de Lapa no Paraná nos meses de janeiro de 2013 a 2017 a somatória total das vítimas, e assim subsequentemente como solicitado na instrução *select*.



**Tabela 2. Select 2 Firebird sem cláusula Cube**

```

select ci.nomecidade, ci.uf, sum(ac.mortosdata) as "SOMA MORTOS",
sum(ac.feridoslevesdata) as "SOMA FERIDOS LEVES",
sum(ac.feridosgravesdata) as "SOMA FERIDOS GRAVES",
sum(ac.ilesosdata) as "SOMA ILESOS"
from cidade ci, acidentes ac
where ci.idcidade = ac.idcidade
group by ci.nomecidade, ci.uf
order by ci.uf;

```

NOMECIDADE	UF	SOMA MORTOS	SOMA FERIDOS LEVES	SOMA FERIDOS GRAVES	SOMA ILESOS
BUJARI	AC	8	57	18	119
CAPTIVA	AC	11	47	18	85
BRASILEIA	AC	1	20	4	11
ACRELANDIA	AC	3	28	7	61
ASSIS BRASIL	AC	0	0	0	2
EPITACIOLANDIA	AC	6	32	6	53
CRUZEIRO DO SUL	AC	0	0	0	1
MANOEL URBANO	AC	1	0	4	3
RIO BRANCO	AC	49	998	306	1976
PORTO ACRE	AC	1	7	5	11
SENA MADUREIRA	AC	9	54	16	82
SENAHOR GUIOMARD	AC	7	45	14	77
PLACIDO DE CASTRO	AC	7	7	5	28
XAPURI	AC	8	36	14	92
BELÉM	AL	5	32	19	25
CANAPI	AL	3	37	19	117
ATALAIA	AL	30	182	128	343
BRANQUINHA	AL	13	59	41	86
AGUA BRANCA	AL	8	41	21	58
CACIMBINHAS	AL	15	19	18	61

**Figura 12. IBExpert - select 2**

**Elaborado: Os autores**

Na segunda instrução *select* executada no Firebird os dados são agrupados por nomecidade, uf e ordenados por uf, ou seja, a somatória total das vítimas é feita primeiro por estados e depois por cidades.

**Tabela 3. Select 3 Firebird sem cláusula Cube**

```

select te.mes, te.ano, sum(ac.mortosdata) as "SOMA MORTOS",
sum(ac.feridoslevesdata) as "SOMA FERIDOS LEVES",
sum(ac.feridosgravesdata) as "SOMA FERIDOS GRAVES",
sum(ac.ilesosdata) as "SOMA ILESOS"
from tempo te, acidentes ac
where te.idtempo = ac.idtempo
group by te.mes, te.ano
order by te.ano;

```



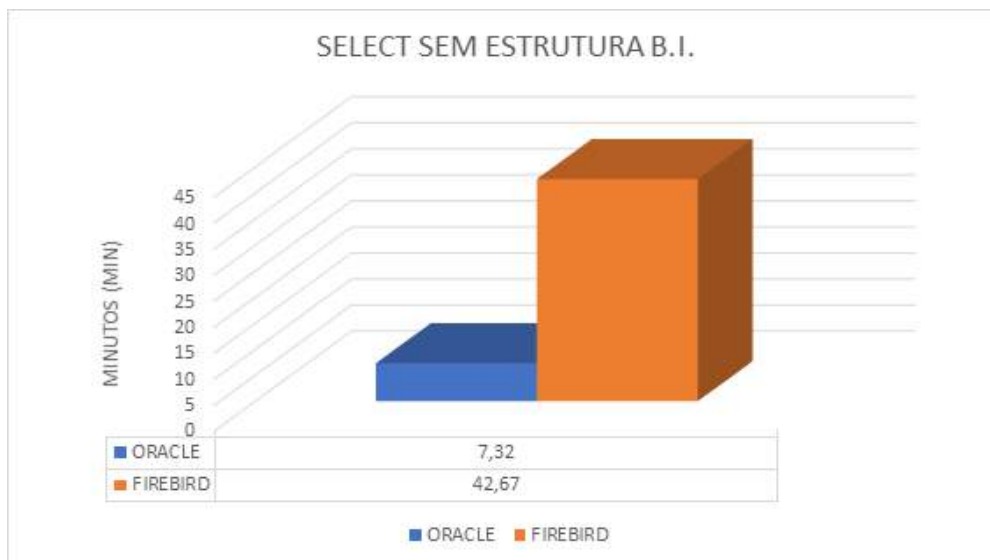
MES	ANO	SOMA MORTOS	SOMA FERIDOS LEVES	SOMA FERIDOS GRAVES	SOMA ILESOS
1	2013	705	6671	2232	23662
2	2013	607	5675	1918	20230
3	2013	652	6537	2283	23674
4	2013	595	5882	1995	21553
5	2013	724	6243	2200	23396
6	2013	698	6117	2233	22627
7	2013	734	6356	2282	24618
8	2013	692	6456	2260	22983
9	2013	725	6419	2298	22313
10	2013	680	6223	2265	22199
11	2013	738	6462	2332	23782
12	2013	876	7807	2664	25213
1	2014	641	6669	2162	23104
2	2014	547	5833	2051	20235
3	2014	739	6405	2208	21066
4	2014	704	6058	2154	21757
5	2014	727	5963	2285	20353
6	2014	702	5837	2150	18884
7	2014	688	5882	2130	19774
8	2014	704	6193	2269	19892

**Figura 13. IBEExpert - select 3**

**Elaborado: Os autores**

Na terceira instrução select feita no Firebird os dados são agrupados por mês, ano e ordenados por ano, sendo assim a somatória total das vítimas é feita primeiro por ano abrangendo os meses de janeiro até dezembro e assim sucessivamente.

O Gráfico 1 mostra o tempo decorrido para realização do *select* sem a estrutura *BI* nos dois SGBD's.

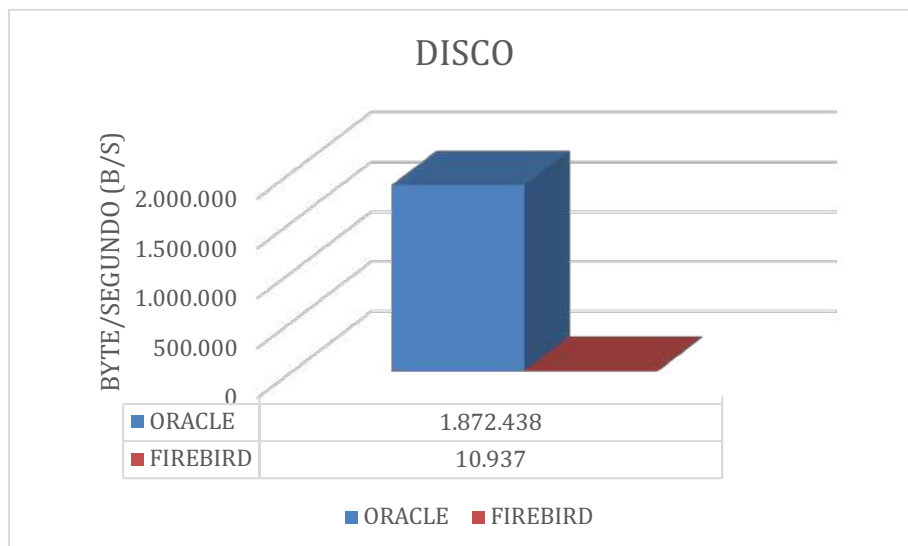


**Gráfico 1. Diferença tempo sem Cube**

**Elaborado: Os autores**

O Gráfico 1 demonstra que o banco de dados Oracle foi quase 6 vezes mais rápido do que o banco de dados Firebird quando não utiliza a cláusula *Cube*.

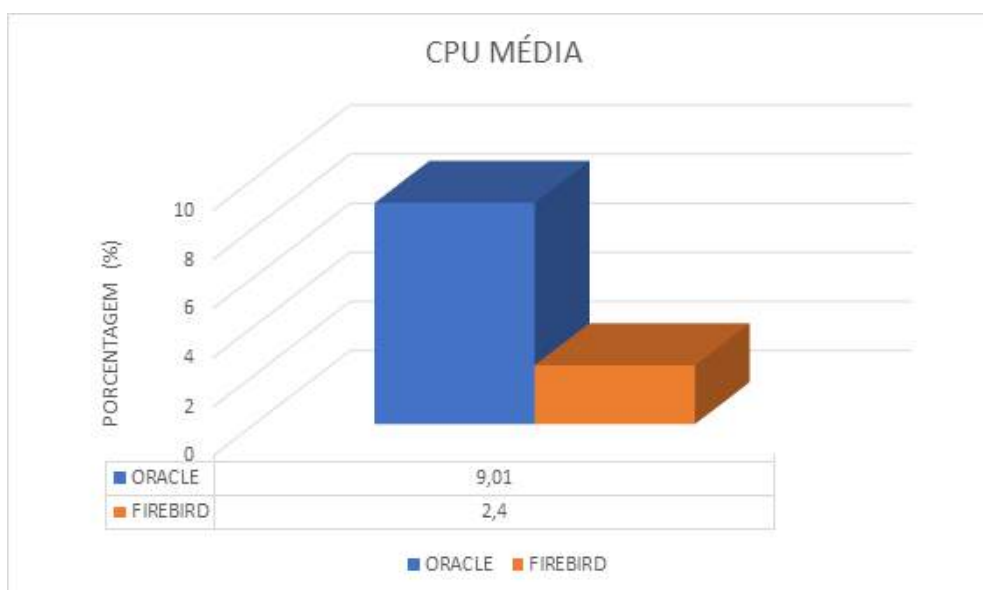
Os gráficos 2, 3 e 4 revelam o desempenho dos bancos durante a realização dos processos na execução do *select* sem a cláusula *Cube*.



**Gráfico 2. Comparativo de utilização do disco**

**Elaborado: Os autores**

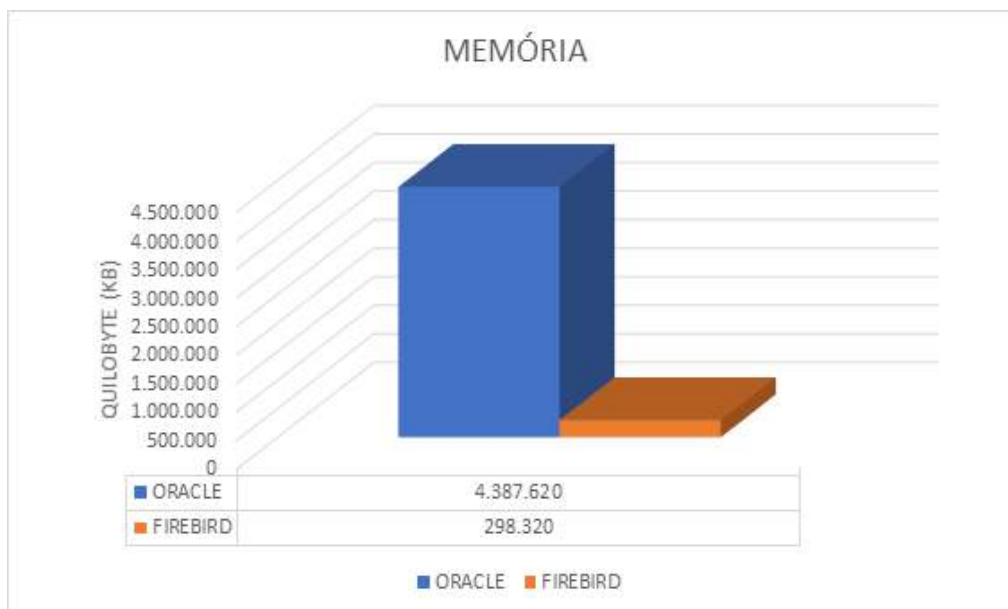
O Gráfico 2 mostra que o SGBD Oracle utiliza muito mais disco para executar a instrução SQL se comparado ao banco de dados Firebird.



**Gráfico 3. Comparativo de utilização média de CPU**

**Elaborado: Os autores**

O Gráfico 3 representa que o Firebird tem um consumo médio de CPU quase 4 vezes menor, que o banco Oracle.



**Gráfico 4. Comparativo de utilização da MEMÓRIA**

**Elaborado: Os autores**

O Gráfico 4 demonstra que a utilização da Memória também é maior no banco Oracle.

## 7. Conclusão

Conclui – se então que, de acordo com os resultados apresentados nos gráficos, o Banco de dados Oracle é o mais recomendável para ser usado em um ambiente onde se esteja preocupado com a tecnologia *BI*, pois além de possuir a cláusula *Cube* e outras que com apenas uma consulta *select* extrai-se os dados com totais e subtotais que facilitam o entendimento da informação, ele é realizado com menor tempo de execução, todavia com maior consumo de CPU, DISCO e MEMÓRIA quando comparado ao seu adversário. Pode se afirmar também que no Firebird sem suporte a *BI* é possível criar estruturas e realizar consultas multidimensionais, porém diversas consultas devem ser geradas como mostram os gráficos, com um tempo de execução maior, contudo com um consumo menor de *hardware* para execução de seus processos.

Vale ressaltar que em ambientes com pouco investimento é viável criar uma estrutura de *business intelligence*, demanda mais tempo para o retorno das informações e de mais habilidade dos profissionais envolvidos, contudo é possível.

## Referências

- Bonomo, P. (2009). “Construção de *Data Warehouse* e *Data Mart*”, <https://imasters.com.br/artigo/11178/gerencia-de-ti/construcao-de-data-warehouse-dw-e-data-mart-dm/?trace=1519021197&source=search>, outubro.
- Ceci, F. (2012). *Business Intelligence*: livro digital, Palhoça, UnisulVirtual, 2º edição.
- Costa, S.A.R. da (2012). “Sistema de *Business Intelligence* como suporte a Gestão Estratégica.”, Braga – Portugal, Escola de Engenharia – Universidade do Minho, p.135.
- Elias, D. (2013). “Arquitetura do *Business Intelligence*”, <https://www.binapratice.com.br/arquitetura-bi>, agosto.
- Elias, D. (2013). “O que o negócio realmente espera do *Business Intelligence*”, <https://www.binapratice.com.br/negocio-bi>, agosto.
- Floriano, A.C.; Lemes, E.F.e Heofacker, V.H.G. Análise de ferramentas de *Business Intelligence*. 2016. 141f. Trabalho de conclusão de curso (Análise e desenvolvimento de sistemas) – Universidade federal do Paraná, Curitiba.
- Gartner (s/d). “*Business Intelligence*”, <http://www.gartner.com/it-glossary/business-intelligence-bi>, setembro.
- Gomes, J. A. M. (2010). “Apoio tecnológico à estratégia organizacional: fatores para o sucesso no uso do *Business Intelligence*”, Belo Horizonte, Faculdade de Ciências Empresariais da Universidade FUMEC, p. 90.
- Imhoff, C., Gammemo, N., Geiger, J. G. (2003). *Mastering Data Warehouse Design: Relational and Dimensional Techniques*. Wiley Publishing. Indianapolis.
- Kimball, R.; Ross M. (2013). *The Data Warehouse Toolkit, The Definitive Guide to Dimensional Modeling, Third Edition*. John Wiley & Sons Inc. Indianapolis.
- Machado, F. N. R. (2010). Tecnologia e projeto de *Data Warehouse*, São Paulo, Érica, 5º edição.
- Microsoft Corporation (2011). “Como funciona *Business Intelligence*”, <https://www.microsoft.com/brasil/servidores/bi/about/how-does-bi-work.aspx>, setembro.
- Primak, F. V. (2008). Decisões com B.I.(*Business Intelligence*), Rio de Janeiro, Ciência Moderna.
- Silva, A. P. (2016). *Data Warehouse e Data Mart* como Ferramentas de Inteligência em negócios (BI) Monografia (Desenvolvimento para web) – Universidade estadual de Maringá, Maringá.
- Singh, H. S. (2001). “*Data Warehouse: Conceitos, Tecnologias, Implementação e Gerenciamento*”. Makron Books, São Paulo. Tradução: Rosemberg, M.