

INSTITUTO FEDERAL DO PARANÁ

EVANDRO TEIXEIRA

**OTIMIZAÇÃO DE CONSULTAS EM UM BANCO DE DADOS ORACLE
(*Tuning*)**

**PALMAS
2018**

EVANDRO TEIXEIRA

**OTIMIZAÇÃO DE CONSULTAS EM UM BANCO DE DADOS ORACLE
(*Tuning*)**

V.1

Trabalho de Conclusão de Curso
apresentado ao Curso Superior de
Sistemas de Informação do Instituto
Federal do Paraná como requisito
parcial de avaliação.

Orientador: Profº Bruno Guaringue
Trindade

**PALMAS
2018**

FOLHA DE APROVAÇÃO

EVANDRO TEIXEIRA

OTIMIZAÇÃO DE CONSULTAS EM UM BANCO DE DADOS ORACLE (*Tuning*)

Trabalho aprovado como requisito parcial para obtenção do título de bacharel, ao Curso Superior de Sistemas de Informação do Instituto Federal do Paraná formado pela seguinte banca examinadora:

Orientador: Profº Bruno Guaringue
Trindade

Palmas, 26 de Novembro de 2018.

Se você errou aprenda a acertar, não a desistir. (Schimits, Roberto).

RESUMO

A utilização de sistemas de informação vem aumentando dia após dia nas mais variadas atividades. O emprego de um sistema que possibilite gerir com precisão as atividades de uma organização, facilitando e também possibilitando um planejamento da mesma dando suporte à alta diretoria para uma tomada de decisão mais precisa, têm se tornado quase que prioridade dentro de uma empresa, uma vez que temos diversas dessas ferramentas a nossa disposição. Todavia, vemos um crescente volume de dados sendo produzido, o que torna o desempenho desses sistemas já não muito satisfatório, ou seja, o sistema continua eficaz (cumpre com o objetivo proposto inicialmente), porém não é eficiente (cumpre com o objetivo inicial, mas não com o custo de tempo desejado). O referido trabalho aborda o estudo e aplicação de técnicas de tuning visando a otimização de consultas SQL, também chamado de sintonização. A motivação desta pesquisa deu-se pela pertinente necessidade de melhora no desempenho do banco de dados do Grupo Sudati Ltda., que de acordo com relatos constantes dos usuários, entendeu-se que haviam gargalos em determinadas rotinas do sistema e que deveriam receber uma atenção especial.

Palavras-chave: Otimização de Banco de Dados. Banco de Dados. Consultas SQL.

ABSTRACT

The use of information systems is increasing day after day in the most varied activities. The use of a system that makes it possible to accurately manage an organization's activities, making it easier to plan and supporting the top management for more precise decision making, has become almost a priority within a company once we have several of these tools at our disposal. However, we see a growing volume of data being produced, which makes the performance of these systems no longer very satisfactory, that is, the system remains effective (fulfills the initially proposed objective), but is not efficient (fulfills the initial objective, but not at the desired time cost). This work addresses the study and application of tuning techniques aimed at the optimization of SQL queries, also called tuning. The motivation of this research was due to the pertinent need for improvement in the performance of the Sudati Ltda. Group database, which according to constant reports from users, it was understood that there were bottlenecks in certain routines of the system and that they should receive attention Special.

Key words:. Database Optimization. Database. SQL queries.

LISTA DE FIGURAS

Figura 1. Tamanho de cada Tablespace.....	13
Figura 2. Total de Tabelas na Base de Dados.....	14
Figura 3. Total de Tabelas do ERP Totvs Logix.....	14
Figura 4. Total de Linhas de cada tablespace.....	15
Figura 5. Arquitetura Banco de Dados Oracle.....	20
Figura 6. Informações referente ao armazenamento e disco do servidor de Banco de Dados.....	30
Figura 7. Informações referentes ao Sistema Operacional do servidor de Aplicação.....	31
Figura 8. Totalização referente à percepção de lentidão.....	31
Figura 9. Totalização de Sistemas/Rotinas.....	32
Figura 10. Totalização por setor.....	32

LISTA DE SIGLAS

BI – *Business Intelligence*

CNPJ – Cadastro Nacional de Pessoa Jurídica

COFINS - Contribuição para o Financiamento da Seguridade Social

ERP – *Enterprise Resource Planning*

FATEC – Faculdade de Tecnologia

ICMS - Imposto sobre Circulação de Mercadorias e Serviços.

IPI – Imposto sobre Produtos Industrializados

MDF - *Medium Density Fiberboard*

PIS - Programa de Integração Social

PR – Paraná

SGBD – Sistema Gerenciador de Banco de Dados

SQL - *Structured Query Language*

SUMÁRIO

1	INTRODUÇÃO	10
1.1	JUSTIFICATIVA.....	11
1.1.1	A Empresa	14
1.1.1.1	Missão	14
1.1.1.2	Visão	15
1.1.1.3	Valores	15
1.1.1.4	Responsabilidade com o ser humano	15
1.1.1.5	Responsabilidade com o meio ambiente	15
2	OBJETIVOS.....	15
2.1	Objetivo Geral	15
2.2	Objetivos Específicos	16
3	MATERIAIS E MÉTODOS.....	16
4.	REFERENCIAL TEÓRICO.....	17
4.1	Sistema Gerenciador de Banco de Dados	17
4.1.1	Linguagens De Banco De Dados (DDL, DML, DQL).....	18
4.2	Tablespaces.....	18
4.2.1	O que é o tablespace users?.....	19
4.2.2	O que é o tablespace system?	20
4.2.3	Gerenciamento de espaço em tablespaces.....	20
4.3	Algebra Relacional.....	21
4.3.1	Operações Relacionais Unárias: Seleção (SELECT)	21
4.3.2	Operações União (UNION), interseção (INTERSECTION) e subtração (MINUS)	21
4.3.3	Operações Relacionais Binárias (Junção JOIN) e divisão	22
4.4	Depuração (Debug)	22
4.5	Tuning de Banco de Dados	22

4.5.1	Tipos de Tuning	23
4.5.2	Técnicas de Tuning	23
4.5.3	Uma visão geral da sintonização de banco de dados em sistemas relacionais	23
4.5.4	Sintonização de Índices.....	24
4.5.5	Sintonização do Projeto de Banco de Dados	25
4.5.6	Sintonização de Consultas	25
4.5.7	Orientações adicionais para a sintonização de consultas	27
5.	Estado da Arte	28
6.	DESENVOLVIMENTO	29
6.1	Visão Geral.....	29
6.2	Configuração do Ambiente	29
6.3	Questionário (a fim de identificar os pontos de gargalo)	30
7.	CENÁRIO ATUAL.....	33
7.1	Análise 1: Conhecimentos De Frete.....	34
7.2	Análise 2: Faturamento.....	37
7.3	Análise 3: Notas De Entrada.....	39
7.4	Análise 4: Movimentação Estoque.....	40
7.5	Demais Relatórios.....	42
	CONCLUSÃO.....	43
	REFERÊNCIAS.....	45
	APÊNDICE.....	46

1 . INTRODUÇÃO

O emprego de sistemas de informação vem aumentando dia após dia nas mais variadas atividades. O uso de um sistema que possibilite gerir com precisão as atividades de uma organização, que facilite e automatize a transmissão de informações ao governo sejam elas informações fiscais, tributárias, previdenciárias, entre outras e que também possibilite um planejamento da organização dando suporte à alta diretoria para uma tomada de decisão mais precisa, têm se tornado quase que prioridade dentro de uma empresa, uma vez que temos diversas dessas ferramentas a nossa disposição. Em contrapartida, vemos um grande volume de dados sendo produzido, o que torna o desempenho desses sistemas já não muito satisfatório, ou seja, o sistema continua eficaz (cumpre com o objetivo proposto inicialmente), porém não é eficiente (cumpre com o objetivo inicial, mas não com o custo de tempo desejado).

De acordo com LEITE A.K.L, et al., na etapa de desenvolvimento do Banco de Dados pode não possível avaliar o seu desempenho (embora hajam controvérsias), porém, após o banco estar num ambiente de produção, é possível identificar as falhas que ainda não haviam sido consideradas e aplicar as técnicas para otimização.

Conforme Couto, E. (2006), o objetivo da otimização é exibir a informação requisitada pelo usuário com o menor custo e tempo. Conforme LEITE A.K.L, et al., nesta fase de aperfeiçoamento, pode se aplicar o *Tuning*, que irá otimizar o acesso aos dados, fazendo com que o tempo de resposta fornecido pela aplicação seja cada vez menor, tornando o sistema mais eficiente num todo.

Para se ter um melhor entendimento do que é o tuning, sua tradução literal nada mais é que: afinação, sintonização, refinamento, ajuste.

Segundo Navathe & Elmasri (2005), um projeto e implementação de Banco de Dados se dá nos seguintes passos:

1. Levantamento e análise de requisitos.
2. Projeto conceitual do banco de dados.
3. Escolha de um SGBD.
- 4- Mapeamento do modelo de dados (também chamado projeto lógico de banco de dados).
5. Projeto físico do banco de dados.

6. Implementação e sintonização (*tuning*) do sistema de banco de dados.

No referente trabalho, será abordado apenas a sintonização (*tuning*).

Conforme Navathe & Elmasri (2005), nesta fase, o banco de dados e os programas de aplicação são implementados, testados e eventualmente utilizados em produção. São testadas também várias transações e aplicações individualmente e, posteriormente em conjunto. Geralmente, esta fase revela necessidades de alterações no projeto físico, na indexação, na reorganização e na alocação de dados, atividade esta que é denominada de sintonização (afinação, *tuning*) do banco de dados.

Sintonizar é uma atividade contínua, parte da manutenção do sistema que perdura durante todo o ciclo de vida de um banco de dados, contanto que o banco de dados e as aplicações continuem evoluindo ou à medida que forem surgindo problemas de desempenho.

A motivação desta pesquisa deu-se a partir da necessidade de melhora no desempenho do banco de dados do Grupo Sudati Ltda., que conforme relatos de usuários percebeu-se que havia lentidão em determinadas rotinas. Motivações em paralelo deram-se também pela afinidade para com a área de banco de dados, e, também pelo fato do objeto da pesquisa ser pertinente, visto que é um assunto que desperta um grande interesse entre profissionais da área de BD e também estar relacionado a um dos tópicos para submissão de artigos do SBBD (Simpósio Brasileiro de Banco de Dados) sendo ele, a avaliação de Desempenho e Benchmarking de Bancos de Dados.

1.1 JUSTIFICATIVA

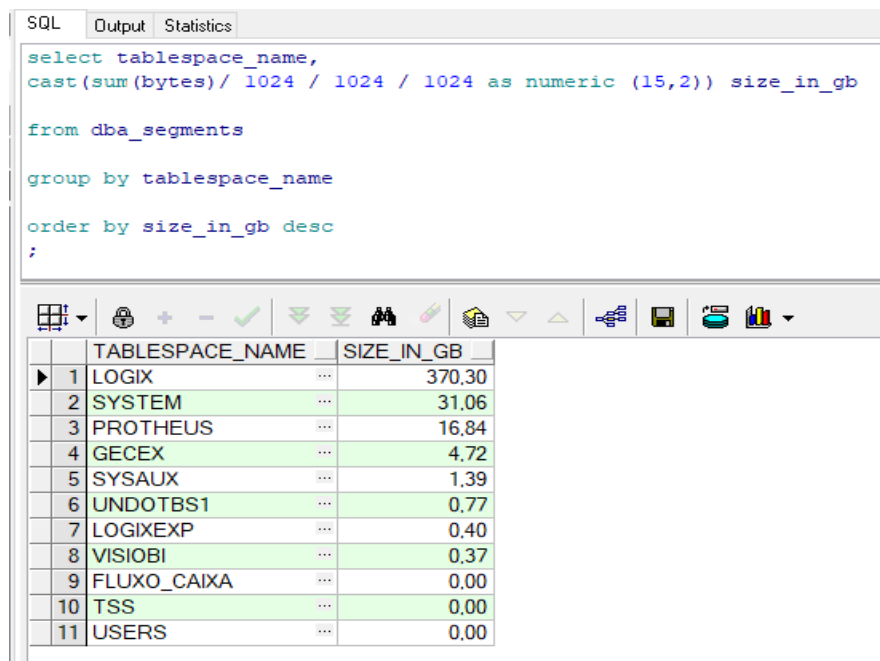
A justificativa para este trabalho, deu-se a partir da busca pela melhoria no desempenho de execução das queries a partir do ERP, e, posteriormente nas *views* que serão executadas através da ferramenta de Business Intelligence.

Atualmente, algumas das tabelas com mais sobrecarga têm um tempo relativamente alto quando solicitado um retorno dos seus dados. Esse tempo, fica em maior evidência quando executados alguns relatórios da ferramenta de BI, quando o usuário passa como parametro um período superior a 1 ano, (exibir faturamentos de 2014 a 2016 por exemplo) que é um volume de dados considerável. Esses relatórios por sua vez, são montados a partir da criação de uma *view*, que é

criada sem qualquer tipo de análise realizada antes e não é pensada de forma que vise o melhor desempenho. A única “análise” realizada, é da depuração das rotinas do ERP onde são identificadas as tabelas relacionadas à consulta ou transação. Na ferramenta de BI, o usuário pode utilizar de parametros para filtrar os relatórios, tais como: empresa, período, nota fiscal, entre outros.

Com isso, a proposta para melhoria de desempenho e otimização, é a aplicação de técnicas de tuning de banco de dados para redução no tempo de execução das queries, trabalhando em cima de índices nas tabelas e/ou reescrita de queries.

O tamanho total de tablespaces é de 425,76 gb, que se subdivide em diversos tablespaces, sendo eles correspondentes à webservices, ao ERP, Sistema de Comércio Exterior, entre outros conforme imagem abaixo.



```
SQL Output Statistics
select tablespace_name,
cast(sum(bytes)/ 1024 / 1024 / 1024 as numeric (15,2)) size_in_gb
from dba_segments
group by tablespace_name
order by size_in_gb desc
;
```

	TABLESPACE_NAME	SIZE_IN_GB
1	LOGIX	370,30
2	SYSTEM	31,06
3	PROTHEUS	16,84
4	GECEX	4,72
5	SYSAUX	1,39
6	UNDOTBS1	0,77
7	LOGIXEXP	0,40
8	VISIABI	0,37
9	FLUXO_CAIXA	0,00
10	TSS	0,00
11	USERS	0,00

Figura 1. Tamanho de cada Tablespace. FONTE – Autor.

Podemos ver que há também uma grande quantidade no total de tabelas na Base de Dados, em torno de 20.000 tabelas conforme imagem abaixo:

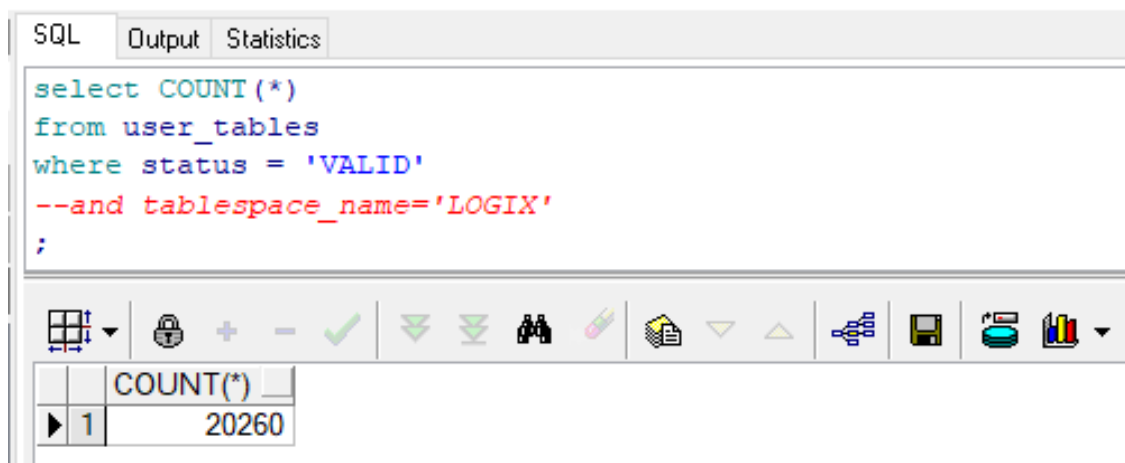


Figura 2. Total de Tabelas na Base de Dados. FONTE – Autor.

E desse total de 20.000 tabelas, cerca de 14.000 são pertencentes ao tablespace do sistema ERP Logix conforme imagem abaixo:

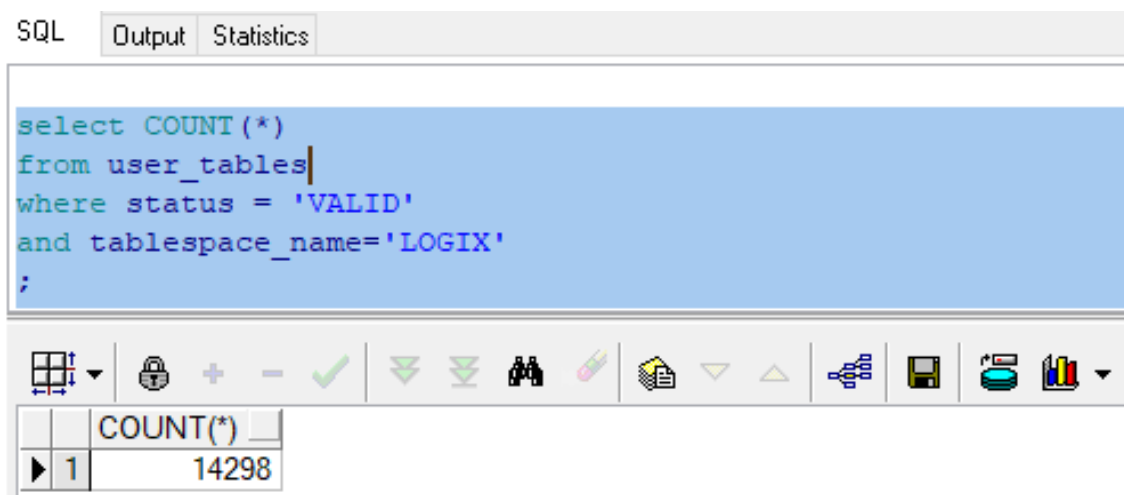


Figura 3. Total de Tabelas do ERP Totvs Logix. FONTE – Autor.

Nota-se um volume grande de registros (linhas) nos tablespaces dos principais sistemas utilizados sendo eles: ERP Logix e Gecex – Gestão de Comércio Exterior, conforme imagem abaixo:

SQL Output Statistics		
<pre>SELECT owner usuario, SUM(num_rows) num_linhas FROM all_tables GROUP BY owner</pre>		
	USUARIO	NUM_LINHAS
1	GECEX	15659015
2	MDSYS	23505
3	LOGIX	795447440
4	CTXSYS	736
5	OLAPSYS	
6	SYSTEM	919
7	APEX_030200	100
8	LOGIXEXP	1282755
9	EXFSYS	
10	XDB	0
11	SYS	1751

Figura 4. Total de Linhas de cada tablespace. FONTE – Autor.

1.1.1 A EMPRESA

Fundado em janeiro de 1998, o Grupo Sudati é hoje referência no setor de base florestal no Brasil. Sediado em Palmas (PR), o Grupo possui quatro unidades industriais no Paraná e em Santa Catarina, sendo responsável por cerca de 30% das exportações de placas de compensado do país. Além das unidades industriais, o Grupo Sudati conta também com um escritório administrativo em Curitiba.

A unidade industrial de Otacílio Costa é um marco especial para o Grupo: em 2008 esta unidade produziu o primeiro painel de MDF do Grupo e do Estado de Santa Catarina.

A Sudati opera seguindo rigorosos padrões internacionais de qualidade. Seu moderno parque industrial proporciona inovação e excelência de operação. Suas matérias-primas certificadas e seu capital humano capacitado completam o ciclo de esforços em nome da inovação e qualidade.

1.1.1.1 MISSÃO

Ser a empresa com maior proximidade aos clientes dentre as indústrias de painéis de madeira, com foco na agilidade e parcerias sustentáveis, gerar valor e resultado em cada relacionamento.

1.1.1.2 VISÃO

Entregar aos clientes soluções, qualidade e facilidade com crescimento sustentável.

1.1.1.3 VALORES

Comprometimento, valorização de pessoas, responsabilidade social e ambiental, busca por tecnologia, inovação e sustentabilidade.

1.1.1.4 RESPONSABILIDADE COM O SER HUMANO

O Grupo Sudati exercita sua responsabilidade social através de trabalhos de educação ambiental e ações comunitárias que valorizam o desenvolvimento humano e a vida.

O comprometimento do Grupo Sudati com o desenvolvimento social não é pontual: além dos projetos, a busca pelo desenvolvimento intelectual e o preparo para avanços profissionais e pessoais são constantes dentro e fora da organização.

1.1.1.5 RESPONSABILIDADE COM O MEIO AMBIENTE

Uma das prioridades do Grupo Sudati é a manutenção do meio ambiente. Campanhas de reciclagem, manejo sustentável, responsabilidade com o uso dos recursos naturais e rigorosos processos visando a qualidade produtiva são resultados deste posicionamento.

A empresa também promove parcerias florestais através de investimentos, troca de tecnologias e controle de certificações através de acordos com pequenos e médios produtores.

A Sudati firma com seus *stakeholders* a missão de se manter sempre à frente do mercado sem dispensar a responsabilidade ambiental.

2. OBJETIVOS

2.1 OBJETIVO GERAL

O objetivo geral deste trabalho é aplicar técnicas de *tuning* nas queries visando a otimização das consultas em um banco de dados oracle.

2.2 OBJETIVOS ESPECÍFICOS

- a) Estudar técnicas de *tuning* a partir de revisão bibliográfica;
- b) Levantar requisitos com o usuário a fim de identificar possíveis gargalos nas rotinas;
- c) Depurar estas rotinas a fim de identificar os comandos sql;
- d) Analisar a depuração e, caso se faça necessário, reescrever as queries, desfazer, criar ou recriar os índices das tabelas relacionadas;
- e) Testar as queries e avaliar a possível melhora no tempo de execução.

3. MATERIAIS E MÉTODO

Para o projeto, inicialmente será realizada uma pesquisa bibliográfica a fim de aprimorar as principais técnicas de *tuning* de banco de dados, bem como suas melhores práticas e métodos.

Após isso, serão identificadas as principais rotinas com lentidão e possíveis gargalos através de uma pesquisa com o usuário por meio de formulário.

Identificadas as rotinas, as mesmas serão executadas com a ferramenta de debug do ERP a fim de extrair os comandos *sql* e verificar seu tempo de execução. Com isso, serão aplicadas técnicas de *tuning* nas queries para comparar a melhora no tempo de execução.

O ERP que será utilizado será o TOTVS Logix, bem como sua ferramenta de BI correspondente, fornecida pela empresa desenvolvedora de software TOTVS. O Sistema Gerenciador de Banco de Dados utilizado será o Oracle *11g Enterprise Edition Release 11.2.0.1.0*.

Para a administração e manipulação do Banco de Dados, bem como a aplicação das técnicas de *tuning*, será utilizada a ferramenta PL/SQL *Developer* versão 8.0.0.1480.

Para demonstrar os resultados obtidos, será utilizada a ferramenta CamStudio, gravando a execução dos comandos sql exibindo o tempo de execução, pois assim evitamos qualquer tipo de desconfiança com relação aos dados apresentados (como manipulação de imagem alterando o tempo de execução por exemplo) e aplicamos as técnicas no próprio ambiente de produção, sem qualquer tipo de interferência interna e externa coletando informações mais precisas com

relação à melhora no desempenho.

4. REFERENCIAL TEÓRICO

4.1 SISTEMA GERENCIADOR DE BANCO DE DADOS

Conforme Silberschatz *et. al* (2006), um SGBD (Sistema Gerenciador de Banco de Dados) é uma coleção de dados inter-relacionados juntamente com conjunto de programas para acesso desses dados. Essa coleção é comumente chamada de Banco de Dados e contém informações pertinentes à uma determinada organização. O principal objetivo de um SGBD é possibilitar a recuperação da informação do Banco de Dados que seja tanto conveniente quanto eficiente.

Silberschatz (2006) *et. al* diz ainda que os SGBD's são projetados para trabalhar com grandes blocos de informação. Gerenciar estes dados, demanda definir estruturas para armazenamento e mecanismos para manipulação dessas informações. Além disso, é crucial a garantia da segurança das informações ali armazenadas, embora ocorram falhas na aplicação ou ataques visando o roubo das mesmas.

Uma outra análise sobre conceitos de Banco de Dados e SGBD do ponto de vista de Navathe & Elmasri (2005), é que um banco de dados é uma coleção de dados relacionados. Os dados são informações que podem ser gravadas e que possuem um significado implícito.

Essas informações são uma coleção de dados com um significado implícito, conseqüentemente, um banco de dados. Os mesmos autores, afirmam ainda que um banco de dados pode ser de qualquer tamanho e de complexidade variável.

Sob a perspectiva de Navathe & Elmasri (2005), um sistema gerenciador de banco de dados (SGBD) é uma coleção de programas que permite aos usuários criar e manter um banco de dados. O SGBD é, portanto, um sistema de software de propósito geral que facilita os processos de definição, construção, manipulação e compartilhamento de bancos de dados entre vários usuários e aplicações. A definição de um banco de dados implica especificar os tipos de dados, as estruturas e as restrições para os dados a serem armazenados em um banco de dados.

4.1.1 LINGUAGENS DE BANCO DE DADOS (DDL, DML, DQL)

Um banco de dados necessita de linguagens tanto para sua definição, quanto para sua manipulação, tendo em vista que alguns autores consideram a consulta um outro tipo de linguagem.

Segundo Navathe & Elmasri (2005), a linguagem DDL (*Data Definition Language*) é usada pelo Database Administrator (DBA) e pelos projetistas do banco de dados para definir ambos os esquemas. O SGBD terá um compilador DDL cuja função é processar os comandos DDL a fim de identificar os construtores e para armazenar a descrição do esquema no catálogo do SGBD. Os autores ainda sugerem que quando os esquemas do banco de dados estiverem compilados e o banco de dados populado com os dados, os usuários devem ter alguns meios para manipular esse banco. As manipulações típicas são a recuperação, inserção, remoção e modificação dos dados. O SGBD fornece uma série de operações, ou uma linguagem chamada DML (*Data Manipulation Language*).

Segundo Salish, A. (2010), os comandos DQL são usados para exibir registros de tabelas para o usuário/DBA. Os dados atuais na forma de tabelas são armazenados no disco rígido do Oracle Database Server e os comandos DQL permitem a visualização das informações / registros armazenados no banco de dados no monitor por meio do comando SELECT.

4.2 TABLESPACES

De acordo com Legatti (2011), o Oracle armazena dados logicamente em tablespaces e fisicamente em arquivos de dados (*datafiles*). Apesar dos arquivos de dados e os tablespaces se encontrarem demasiadamente "inter-relacionados", os mesmos dispõem de diferenças importantes conforme o autor:

- Um banco de dados Oracle constitui-se em uma ou mais unidades de armazenamento lógicas chamada de tablespaces, que armazenam coletivamente todos os dados do banco de dados.
- Cada tablespace em um banco de dados Oracle consiste em um ou mais arquivos denominados arquivos de dados (*datafiles*), que são estruturas físicas compatíveis com o sistema operacional no qual o Oracle é executado.

- Os dados de um banco de dados são armazenados coletivamente nos arquivos de dados que constituem cada tablespace do banco de dados.

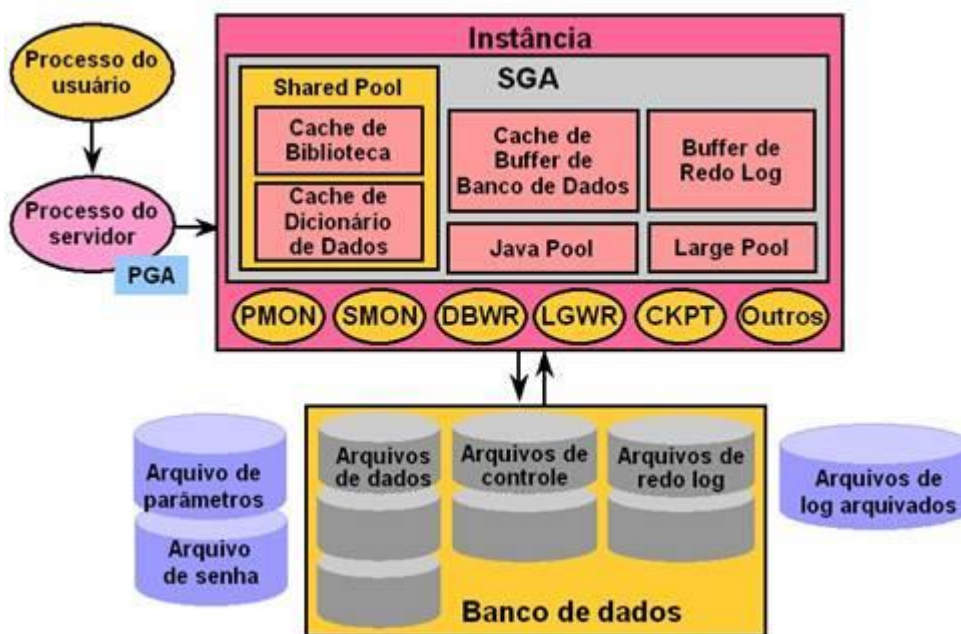


Figura 5. Arquitetura Banco de Dados Oracle. (Rezende, R. 2007).

O Oracle agrupa esses conjuntos de arquivos de dados sob a proteção de um objeto de banco de dados chamado tablespace. Antes mesmo de inserir dados em um banco de dados Oracle, primeiro é necessário a criação de um tablespace e posteriormente uma tabela dentro desse tablespace que conterá os dados, segundo Legatti (2011). Pode-se observar também, que na criação de um banco de dados utilizando o **DBCA**, o Oracle por padrão sempre cria um tablespace de dados chamado USERS. Ao criar uma tabela, se faz necessário realizar a inclusão de todas as informações pertinentes ao tipo de dados que se deseja manter.

4.2.1 O QUE É O TABLESPACE USERS?

Como citado anteriormente, Legatti (2011) afirma que um tablespace USERS geralmente é o tablespace padrão para os usuários. Caso um usuário crie um objeto, bem como uma tabela ou um índice sem especificar o tablespace, o Oracle

realiza a criação no tablespace padrão do usuário, caso o tablespace padrão do usuário seja definido para utilizar o tablespace USERS.

4.2.2 O QUE É O TABLESPACE SYSTEM?

O autor ainda destaca, que um tablespace SYSTEM (tablespace de sistema) é uma parte obrigatória de todo banco de dados Oracle. É nesse tablespace que o Oracle armazena todas as informações necessárias para o seu próprio gerenciamento, ou seja, SYSTEM é o tablespace mais crítico do banco de dados porque ele contém o dicionário de dados. Caso, por algum motivo ele se tornar indisponível, a instância do Oracle abortará. Por conta disto, o tablespace SYSTEM nunca poderá ficar offline, ao contrário de um tablespace comum como, por exemplo, o tablespace USERS.

4.2.3 GERENCIAMENTO DE ESPAÇO EM TABLESPACES

Os tablespaces alocam espaço em extensões (extents). Estes podem ser criados para usar um dos dois métodos de controle de espaço livre e utilizado conforme destaca Legatti (2011):

Tablespaces gerenciados localmente: As extensões são gerenciadas no tablespace por bitmaps. Cada bitmap corresponde a um bloco ou a um grupo de blocos. Quando uma extensão é alocada ou liberada para reutilização, o servidor Oracle altera os valores do bitmap para mostrar o novo status dos blocos. A partir do Oracle 9i este gerenciamento local é o padrão.

Tablespaces gerenciados por dicionário: As extensões são gerenciadas pelo dicionário de dados. O servidor atualiza as tabelas apropriadas no dicionário de dados sempre que uma extensão é alocada ou desalocada.

Como citado anteriormente, o Oracle mantém um bitmap em cada arquivo de dados de um tablespace gerenciado localmente. Para criar um tablespace gerenciado localmente, se faz necessário a utilização da cláusula **EXTENT MANAGEMENT LOCAL** com o comando create tablespace.

O autor acrescenta que, os tablespaces gerenciados localmente auxiliam na redução do *overhead* de gerenciamento de espaço eliminando a necessidade de diversas gravações nas tabelas do dicionário de dados ou nos segmentos de

rollback, o que ocorre quando o espaço é gerenciado centralmente através do dicionário de dados.

4.3 ÁLGEBRA RELACIONAL

Conforme Navathe & Elmasri (2005), um modelo de dados abarca um conjunto de operações para manipulação do banco de dados, além claro, dos conceitos de modelo de dados para a definição das restrições e estrutura mesmo. O conjunto básico de operações para o modelo relacional é a **álgebra relacional**.

Os autores reforçam ainda, que estas operações possibilitam à um usuário especificar as solicitações básicas de recuperação. O resultado dessa recuperação consistirá numa nova relação, que pode ter sido formada a partir de uma ou mais relações. As operações de álgebra originam, assim, novas relações, que podem ser manipuladas adiante utilizando as operações de mesma álgebra.

A álgebra relacional é de suma importância, pois provê um fundamento formal para operações do modelo relacional, servindo de base para implementar e **otimizar** as consultas em sistemas de gerenciadores de banco de dados relacional (SGBDRs). Por fim, alguns de seus conceitos são incorporados na linguagem de consulta-padrão SQL para os SGBDRs.

4.3.1 OPERAÇÕES RELACIONAIS UNÁRIAS: SELEÇÃO (SELECT)

Segundo os autores Navathe & Elmasri (2005), a operação *SELECT* é utilizada para recuperar um subconjunto de tuplas de uma relação que satisfaça uma condição de seleção. Uma operação pode ser considerada *SELECT* quando o filtro mantém apenas as tuplas que satisfaçam uma condição de qualificação. Ainda conforme os autores Navathe & Elmasri (2005), a operação *SELECT* pode também ser visualizada como um particionamento horizontal da relação em dois conjuntos de tuplas — aquelas tuplas que satisfazem a condição e são selecionadas, e as tuplas que não satisfazem a condição e são descartadas.

4.3.2 OPERAÇÕES UNIÃO (UNION), INTERSEÇÃO (INTERSECTION) E SUBTRAÇÃO (MINUS)

Conforme Navathe & Elmasri (2005), inúmeros conjuntos de operações teóricas são utilizados para união dos elementos de dois conjuntos de diferentes

maneiras, incluindo UNIÃO, INTERSEÇÃO e DIFERENÇA DE CONJUNTO (também chamada de SUBTRAÇÃO). Os autores ainda sugerem que as operações citadas anteriormente são operações binárias, ou seja, cada uma delas é aplicada à dois conjuntos (de tuplas). Quando estas operações são convertidas à um banco de dados relacional, as duas relações nas quais qualquer uma dessas três operações for aplicada devem ter o mesmo tipo de tuplas; essa condição comumente tem sido chamada de compatibilidade de união.

4.3.3 OPERAÇÕES RELACIONAIS BINÁRIAS: (JUNÇÃO JOIN) E DIVISÃO

Conforme Navathe & Elmasri (2005), a operação JUNÇÃO, indicada por \bowtie , é utilizada para combinar as tuplas relacionadas em duas relações dentro de uma única tupla. Essa operação é de grande importância para qualquer banco de dados relacional com mais de uma relação, porque possibilita o processamento dos relacionamentos entre as relações.

Ainda conforme os autores, na JUNÇÃO, apenas as combinações de tuplas que satisfizerem a condição de junção aparecerão no resultado, enquanto que no produto cartesiano todas as combinações de tuplas serão compreendidas no resultado.

4.4 DEPURAÇÃO (Debug)

Segundo Macêdo (2012), um depurador é uma ferramenta para testar outros programas e fazer sua depuração, com o objetivo de encontrar anormalidades e falhas do programa. O autor ainda cita que para determinados tipos de problema existem ferramentas de análise do código fonte, que buscam por erros específicos no código, o que depende da linguagem de programação em uso.

4.5 TUNING DE BANCO DE DADOS

Conforme Ikematu (2009), a tradução literal de 'tuning' é uma sintonia ou ajuste de algo com o objetivo de uma melhora no seu desempenho. Um SGBD é um produto de software sofisticado e flexível, de forma que permita vários ajustes. Ajustes estes que podem vir a afetar a performance do banco de dados de forma que obtenha um bom desempenho ou não. Ainda segundo Ikematu (2009), por 'tuning' da base de dados, podemos entender como uma customização do sistema

feita sob medida para que a performance atenda melhor as suas necessidades.

4.5.1 TIPOS DE TUNING

O tuning pode ser realizado em diversos pontos do seu ambiente, como por exemplo uma melhora na infraestrutura de rede, uma melhora nos equipamentos de hardware. Ou, como estaremos abordando no referido trabalho, otimização de consultas SQL, entre diversas outras.

Zorzi, M.T. (2015) diz que podem ser aplicadas técnicas de tuning por exemplo no Sistema Operacional, a fim de eliminar gargalos e uma melhor utilização dos componentes de hardware. No Banco de Dados poderíamos dividir entre ajustes na arquitetura do banco de dados, otimizações de instruções e objetos SQL, bem como ajustes no projeto do Banco de Dados.

4.5.2 TÉCNICAS DE TUNING

Conforme Zorzi, M.T. (2015), o tuning se divide em três etapas, sendo elas:

- Entendimento do problema;
- Elaboração do diagnóstico;
- Aplicar as técnicas de tuning.

O autor ainda destaca que a aplicação das técnicas de *tuning*, é feita com base no diagnóstico elaborado na etapa anterior. Nessa etapa são aplicados os ajustes em cada componente que está apresentando gargalos, a fim de atingir o desempenho esperado.

4.5.3 UMA VISÃO GERAL DA SINTONIZAÇÃO DE BANCO DE DADOS EM SISTEMAS RELACIONAIS

Após o projeto e implementação de um Banco de Dados, ou seja, a partir do uso real das aplicações, das transações, das consultas e das visões são revelados fatores e áreas de problemas que podem não ter sido identificados durante o projeto físico inicial, afirmam Navathe & Elmasri (2005). A utilização dos recursos, bem como o processamento interno do SGBD (otimização de consultas por exemplo) podem ser monitorados à fim de identificar gargalos, tais como a disputa pelos mesmos dados ou dispositivos. Com isso, os volumes de atividades e os tamanhos

dos dados podem ser mais bem estimados. Portanto, é necessário monitorar e revisar o projeto físico do banco de dados constantemente. Os objetivos da sintonização são os seguintes:

- Fazer com que as aplicações sejam executadas mais rapidamente.
- Diminuir o tempo de resposta de consultas/transações.
- Melhorar o desempenho geral das transações.

4.5.4 SINTONIZAÇÃO DE ÍNDICES

De acordo com Navathe & Elmasri (2005), a escolha inicial de índices pode precisar de uma revisão pelos seguintes motivos:

- As consultas podem ter um custo de tempo alto para ser executadas por conta da ausência de um índice.
- Certos índices podem não ser utilizados.
- Certos índices podem estar ocasionando algum tipo de sobrecarga porque são baseados em um atributo que sofre sucessivas alterações.

Os autores ainda citam que, a maioria dos SGBDs possuem um comando ou um meio de rastreamento (*trace facility*) que pode ser usado pelo DBA para solicitar que o sistema mostre como uma consulta foi executada, quais operações foram realizadas e em qual ordem e quais estruturas de acesso secundário foram utilizadas. Através da análise desses planos de execução, é possível diagnosticar as causas dos problemas citados anteriormente. Alguns índices podem ser excluídos e alguns novos índices podem ser incluídos com base na análise de sintonização.

Segundo Navathe & Elmasri (2005, p.388):

O objetivo da sintonização é avaliar dinamicamente os requisitos, os quais às vezes variam sazonalmente ou durante diferentes períodos do mês ou da semana, e reorganizar os índices para proporcionar melhor desempenho geral.

Os autores afirmam ainda que, a exclusão e a criação de novos índices são uma sobrecarga que pode ser justificada em consequência das melhorias de desempenho. A atualização de uma tabela geralmente é suspensa enquanto um índice estiver sendo excluído ou criado; deve-se contabilizar essa perda de serviço. Além da exclusão ou da criação de índices e a mudança a partir de um índice que não é *clustering* para um índice *clustering* e vice-versa, a **reconstrução** do índice pode melhorar o desempenho. A grande maioria dos SGBDRs usam árvores-B em

seus índices. Caso haja muitas exclusões na chave do índice, páginas do índice poderão ter espaço desperdiçado, o que pode ser recuperado ao longo da operação de reconstrução. Em contrapartida, inclusões em excesso podem ocasionar *overflow* em um índice *clustering* afetando seu desempenho. A reconstrução de um índice *clustering* corresponde a reorganizar a tabela ordenada segundo aquela chave.

4.5.5 SINTONIZAÇÃO DO PROJETO DE BANCO DE DADOS

Uma vez que um projeto físico de banco de dados não atinja os objetivos esperados, pode-se voltar ao projeto lógico do banco de dados realizando ajustes no esquema lógico, remapeando-o num novo conjunto de tabelas e índices físicos afirmam Navathe & Elmasri (2005).

Ainda, conforme destacado pelos autores, todo o projeto do banco de dados deve ser direcionado pelos requisitos de processamento e pelos requisitos dos dados. Caso os requisitos de processamento se alterarem dinamicamente, o projeto precisa responder por meio de alterações no esquema conceitual, caso se faça necessário, e refletir as mudanças no esquema lógico e no projeto físico.

4.5.6 SINTONIZAÇÃO DE CONSULTAS

Percebe-se como o desempenho de consultas é dependente da seleção apropriada de índices e como eles precisam ser sintonizados mediante análise das consultas, realizadas através do plano de execução da consulta. Navathe & Elmasri (2005) afirmam que há principalmente duas indicações que sugerem que a sintonização da consulta pode ser necessária:

1. Uma consulta resulta em muitos acessos a disco (por exemplo, uma consulta de correspondência exata que percorre uma tabela inteira);
2. Um plano de consulta mostra que os índices relevantes não estão sendo utilizados.

Segundo Navathe & Elmasri (2005, p.389), existem algumas situações típicas que indicam a necessidade da sintonização de consultas:

1. Muitos otimizadores de consulta não usam índices na presença de expressões aritméticas (tais como $SALÁRIO/365 > 10,50$), de comparações numéricas de atributos de diferentes tamanhos e níveis de precisão (tais como $QTDDE_A = QTDDE_B$, onde $QTDDE_A$ é do tipo

INTEGER e QTDE_B é do tipo SMALLINTEGER), comparações COM NULL (tais como DATANASC IS NULL) , e comparações com *substrings* (tais como UNOME LIKE "%MANN").

2. Frequentemente os índices não são usados em consultas aninhadas usando IN; por exemplo, a consulta **SELECT SSN FROM EMPREGADO**

WHERE NRD IN (SELECT NUMEROD FROM DEPARTAMENTO WHERE GERSSN = '333445555');

pode não usar o índice para NRD em EMPREGADO, enquanto o uso de NRD = NUMEROD na cláusula WHERE com uma consulta em um único bloco pode gerar o índice a ser usado.

3. Algumas cláusulas DISTINCT podem ser redundantes e podem ser evitadas sem a modificação do resultado. Um DISTINCT frequentemente causa uma operação de ordenação e deve ser evitado sempre que possível.

4. O uso desnecessário de tabelas de resultado temporário pode ser evitado por meio da aglutinação de múltiplas consultas em uma única consulta, *a menos* que a relação temporária seja necessária para algum processamento intermediário.

5. Em algumas situações que envolvem o uso de consultas correlatas, os temporários são úteis. Considere a consulta:

```
SELECT SSN  
FROM EMPREGADO E  
WHERE SALÁRIO = SELECT MAX (SALÁRIO)  
FROM EMPREGADO AS G  
WHERE G.NRD = E.NRD;
```

Ela possui o perigo potencial de pesquisar toda a tabela EMPREGADO G interna para cada tupla da tabela EMPREGADO E externa.

Para torná-la mais eficiente, ela pode ser quebrada em duas consultas, na qual a primeira apenas calcula o máximo salário em cada departamento conforme segue:

```
SELECT MAX (SALÁRIO) AS MAIORSALARIO, NRD INTO TEMP FROM EMPREGADO  
GROUP BY NRD;  
SELECT SSN  
FROM EMPREGADO, TEMP  
WHERE SALÁRIO = MAIORSALARIO AND EMPREGADO.NRD = TEMP.NRD;
```

6. Se múltiplas opções de condições de junção são possíveis, escolha uma que use um índice *clustering* e evite as que contenham comparações de cadeias de caracteres. Por exemplo, supondo que o atributo NOME é uma chave candidata em EMPREGADO e ALUNO, é melhor usar EMPREGADO.SSN = ALUNO.SSN como uma condição de junção em vez de

EMPREGADO.NOME : ALUNO.NOME se SSN possuir um índice *clustering* em uma ou ambas as tabelas.

7. Uma idiossincrasia dos otimizadores de consulta é que a ordem das tabelas na cláusula FROM pode afetar o processamento de junções. Se esse for o caso, pode-se precisar alterar essa ordem de forma que a menor das duas relações seja varrida e a maior relação seja usada com um índice adequado.

8. Alguns otimizadores de consulta têm um desempenho pior em consultas aninhadas em comparação com suas consultas não aninhadas correspondentes. Há quatro tipos de consultas aninhadas:

- Subconsultas não correlatas com agregações na consulta interna.
- Subconsultas não correlatas sem agregações.
- Subconsultas correlatas com agregações na consulta interna.
- Subconsultas correlatas sem agregações.

Dos quatro tipos acima, o primeiro geralmente não apresenta problemas, uma vez que a maioria dos otimizadores de consulta avalia a consulta interna uma vez. Entretanto, para uma consulta do segundo tipo, tal como o exemplo n^o 2 acima, a maioria dos otimizadores pode não utilizar um índice para NRD em EMPREGADO. OS mesmos otimizadores podem fazê-lo se a consulta for escrita como uma consulta não aninhada. A transformação de subconsultas correlatas pode envolver o estabelecimento de tabelas temporárias.

9. Finalmente, muitas aplicações são baseadas em visões que definem os dados de interesse para aquelas aplicações. Às vezes, essas visões se tornam excessivas, porque uma consulta pode ser aplicada diretamente sobre uma tabela base em vez de ser realizada sobre uma visão que é definida por uma junção.

4.5.7 ORIENTAÇÕES ADICIONAIS PARA A SINTONIZAÇÃO DE CONSULTAS

1. Uma consulta com múltiplas condições de seleção que são conectadas por meio de OR podem não estar apontando ao otimizador de consultas para usar um determinado índice. Tal consulta poderá ser dividida e expressa como uma união de consultas, cada uma com uma condição para um atributo que tenha um índice a ser utilizado.

2. Para auxiliar na redução de custo de tempo de uma consulta, podemos efetuar as seguintes transformações:

- A condição NOT pode ser transformada em uma expressão afirmativa.

- Blocos SELECT embutidos utilizando IN, = ALL, = ANY e = SOME podem ser substituídos por junções.
- Se uma junção de igualdade for definida entre duas tabelas, o predicado de faixa (condição de seleção) para o atributo de junção definido em uma tabela pode ser repetido para a outra tabela.
- Condições WHERE podem ser reescritas para utilizar os índices para múltiplas colunas.

5. ESTADO DA ARTE

A motivação pela escolha do tema deu-se a partir da busca pela melhoria no desempenho de execução das queries a partir do ERP, e, posteriormente nas *views* que serão executadas através da ferramenta de Business Intelligence - BI. Seguindo esta linha foram encontrados trabalhos que abordam diferentes práticas de *tuning*, como exemplo temos a “**Avaliação de Desempenho de Ferramentas para Tuning em Banco de Dados**” que é um trabalho datado de 2012/2013 e testa as principais ferramentas de auxílio no ajuste de desempenho dos principais SGBD's, com consultas SQL pré estabelecidas, à fim de sugerir melhorias de processamento que trouxessem um ganho de desempenho significativo.

Outro trabalho é o “**Estudo Da Metodologia De Tuning Em Banco De Dados Oracle**”, que é datado de 2012 e trata da metodologia e técnicas de *tuning*, bem como sua otimização, apresentando um estudo de caso com base em um SGBD Oracle 11g, com propostas para instruções DML dentro do Banco de Dados, implementação e testes. Os resultados obtidos apresentam situações de melhora ou degradação no desempenho de várias funcionalidades do software.

O *tuning* pode ser aplicado em diversas etapas de desenvolvimento e implementação do Banco de Dados, bem como pode ser realizado um ajuste fino no Sistema Operacional ou Hardware. Neste campo de pesquisa foi encontrado um trabalho que traz uma abordagem diferente quando se trata de otimização, com o título de “**Tuning de Sistema Operacional em Banco de Dados Oracle**”, é um trabalho de 2015 que tem como objetivo realizar o processo de *tuning* utilizando as melhores práticas recomendadas pela Red Hat para o sistema operacional Linux, servindo como base de execução o banco de dados Oracle. Além dos parâmetros de configuração do Linux, o trabalho aponta quais parâmetros e configurações podem

ser realizados no banco de dados Oracle para obtenção do melhor desempenho.

Por fim, foi localizado um artigo da FATEC, que não possui data de publicação cujo título “**Tuning em queries em Banco de Dados**”, aborda através da ferramenta Database Engine Tuning Advisor, inúmeras técnicas para otimização de banco de dados, que conforme os autores especificam, houve uma melhoria considerável no tempo de execução das consultas.

6. DESENVOLVIMENTO

6.1 VISÃO GERAL

É proposto a aplicação de técnicas de tuning de Banco de Dados, cujo objetivo é obter uma melhora no tempo de execução das consultas, analisando a depuração e, caso se faça necessário, reescrever as queries, desfazer, criar ou recriar os índices das tabelas relacionadas.

6.2 CONFIGURAÇÃO DO AMBIENTE

O servidor de banco de dados utilizado consiste numa arquitetura x86_64, possui um processador Intel® Xeon® CPU E5-2620 v3 com 2.40 GHz e memória RAM com um total de 62Gb e conta com o sistema operacional Linux. Devido a grande quantidade de partições, segue informações pertinentes ao armazenamento do servidor de banco de dados conforme figura abaixo:

```
Disk /dev/sda: 805.3 GB, 805306368000 bytes, 1572864000 sectors
/dev/sda1 *          2048      1026047      512000      83  Linux
/dev/sda2          1026048  1572863999  785918976    8e  Linux LVM
Disk /dev/sdb: 322.1 GB, 322122547200 bytes, 629145600 sectors
/dev/sdb1 *          2048      1026047      512000      83  Linux
/dev/sdb2          1026048  629145599   314059776    8e  Linux LVM
Disk /dev/mapper/ol-root: 53.7 GB, 53687091200 bytes, 104857600 sectors
Disk /dev/mapper/ol-swap: 25.4 GB, 25367150592 bytes, 49545216 sectors
Disk /dev/mapper/ol-home: 725.7 GB, 725656535040 bytes, 1417297920 sectors
Disk /dev/mapper/ol100-swap: 25.4 GB, 25367150592 bytes, 49545216 sectors
Disk /dev/mapper/ol100-home: 242.5 GB, 242472714240 bytes, 473579520 sectors
Disk /dev/mapper/ol100-root: 53.7 GB, 53687091200 bytes, 104857600 sectors
```

Figura 6. Informações referente ao armazenamento e disco do servidor de Banco de Dados. FONTE – Autor.

Já o servidor de aplicação (onde rodam todos os sistemas e BI), possui as especificações conforme figura abaixo:

Sistema

Processador:	Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz 2.40 GHz (2 processadores)
Memória instalada (RAM):	36,0 GB
Tipo de sistema:	Sistema Operacional de 64 bits, processador com base em x64
Caneta e Toque:	Suporte a Toque Limitado com 10 Pontos de Toque

Figura 7. Informações referentes ao Sistema Operacional do servidor de Aplicação.
FONTE – Autor.

O servidor de aplicação possui 2 partições, sendo elas: Disco Local (C:) com 599Gb, sendo 69.8Gb livres e o Disco Local (D:) com 499Gb, sendo 477Gb livres.

6.3 QUESTIONÁRIO (A FIM DE IDENTIFICAR OS PONTOS DE GARGALO)

O questionário abaixo, obteve resposta de 29 usuários de diversos setores da empresa, levando em consideração que alguns usuários de setores chave não responderam ao questionário por motivos até então desconhecidos.

Este questionário foi realizado através da plataforma Google Forms no período de 30/07/2018 à 10/08/2018, segue abaixo o resultado:

Você percebe algum tipo de lentidão no sistema (ERP Logix, BI, relatórios, Gecex)?

29 respostas

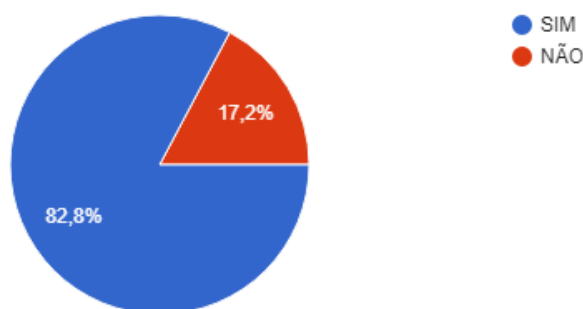


Figura 8. Totalização referente à percepção de lentidão. FONTE – Autor.

Em seguida, foi realizada uma pergunta à fim de identificar qual sistema, bem como quais rotinas existia possível lentidão, conforme figura abaixo:

Se a resposta anterior foi sim, selecione (pode ser mais de 1 opção) em quais rotinas você percebe essa lentidão:

29 respostas

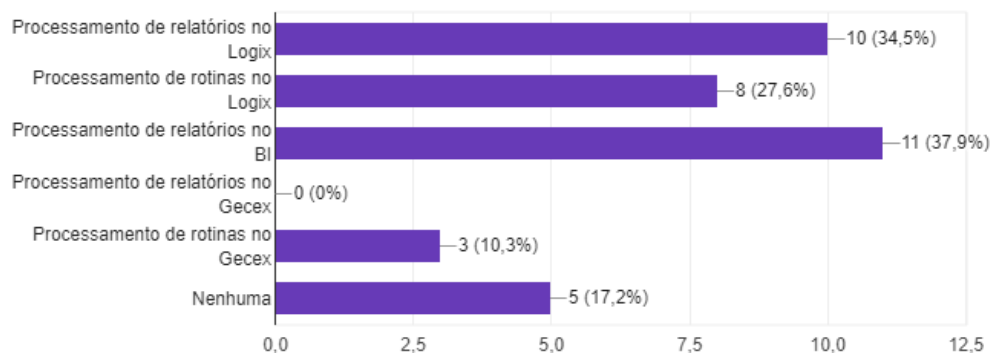


Figura 9. Totalização de Sistemas/Rotinas. FONTE – Autor.

Na sequencia o usuário deveria informar seu setor, à fim de identificar os módulos com maior congestionamento de dados:

Selecione qual o seu setor:

29 respostas

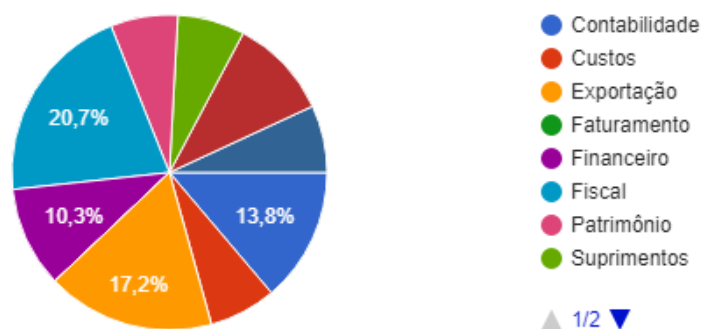


Figura 10. Totalização por setor. FONTE – Autor.

Todavia, as informações obtidas com o questionário acima foram insuficientes, visto que a percepção da lentidão é subjetiva e pode não ser precisa. Contudo, foi elaborado um novo questionário composto de 9 perguntas, à fim de obter informações pertinentes à qual sistema com maior utilização pelo usuário,

setor em que atua, se o mesmo atua na matriz ou filiais, qual o dispositivo utilizado para acesso ao sistema, tipo de rede utilizada, percepção de lentidão e se a mesma ocorre em horário e período do mês em específico.

Como citado acima, alguns usuários não responderam ao primeiro questionário e com isso, buscou-se saber o motivo desta não participação. Conversando com estes usuários, constatou-se que alguns não responderam devido à bloqueios no firewall das filiais, portanto os mesmos não obtiveram acesso ao link do formulário do google. Pensando nesta situação, foi elaborado um novo questionário no editor de texto word (salvo em 2 formatos, .doc e .txt para usuários que não dispunham de uma ferramenta da suíte office ou libreoffice) e enviado por email para os mesmos 33 usuários do questionário anterior no período de 04/09/2018 à 06/09/2018. Devido ser um período do mês de muita demanda de trabalho (fechamento fiscal, contábil) e com o feriado de 07 de setembro nesta mesma semana, foram obtidas 17 respostas neste período, e, embora fugindo do cronograma, será aceito o formulário preenchido até 14/09/2018 por ser de suma importância a opinião dos usuários, e logo após efetuada a tabulação dos dados. Contudo, caso não seja obtido o *feedback* esperado, serão adotadas de 1 a 2 consultas SQL de cada módulo do sistema/BI para análise.

Neste novo questionário apenas 4 usuários distintos perceberam certa lentidão, dos quais em contato individual identificou-se 8 relatórios do BI para posterior análise.

7. CENÁRIO ATUAL

Analisando diversos arquivos de logs das principais rotinas do sistema, constatou-se que são utilizados comandos sql específicos para cada função e/ou retorno de informação, com isso, comandos de *include*, *update*, *delete* e até mesmo *select* não levam tanto tempo para serem executados pois não são muitos complexos e em diversos casos, com relação aos comandos *select* são “quebradas” em diversas consultas.

Com isso, decidiu-se apenas focar nas análises do BI, sendo que ao rodar os comandos do ERP extraídos através de *debug*, percebeu-se que o seu tempo de execução era satisfatório, pois comandos complexos eram subdivididos em vários outros comandos.

Baseando-se no questionário, identificou-se 3 usuários distintos que têm algum tipo de lentidão no processar os relatórios. Com essa informação, foi identificado junto aos usuários 8 relatórios do BI que foram analisados.

7.1 ANÁLISE 1: CONHECIMENTOS DE FRETE

Um dos relatórios analisados foi obtido junto ao setor de tributos, que hoje é utilizado para conferência de valores de impostos de pis e cofins. Este relatório, como é utilizado para conferência, geralmente é utilizado com filtros de conhecimentos de frete do mês corrente, fazendo com que o seu processamento ao fim do mês retorne um número maior de registros.

Esse relatório, como está montado hoje utiliza-se de uma tabela mestre onde constam informações pertinentes ao conhecimento de frete, bem como tabelas relacionadas que contem informações de notas fiscais de entrada, de saída e relacionamentos entre conhecimentos de frete e notas fiscais.

No caso dos conhecimentos de frete, ao emitir uma nota fiscal de saída ou entrada, é emitido este documento a fim de acompanhar a carga até o seu destino. No caso da Sudati, também é dado entrada em conhecimentos de frete que não possuem relacionamentos com notas fiscais, fazendo com que não existam informações na tabela SUP_FRETE_X_NF_ENTRADA.

Analisando a *query* do atual relatório, a mesma foi montada com 3 *selects* distintos, sendo eles: fretes de entrada, fretes de saída e os fretes relacionados.

Na querie atual, foi identificado que haviam campos em duplicidade que no momento do *tuning* foram eliminados, e problemas no relacionamento entre as tabelas, fazendo com que a leitura de todos os registros para aplicar os filtros passados no parâmetro *where* fossem um número muito maior.

Um exemplo dessa falha no relacionamento se dá no uso incorreto de um *left join*, uma vez que a maioria dos relacionamentos obriga a se ter um relacionamento entre 2 tabelas diferentes, ou seja, o que contenha em tabela A, deverá conter na tabela B, sendo que o uso do *left join* retornaria tudo o que contesse em A, e não apenas o que resultasse da junção entre a A e B.

Para resolver esse problema, foi utilizado o uso apenas do *join*, que retorna apenas a junção entre a A e B, retornando um número bem menor de registros.

Ex: Toda nota fiscal deverá ter um item, logo o relacionamento entre nf x item deverá ser verdadeiro, pois não existiria uma nota fiscal sem item.

Outros problemas encontrados na *query* atual, foram que argumentos passados no parâmetro *where*, faziam com que a consulta fizesse uma varredura em registros que não seriam utilizados no relatório, como por exemplo:

“Para a conferência, são utilizados apenas conhecimentos e notas fiscais com situação normal, ou seja, não há necessidade de fazer a leitura de documentos cancelados”. Nesse caso, a validação da situação da nota fiscal e conhecimento de frete, foi retirado da cláusula *where* e foi inserido nos parâmetros de relacionamento no próprio *join*, fazendo com que por exemplo, num total de 100 registros apenas 10 tivessem situação normal, apenas 10 registros estão sendo lidos agora, melhorando consideravelmente o tempo de execução com um numero grande de registro ao final do mês.

Outra situação que se apresentou nesse relatório, foi a duplicação de conhecimentos de frete retornados, pois no *select* onde se faz a leitura dos fretes de entrada, não se distingue dos que possuem relacionamento com a nota fiscal de entrada. Nesse sentido, fez-se uso do *not exists* para validar os conhecimentos lançados a vulso, dos que possuem relacionamento.

Porém, quando foi efetuada essa validação, verificou-se junto ao pessoal do setor de tributos que alguns conhecimentos de frete não constavam no relatório. Analisando esta situação, foi identificado que alguns conhecimentos de frete não estavam sendo relacionados com a sua respectiva nota fiscal no momento de seu lançamento, embora estes possuíssem uma nota fiscal “relacionada” perante a receita federal. Com isso, a partir de então foi atentado para esta situação e corrigidas estas falhas.

Percebeu-se também que na *query* atual se fazia uso do *decode* e *case*. No trabalho foi adotado o uso apenas do *case* para manter um certo padrão, porém não se notou alteração nos tempos de execução da *query*.

Ao final, com a aplicação das técnicas de *tuning* e a reescrita das *queries*, obteve-se um número menor de registros retornados e de colunas retornadas no relatório enxugando as informações e facilitando a sua leitura no momento da conferência. Consequentemente teve um número menor de registros lidos e realizadas as aplicações dos filtros, logo seu tempo de execução foi menor do que com a *querie* atual.

Segue abaixo antes e depois do *tuning*:

Atualmente:

```

FROM FRETE_SUP A
  LEFT JOIN NF_SUP B ON (B.COD_EMPRESA = A.COD_EMPRESA
    AND B.COD_TRANSPOR = A.COD_TRANSPOR
    AND B.NUM_CONHEC = A.NUM_CONHEC
    AND B.SER_CONHEC = A.SER_CONHEC
    AND B.SSR_CONHEC = A.SSR_CONHEC )

  RIGHT JOIN AVISO_REC C ON (C.COD_EMPRESA = B.COD_EMPRESA
    AND C.NUM_AVISO_REC = B.NUM_AVISO_REC )

  LEFT JOIN FORNECEDOR E ON (E.COD_FORNECEDOR =
B.COD_TRANSPOR)

  LEFT JOIN SUP_PAR_AR F ON (F.EMPRESA = C.COD_EMPRESA
    AND F.AVISO_RECEBTO = C.NUM_AVISO_REC
    AND F.SEQ_AVISO_RECEBTO = C.NUM_SEQ
    AND F.PARAMETRO IN ('COD_CST_COFINS_FRT') )

WHERE A.TIP_FRETE = 'C'

```

Tempo de execução: 10,218 segundos

Após o tuning:

```

FROM FRETE_SUP A
  JOIN NF_SUP B ON (B.COD_EMPRESA = A.COD_EMPRESA
    AND B.COD_TRANSPOR = A.COD_TRANSPOR
    AND B.NUM_CONHEC = A.NUM_CONHEC
    AND B.SER_CONHEC = A.SER_CONHEC
    AND B.SSR_CONHEC = A.SSR_CONHEC )

  JOIN AVISO_REC C ON (C.COD_EMPRESA = B.COD_EMPRESA
    AND C.NUM_AVISO_REC = B.NUM_AVISO_REC )

  JOIN FORNECEDOR E ON (E.COD_FORNECEDOR = B.COD_TRANSPOR)

```

```

JOIN SUP_PAR_AR F ON (F.EMPRESA      = C.COD_EMPRESA
                      AND F.AVISO_RECEBTO  = C.NUM_AVISO_REC
                      AND F.SEQ_AVISO_RECEBTO = C.NUM_SEQ
                      AND F.PARAMETRO      IN ('COD_CST_COFINS_FRT') )
WHERE A.TIP_FRETE      = 'C'
AND NOT EXISTS (
    SELECT * FROM SUP_FRETE_X_NF_ENTRADA REL
    WHERE A.COD_EMPRESA = REL.COD_EMPRESA
    AND A.NUM_CONHE=REL.NUM_CONHEC
    AND A.SER_CONHEC=REL.SER_CONHEC
    AND A.COD_TRANSPOR=REL.COD_TRANSPOR
)

```

OBSERVAÇÃO: As queries completas encontram-se no apêndice.

Tempo de execução: 4,219 segundos

7.2 ANÁLISE 2: FATURAMENTO

Outro relatório analisado foi o de informações pertinentes ao faturamento da empresa, obtido juntamente ao setor de faturamento da filial de Otacílio Costa (MDF).

Este relatório é utilizado para exibir o faturamento da empresa em um determinado período, de um determinado cliente, de uma determinada família de itens, entre outros.

Como podemos perceber é um relatório bastante volátil, fazendo com que certas execuções (principalmente um período de tempo superior à um trimestre) sejam impossíveis de serem executadas pela ferramenta de BI, ocasionando a queda do serviço.

Este relatório é composto de 4 consultas diferentes (notas de venda, devolução, notas com entrega futura e notas de complemento de valor).

Assim como o relatório anterior, este também apresenta falhas nos relacionamentos entre as tabelas, quebras de campos indevidos, utilização de parâmetros na cláusula *where* ocasionando uma leitura de registros desnecessários.

Então para resolução do problema desta *query*, foram unificados campos de data (que atualmente estavam dispostos de 3 colunas):

Atualmente:

```
YEAR(A.DAT_HOR_EMISSAO) ANO,  
MONTH(A.DAT_HOR_EMISSAO) MES,  
TRUNC(A.DAT_HOR_EMISSAO) DAT_HOR_EMISSAO,
```

Após o tuning:

```
TRUNC(A.DAT_HOR_EMISSAO) DAT_HOR_EMISSAO,
```

Também foi revisto o relacionamento entre as tabelas, onde se estava fazendo uso de *left join*, visto que para se obter a informação do faturamento temos 2 elementos base: pedido e nota fiscal. Com isso, percebe-se a obrigatoriedade da junção apenas de pedidos que possuam nota fiscal relacionada e, esta nota fiscal não esteja cancelada. Nesse mesmo contexto, percebeu-se também a utilização de parâmetros na cláusula *where* (situação da nota fiscal e estatística da natureza de operação) que foram adicionados como filtros na própria junção das tabelas, evitando assim a leitura de registros de notas canceladas (para validação da situação) bem como a estatística da natureza de operação.

Segue abaixo antes e depois do *tuning*:

Atualmente:

```
FROM FAT_NF_MESTRE A
```

```
LEFT JOIN FAT_NF_ITEM B
```

```
ON A.EMPRESA = B.EMPRESA
```

```
AND A.TRANS_NOTA_FISCAL = B.TRANS_NOTA_FISCAL
```

```
LEFT JOIN NAT_OPERACAO C
```

```
ON A.NATUREZA_OPERACAO = C.COD_NAT_OPER
```

WHERE

A.SIT_NOTA_FISCAL = 'N'

AND C.IES_ESTATISTICA <> 'N'

Tempo de execução: 53,187 segundos

Após o *tuning*:

FROM FAT_NF_MESTRE A

JOIN FAT_NF_ITEM B

ON A.EMPRESA = B.EMPRESA

AND A.TRANS_NOTA_FISCAL = B.TRANS_NOTA_FISCAL

AND A.SIT_NOTA_FISCAL='N'

JOIN NAT_OPERACAO C

ON A.NATUREZA_OPERACAO = C.COD_NAT_OPER

AND C.IES_ESTATISTICA <> 'N'

AND A.SIT_NOTA_FISCAL='N'

Tempo de execução: 20,828 segundos

Com a retirada da situação da nota fiscal da cláusula *where*, efetuamos a listagem apenas das notas com situação normal dentro deste SQL que como no exemplo anterior, reduzimos a quantidade de registros retornados e lidos.

Também foi adotado a utilização do *UNION ALL*, que segundo alguns autores têm um desempenho melhor pois faz a união sem realizar qualquer tipo de identificação de registros duplicados, visto que foi utilizado o comando *distinct* em todas as consultas, bem como os filtros à fim de eliminar a possibilidade de duplicações.

OBSERVAÇÃO: As queries completas encontram-se no apêndice.

7.3 ANÁLISE 3: NOTAS DE ENTRADA

O relatório de notas de entrada analisado, geralmente é utilizado para conferência nos fechamentos dos setores de tributos e fiscal, e consiste basicamente em informações pertinentes a notas fiscais de entrada, especificamente englobando a parte de impostos onde poderíamos citar ICMS, IPI, PIS E COFINS.

Atualmente, este relatório consiste nos mesmos problemas dos relatórios anteriores com o uso do *left join*, que neste caso em específico, quando se dá a entrada da nota fiscal, a mesma deve possuir seus respectivos itens e ao final é gerado um documento chamado “aviso de recebimento”, portanto notas válidas para este relatório, bem como sua conferência, devem ser notas válidas (e com isso devem possuir itens e aviso de recebimento).

Outro problema encontrado, foi a utilização de uma segunda *view* que realiza a conversão do formato de data, o que foi julgado desnecessário, uma vez que o campo de data de emissão e data de entrada da nota fiscal atendem os requisitos de formato utilizado pelo usuário.

Para a resolução do problema desta *query*, foram retiradas as junções *left join* em tabelas que o relacionamento é obrigatório, e por final foi retirado a junção com a *view* de formato de data.

Segue abaixo antes e depois do *tuning*:

Atualmente:

```
FROM NF_SUP A
LEFT JOIN AVISO_REC B
ON (B.COD_EMPRESA = A.COD_EMPRESA AND B.NUM_AVISO_REC =
A.NUM_AVISO_REC)

LEFT JOIN ITEM_SUP IT ON IT.COD_EMPRESA = B.COD_EMPRESA
AND IT.COD_ITEM = B.COD_ITEM

JOIN EIS_DIMENSAO_DAT DT
ON (TRUNC(A.DAT_ENTRADA_NF) = DT.DAT_REFER)
```

Tempo de execução: 13,515 segundos

Após o tuning:

```
FROM NF_SUP A
```



```
JOIN AVISO_REC B
  ON (B.COD_EMPRESA = A.COD_EMPRESA AND B.NUM_AVISO_REC =
A.NUM_AVISO_REC)
```

```
JOIN ITEM_SUP IT ON IT.COD_EMPRESA = B.COD_EMPRESA -->>
      AND IT.COD_ITEM = B.COD_ITEM
```

```
JOIN FORNECEDOR C -->>
  ON (C.COD_FORNECEDOR = A.COD_FORNECEDOR)
```

Tempo de execução: 8 segundos

OBSERVAÇÃO: As queries completas encontram-se no apêndice.

7.4 ANÁLISE 4: MOVIMENTAÇÃO ESTOQUE

O relatório de movimentação de estoque analisado consiste em informações pertinentes ao setor de custos e manufatura, e geralmente é utilizado para conferência de saídas e produção de itens da empresa.

Vendo este relatório em um primeiro momento, ele apresenta as mesmas falhas dos relatórios anteriores no quesito relacionamento de tabelas. Porém, analisando mais a fundo, percebe-se que há alguns relacionamentos existentes na *view* que não estão sendo utilizados, sendo assim foram eliminados.

Outra situação, é que não estava sendo validado a situação atual do item, sendo que na *view* atual, eram lidos itens cancelados e inativos, fazendo com que houvesse uma sobrecarga e uma utilização de recursos maior.

Então como solução, foi incluída uma validação na própria junção da tabela de itens, alteradas as junções *left join* para apenas *join* e eliminados 2 relacionamentos de tabelas que não estavam sendo utilizadas.

Segue abaixo antes e depois do *tuning*:

Atualmente:

```
FROM ESTOQUE_TRANS A
  LEFT JOIN ITEM B
    ON A.COD_EMPRESA = B.COD_EMPRESA
    AND A.COD_ITEM = B.COD_ITEM
```

LEFT JOIN EMPRESA C

ON A.COD_EMPRESA = C.COD_EMPRESA

LEFT JOIN ESTOQUE_OPERAC D

ON A.COD_EMPRESA = D.COD_EMPRESA

AND A.COD_OPERACAO = D.COD_OPERACAO

LEFT JOIN LOCAL LOCAL_ORIG

ON A.COD_LOCAL_EST_ORIG = LOCAL_ORIG.COD_LOCAL

AND A.COD_EMPRESA = LOCAL_ORIG.COD_EMPRESA

LEFT JOIN LOCAL LOCAL_DEST

ON A.COD_LOCAL_EST_DEST = LOCAL_DEST.COD_LOCAL

AND A.COD_EMPRESA = LOCAL_DEST.COD_EMPRESA

LEFT JOIN ESTOQUE_OBS OBS

ON A.COD_EMPRESA = OBS.COD_EMPRESA

AND A.NUM_TRANSAC = OBS.NUM_TRANSAC

LEFT JOIN FAMILIA FAM

ON B.COD_FAMILIA = FAM.COD_FAMILIA

AND B.COD_EMPRESA = FAM.COD_EMPRESA

Tempo de execução: 1,687 segundos

Após o tuning:

FROM ESTOQUE_TRANS A

JOIN ITEM B

ON A.COD_EMPRESA = B.COD_EMPRESA

AND A.COD_ITEM = B.COD_ITEM

AND B.IES_SITUACAO='A'

JOIN ESTOQUE_OPERAC D

```
ON A.COD_EMPRESA = D.COD_EMPRESA  
AND A.COD_OPERACAO = D.COD_OPERACAO
```

```
JOIN ESTOQUE_OBS OBS  
ON A.COD_EMPRESA = OBS.COD_EMPRESA  
AND A.NUM_TRANSAC = OBS.NUM_TRANSAC
```

```
JOIN FAMILIA FAM  
ON B.COD_FAMILIA = FAM.COD_FAMILIA  
AND B.COD_EMPRESA = FAM.COD_EMPRESA
```

Tempo de execução: 0,5 segundos

OBSERVAÇÃO: As queries completas encontram-se no apêndice.

7.5 DEMAIS RELATÓRIOS

Foram apresentados acima 4 dos 8 relatórios onde foram aplicadas as técnicas de *tuning*, visto que destes outros 4 relatórios alguns deles não obtiveram melhoras nos resultados pois após avaliadas as *views* envolvidas, as mesmas estavam escritas corretamente e sua lentidão se dá pelo número muito grande de registros retornados (sendo o caso de 3 relatórios do setor contábil).

Os relatórios do setor contábil, como mencionado acima, são relatórios muito específicos que requerem diversos tipos de cálculo em algumas colunas e por vezes, retornam um número muito grande de registros devido ao alto número de lançamentos contábeis que são efetuados, e, como a empresa possui diversas filiais este número fica cada vez maior conforme o decorrer do mês.

Um outro relatório que é utilizado tanto pelo setor de tributos, bem como pelo setor de faturamento é pertinente às notas fiscais de saída. Como foram aplicadas as técnicas de *tuning* já na *view* de faturamento, não se fez necessário a aplicação na parte de notas de saída pois ambos os relatórios utilizam a mesma *view*. O motivo por ser dois relatórios diferentes se dá por questão de restrições de acesso, sendo que o relatório utilizado pelo setor de faturamento é utilizado na filial MDF, e o utilizado pelo tributos faz parte da matriz Compensados.

CONCLUSÃO

O cuidado com o desempenho de um Banco de Dados trata-se de um tema atual e de extrema relevância, porém, pouco se é explorado dentro desta área, visto que a maioria do material bibliográfico para pesquisa é escasso na língua portuguesa, fazendo com que muitas vezes o administrador de Banco de Dados busque outras soluções paliativas para resolver determinados problemas. Sabe-se que é indispensável o profissional da área de tecnologia da informação ter conhecimento do básico da língua inglesa, porém alguns profissionais não dispõem de tempo/oportunidade para tal, o que não será julgado no presente trabalho.

Em virtude do grande volume de dados que são gerados com os sistemas de informação atualmente, perante à necessidade de um tempo de resposta cada vez mais rápido faz-se necessário a elaboração de um código limpo que retorne as informações com um custo de tempo considerável.

Pode-se observar através deste estudo referente ao *tuning* em *queries* de Banco de Dados Oracle, que a aplicação destas técnicas é de suma importância desde a sua concepção e até mesmo a sua manutenção posteriormente, visando a melhora no seu tempo de execução, desde que haja um monitoramento prévio ou qualquer outra forma de avaliar e diagnosticar a lentidão, visto que para se obter a informação através do usuário notou-se um grande problema para o trabalho, pois dos 2 questionários enviados o *feedback* foi relativamente pequeno.

Para que se obtenha um retorno de informações em um tempo de execução considerável, faz-se necessário atentar para os relacionamentos corretos entre as tabelas elaborando um plano de consulta, a fim de evitar que hajam duplicidade de campos, ou que sejam quebrados campos indevidamente.

Também se faz necessário atentar para parâmetros utilizados nas cláusulas *where*, que fazem com que sejam lidos um número muito maior de registros ocasionando lentidão na exibição do relatório.

No decorrer do trabalho, buscou-se adotar as melhores práticas de escrita SQL em um Banco de Dados Oracle, das quais algumas queries obtiveram melhora significativa no seu tempo de execução, já outras, não obtiveram melhora significativa pois trabalham com um número muito grande de registros e ao serem analisadas já estavam escritas corretamente.

Contudo, não se descarta como trabalhos futuros uma análise e possível *tuning* de sistema operacional ou até mesmo de infraestrutura, visto que é disposto

de 2 servidores principais (aplicação e Banco de Dados) onde são executados diversos serviços e aplicações pertinentes a realização de tarefas da empresa (em muitos casos uma execução concorrente) fazendo com que tenha um consumo grande de seus recursos.

REFERÊNCIAS

CHAN, Immanuel.; ASHDOWN, Lance. **Oracle Database – Performance Tuning Guide 11g – Release 2 (11.2)**. Copyright ORACLE, 2014.

COUTO, Eder. **Artigo: Aumentando a Performance da Aplicação Através da Otimização de SQL**. Disponível em: <https://imasters.com.br/artigo/4055/bancodedados/aumentando_a_performance_da_aplicacao_atraves_da_otimizacao_de_sql/>. Acesso em: 02/04/2018.

ELMASRI, Ramez.; NAVATHE, Shamkant B. **Sistemas de Banco de Dados. 4. Ed.** São Paulo: Editora Pearson Education do Brasil Ltda., 2005.

IKEMATU, Ricardo Shoiti. **Realizando Tuning na Base de Aplicações**. Disponível em: <<http://www.batebyte.pr.gov.br/modules/conteudo/conteudo.php?conteudo=1592>>. Acesso em: 03/04/2018.

LEGATTI, Eduardo. **Introdução ao conceito de Tablespaces**. Disponível em: <<https://www.oracle.com/technetwork/pt/articles/database-performance/introducao-conceito-de-tablespaces-495850-ptb.html>> Acesso em: 10/09/2018.

LEGATTI, Eduardo. **Qual é mesmo o tamanho de uma tabela no Oracle?** Disponível em: <<http://www.oracle.com/technetwork/pt/articles/database-performance/tamanho-de-uma-tabela-no-oracle-495868-ptb.html>>. Acesso em: 06/04/2018.

LEITE, Alison K. L. et al. **Tuning em queries em Banco de Dados**. Curso de Tecnologia em Banco de Dados - Faculdade de Tecnologia de Bauru (FATEC)

MACÊDO, Diego. **Depuradores (Debuggers)**. Disponível em: <<http://www.diegomacedo.com.br/depuradores-debbugers/>>. Acesso em: 05/04/2018.

REZENDE, Ricardo. **Introdução ao Oracle 9i – Parte II**. Entendendo o Oracle Server. Disponível em: <<https://www.devmedia.com.br/artigo-sql-magazine-13-introducao-ao-oracle-9i-parte-ii/5655>>. Acesso em: 10/09/2018.

SALISH, Asnani, **Oracle Database 11g – Hands on SQL and PL/SQL**, New Delhi: editora PHI Learning Private Limited, 2010.

SILBERSCHATZ, Abraham.; KORTH, Henry F.; SUDARSHAN. S. **Sistema de Banco de Dados. 5. Ed.** São Paulo: Editora Elsevier Ltda., 2006.

ZORZI, Marcelo Tomiello. **Tuning De Sistema Operacional Em Banco De Dados Oracle**. UNIVERSIDADE DE CAXIAS DO SUL CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA, 2015.

APÊNDICE A - PESQUISA À FIM DE IDENTIFICAR POSSÍVEIS PONTOS DE LENTIDÃO NO SISTEMA

- 1 - Quais sistemas você mais utiliza?
() Logix
() Gecex
() BI
() Protheus
- 2 - Qual o setor em que atua?
- 3 - Atua na matriz ou nas filiais?
- 4 - Qual o dispositivo utilizado para acesso ao sistema? (notebook, computador desktop)
- 5 - Utiliza rede wifi ou cabeada?
- 6 - No caso do sistema Logix, utiliza acesso interno ou externo?
- 7 - Percebe algum tipo de lentidão?
- 8 - Se a resposta anterior foi sim, essa lentidão ocorre em algum horário específico? Cite o horário em que ocorre a lentidão.
- 9 - Em qual período do mês você percebe mais lentidão? Cite o período do mês em que ocorre a lentidão.

APÊNDICE B – RELATÓRIO CONHECIMENTOS DE FRETE

```
create or replace view eis_v_cst_fretes as
select distinct
    a.cod_empresa, a.tip_frete, a.dat_entrada_conhec,
    --      G.CONHEC_TRANSPORTE num_conhec, g.serie_conhec ser_conhec,
    g.subserie_conhec ssr_conhec,
    a.num_conhec, a.ser_conhec, a.ssr_conhec,
    a.cod_transpor, e.raz_social, e.cod_uni_feder, j.den_pais,
    a.cod_fiscal_frete, a.gru_ctr_desp_frete, a.cod_tip_despesa,
    c.num_aviso_rec, c.num_seq, c.cod_item, c.den_item, a.val_frete ,
    h.val_pedagio, i.parametro_val val_seguro, f.parametro_val
    CST_pis_cofins, nvl(g.val_bc_pis_decl,0) val_bc_pis_decl,
    nvl(g.pct_pis_item_decl,0)
    pct_pis_item_decl, nvl(g.val_pis_declarado,0) val_pis_declarado,
    nvl(g.val_bc_cofins_decl,0) val_bc_cofins_decl,
    nvl(g.pct_cofins_decl,0) pct_cofins_decl, nvl(g.val_cofins_decl,0)
    val_cofins_decl,
    g.seq_aviso_recebto, b.num_nf num_nota

from frete_sup a
    left join nf_sup b on (b.cod_empresa      = a.cod_empresa
                        and  b.cod_transpor    = a.cod_transpor
                        and  b.num_conhec      = a.num_conhec
                        and  b.ser_conhec      = a.ser_conhec
                        and  b.ssr_conhec      = a.ssr_conhec )

    right join aviso_rec c on (c.cod_empresa    = b.cod_empresa
                        and  c.num_aviso_rec    = b.num_aviso_rec )

    left join fornecedor e on (e.cod_fornecedor = b.cod_transpor)

    left join sup_par_ar f on (f.empresa        = c.cod_empresa
                        and f.aviso_recebto     = c.num_aviso_rec
                        and f.seq_aviso_recebto = c.num_seq
                        and f.parametro        in
('cod_cst_COFINS_FRT') ) --,'cod_cst_COFINS') )

    left join sup_fr_pis_cofins g on (g.empresa      = b.cod_empresa
                        and g.transportador      = b.cod_transpor
                        and g.conhec_transporte = b.num_conhec
                        and g.serie_conhec      = b.ser_conhec
                        and g.subserie_conhec    = b.ssr_conhec
                        and g.aviso_recebto      =
c.num_aviso_rec

                        and g.seq_aviso_recebto = c.num_seq )

    left join pedagio_frete h on (h.cod_empresa    = a.cod_empresa
                        and  h.cod_fornecedor    = a.cod_transpor
                        and  h.num_nf_conhec     = a.num_conhec
                        and  h.ser_nf_conhec     = a.ser_conhec
                        and  h.ssr_nf_conhec     = a.ssr_conhec )

    left join sup_par_frete i on (i.empresa        = a.cod_empresa
                        and i.transportadora      = a.cod_transpor
                        and i.num_conhec          = a.num_conhec
                        and i.serie_conhec        = a.ser_conhec
                        and i.subserie_conhec      = a.ssr_conhec
                        and i.parametro           ='val_seguro_frt' )

    left join paises j on (j.cod_pais            = e.cod_pais )
```



```

where a.tip_frete      = 'C'  -- Compras

union all

-- Frete de Vendas
select
    a.cod_empresa, a.tip_frete, a.dat_entrada_conhec, a.num_conhec,
    a.ser_conhec, a.ssr_conhec,
    a.cod_transpor, e.raz_social, e.cod_uni_feder, m.den_pais,
    a.cod_fiscal_frete, a.gru_ctr_desp_frete, a.cod_tip_despesa,
    g.nota_fiscal,
    i.seq_item_nf, i.item, i.des_item, a.val_frete, j.val_pedagio,
    l.parametro_val_val_seguro,
    f.parametro_val CST_pis_cofins,
    g.val_bas_calc_pis, g.pct_pis, g.val_pis,
    g.val_bc_cofins, g.pct_cofins, g.val_cofins,
    g.sequencia_pedido, g.nota_fiscal

from frete_sup a

    left join fornecedor e on (e.cod_fornecedor = a.cod_transpor )

    left join sup_par_frete f on (f.empresa          = a.cod_empresa
                                and f.transportadora  = a.cod_transpor
                                and f.num_conhec       = a.num_conhec
                                and f.serie_conhec     = a.ser_conhec
                                and f.subserie_conhec  = a.ssr_conhec
                                and f.parametro        in ('cod_cst_COFINS'))
-- , 'cod_cst_PIS') )

    left join sup_frvnd_piscofin g on (g.empresa      = a.cod_empresa
                                       and g.transportador = a.cod_transpor
                                       and g.conhec_transporte = a.num_conhec
                                       and g.serie_conhec   = a.ser_conhec
                                       and g.subserie_conhec = a.ssr_conhec )

    left join fat_nf_mestre h on (h.empresa          = g.empresa
                                and h.nota_fiscal     =
g.nota_fiscal
                                and h.serie_nota_fiscal =
g.serie_nota_fiscal )

    left join fat_nf_item i on (i.empresa            = h.empresa
                               and i.trans_nota_fiscal =
h.trans_nota_fiscal
                               and i.seq_item_nf       =
g.SEQ_ITEM_NOTA_FISCAL_FATURA )

    left join pedagio_frete j on (j.cod_empresa     = a.cod_empresa
                                and j.cod_fornecedor = a.cod_transpor
                                and j.num_nf_conhec  = a.num_conhec
                                and j.ser_nf_conhec   = a.ser_conhec
                                and j.ssr_nf_conhec   = a.ssr_conhec )

    left join sup_par_frete l on (l.empresa          = a.cod_empresa
                                and l.transportadora  = a.cod_transpor
                                and l.num_conhec       = a.num_conhec
                                and l.serie_conhec     = a.ser_conhec
                                and l.subserie_conhec  = a.ssr_conhec
                                and l.parametro        ='val_seguro_frt' )

```

```
left join paises m on (m.cod_pais = e.cod_pais )

where a.tip_frete      = 'V' -- Vendas

union all

--fretes relacionados - entradas
select distinct
    a.cod_empresa, a.tip_frete, a.dat_entrada_conhec,
    G.CONHEC_TRANSPORTE num_conhec, g.serie_conhec ser_conhec,
g.subserie_conhec ssr_conhec,
    a.cod_transpor, e.raz_social, e.cod_uni_feder, j.den_pais,
    a.cod_fiscal_frete, a.gru_ctr_desp_frete, a.cod_tip_despesa,
    c.num_aviso_rec, c.num_seq, c.cod_item, c.den_item, a.val_frete,
    h.val_pedagio, i.parametro_val val_seguro,
    f.parametro_val CST_pis_cofins,
    g.val_bc_pis_decl, g.pct_pis_item_decl, g.val_pis_declarado,
    g.val_bc_cofins_decl, g.pct_cofins_decl, g.val_cofins_decl,
    g.seq_aviso_recebto, b.num_nf num_nota

from frete_sup a


join SUP_FRETE_X_NF_ENTRADA k on (a.cod_empresa=k.empres
                                and a.num_conhec=k.conhec
                                and a.ser_conhec=k.serie_conhec
                                and a.ssr_conhec=k.subserie_conhec
                                and trim(a.cod_transpor)=
trim(k.transportador)

                                )

                                join nf_sup b on (k.empresa=b.cod_empresa
                                and k.aviso_recebto=b.num_aviso_rec)

left join aviso_rec   c on (c.cod_empresa   = k.empres
                            and   c.num_aviso_rec   = k.aviso_recebto )

left join fornecedor e on (e.cod_fornecedor = a.cod_transpor )

left join sup_par_ar f on (f.empres          = c.cod_empresa
                           and f.aviso_recebto       = c.num_aviso_rec
                           and f.seq_aviso_recebto    = c.num_seq
                           and f.parametro           in
('cod cst COFINS FRT') ) --,'cod cst COFINS') )

left join sup_fr_pis_cofins g on (g.empres              = a.cod_empresa
                                  and g.transportador     = a.cod_transpor
                                  and g.conhec_transporte = a.num_conhec
                                  and g.serie_conhec       = a.ser_conhec
                                  and g.subserie_conhec     = a.ssr_conhec
                                  and g.aviso_recebto       =
c.num_aviso_rec
                                  and g.seq_aviso_recebto  = c.num_seq )

left join pedagio_frete h on (h.cod_empresa      = a.cod_empresa
                              and h.cod_fornecedor = a.cod_transpor
                              and h.num_nf_conhec  = a.num_conhec
                              and h.ser_nf_conhec   = a.ser_conhec
                              and h.ssr nf_conhec   = a.ssr_conhec )
```

```
left join sup_par_frete i on (i.empresa      = a.cod_empresa
                             and i.transportadora = a.cod_transpor
                             and i.num_conhec    = a.num_conhec
                             and i.serie_conhec  = a.ser_conhec
                             and i.subserie_conhec = a.ssr_conhec
                             and i.parametro     = 'val_seguro_frt' )

join paises j on (j.cod_pais      = e.cod_pais )

where a.tip_frete      = 'C'  -- Compras
```

APÊNDICE C – RELATÓRIO CONHECIMENTOS DE FRETE – TUNING

```
--create or replace view eis_v_cst_fretes_tuning as
select distinct
    a.cod_empresa, a.tip_frete, a.dat_entrada_conhec,

    a.num_conhec, a.ser_conhec, a.ssr_conhec,
    a.cod_transpor, e.raz_social, e.cod_uni_feder, j.den_pais,
    a.cod_fiscal_frete, a.gru_ctr_desp_frete, a.cod_tip_despesa,
    c.num_aviso_rec, c.num_seq, c.cod_item, c.den_item, a.val_frete ,
    h.val_pedagio, i.parametro_val_val_seguro, f.parametro_val
    CST_pis_cofins, nvl(g.val_bc_pis_decl,0) val_bc_pis_decl,
    nvl(g.pct_pis_item_decl,0) pct_pis_item_decl,
    nvl(g.val_pis_declarado,0) val_pis_declarado,
    nvl(g.val_bc_cofins_decl,0) val_bc_cofins_decl,
    nvl(g.pct_cofins_decl,0) pct_cofins_decl, nvl(g.val_cofins_decl,0)
    val_cofins_decl,
    g.seq_aviso_recebto, b.num_nf num_nota

from frete_sup a
    join nf_sup b on (b.cod_empresa      = a.cod_empresa
                    and  b.cod_transpor   = a.cod_transpor
                    and  b.num_conhec     = a.num_conhec
                    and  b.ser_conhec     = a.ser_conhec
                    and  b.ssr_conhec     = a.ssr_conhec )

    join aviso_rec c on (c.cod_empresa    = b.cod_empresa
                       and  c.num_aviso_rec = b.num_aviso_rec )

    join fornecedor e on (e.cod_fornecedor = b.cod_transpor)

    join sup_par_ar f on (f.empresa        = c.cod_empresa
                       and f.aviso_recebto  = c.num_aviso_rec
                       and f.seq_aviso_recebto = c.num_seq
                       and f.parametro      in
('cod_cst_COFINS_FRT') ) --, 'cod_cst_COFINS') )

    left join sup_fr_pis_cofins g on (g.empresa      = b.cod_empresa
    and g.transportador      = b.cod_transpor
    and g.conhec_transporte = b.num_conhec
    and g.serie_conhec      = b.ser_conhec
    and g.subserie_conhec   = b.ssr_conhec
    and g.aviso_recebto     =

c.num_aviso_rec

    and g.seq_aviso_recebto = c.num_seq )

    left join pedagio_frete h on (h.cod_empresa      = a.cod_empresa
    and h.cod_fornecedor = a.cod_transpor
    and h.num_nf_conhec  = a.num_conhec
    and h.ser_nf_conhec  = a.ser_conhec
    and h.ssr_nf_conhec  = a.ssr_conhec )

    left join sup_par_frete i on (i.empresa        = a.cod_empresa
    and i.transportadora = a.cod_transpor
    and i.num_conhec     = a.num_conhec
    and i.serie_conhec   = a.ser_conhec
    and i.subserie_conhec = a.ssr_conhec
    and i.parametro      ='val_seguro_frt' )

    left join paises j on (j.cod_pais      = e.cod_pais )
```

```

where a.tip_frete          = 'C'  -- Compras
and not exists (
    select * from SUP_FRETE_X_NF_ENTRADA rel
    where a.cod_empresa = rel.cod_empresa
    and a.num_conhe=rel.num_conhec
    and a.ser_conhec=rel.ser_conhec
    and a.cod_transpor=rel.cod_transpor
)

union all

-- Frete de Vendas
select
    a.cod_empresa, a.tip_frete, a.dat_entrada_conhec, a.num_conhec,
    a.ser_conhec, a.ssr_conhec,
    a.cod_transpor, e.raz_social, e.cod_uni_feder, m.den_pais,
    a.cod_fiscal_frete, a.gru_ctr_desp_frete,
    a.cod_tip_despesa,g.nota_fiscal,
    i.seq_item_nf, i.item, i.des_item, a.val_frete, j.val_pedagio,
    l.parametro_val_val_seguro,
    f.parametro_val CST_pis_cofins, g.val_bas_calc_pis, g.pct_pis,
    g.val_pis,
    g.val_bc_cofins, g.pct_cofins, g.val_cofins,
    g.sequencia_pedido, g.nota_fiscal

from frete_sup a

    left join fornecedor e on (e.cod_fornecedor = a.cod_transpor )

    left join sup_par_frete f on (f.empresa          = a.cod_empresa
                                and f.transportadora   = a.cod_transpor
                                and f.num_conhec        = a.num_conhec
                                and f.serie_conhec      = a.ser_conhec
                                and f.subserie_conhec   = a.ssr_conhec
                                and f.parametro         in ('cod_cst_COFINS'))

-- , 'cod_cst_PIS' )

    left join sup_frvnd_piscofin g on (g.empresa          = a.cod_empresa
                                       and g.transportador   = a.cod_transpor
                                       and g.conhec_transporte = a.num_conhec
                                       and g.serie_conhec      = a.ser_conhec
                                       and g.subserie_conhec   = a.ssr_conhec )

    left join fat_nf_mestre h on (h.empresa          = g.empresa
                                  and h.nota_fiscal     =
g.nota_fiscal
                                  and h.serie_nota_fiscal =
g.serie_nota_fiscal )

    left join fat_nf_item i on (i.empresa          = h.empresa
                                and i.trans_nota_fiscal =
h.trans_nota_fiscal
                                and i.seq_item_nf      =
g.SEQ_ITEM_NOTA_FISCAL_FATURA )

    left join pedagio_frete j on (j.cod_empresa      = a.cod_empresa
                                  and j.cod_fornecedor = a.cod_transpor
                                  and j.num_nf_conhec  = a.num_conhec
                                  and j.ser_nf_conhec   = a.ser_conhec
                                  and j.ssr_nf_conhec   = a.ssr_conhec )

```

```

        left join sup_par_frete l on (l.empresa          = a.cod_empresa
                                     and l.transportadora = a.cod_transpor
                                     and l.num_conhec      = a.num_conhec
                                     and l.serie_conhec    = a.ser_conhec
                                     and l.subserie_conhec = a.ssr_conhec
                                     and l.parametro       ='val_seguro_frt' )

        left join paises m on (m.cod_pais              = e.cod_pais )

where a.tip_frete      = 'V' -- Vendas

union all

--fretes relacionados - entradas
select distinct
    a.cod_empresa, a.tip_frete, a.dat_entrada_conhec,
    G.CONHEC_TRANSPORTE num_conhec, g.serie_conhec ser_conhec,
    g.subserie_conhec ssr_conhec,
    a.cod_transpor, e.raz_social, e.cod_uni_feder, j.den_pais,
    a.cod_fiscal_frete, a.gru_ctr_desp_frete, a.cod_tip_despesa,
    c.num_aviso_rec, c.num_seq, c.cod_item, c.den_item, a.val_frete,
    h.val_pedagio, i.parametro_val val_seguro, f.parametro_val
    CST_pis_cofins,
    g.val_bc_pis_decl, g.pct_pis_item decl, g.val_pis declarado,
    g.val_bc_cofins_decl, g.pct_cofins_decl, g.val_cofins_decl,
    g.seq_aviso_recebto, b.num_nf num_nota

from frete_sup a

join SUP_FRETE_X_NF_ENTRADA k on (a.cod_empresa=k.empresa
                                and a.num_conhec=k.conhec
                                and a.ser_conhec=k.serie_conhec
                                and a.ssr_conhec=k.subserie_conhec
                                and trim(a.cod_transpor)=
trim(k.transportador)

                                )

                                join nf_sup b on (k.empresa=b.cod_empresa
                                and k.aviso_recebto=b.num_aviso_rec)

left join aviso_rec c on (c.cod_empresa = k.empresa
                          and c.num_aviso_rec = k.aviso_recebto )

left join fornecedor e on (e.cod_fornecedor = a.cod_transpor )

left join sup_par_ar f on (f.empresa          = c.cod_empresa
                          and f.aviso_recebto = c.num_aviso_rec
                          and f.seq_aviso_recebto = c.num_seq
                          and f.parametro      in
('cod_cst_COFINS_FRT') ) --,'cod_cst_COFINS') )

left join sup_fr_pis_cofins g on (g.empresa          = a.cod_empresa
                                and g.transportador   = a.cod_transpor
                                and g.conhec_transporte = a.num_conhec
                                and g.serie_conhec     = a.ser_conhec
                                and g.subserie_conhec  = a.ssr_conhec
                                and g.aviso_recebto    =
c.num_aviso_rec

```

```

                                and g.seq_aviso_recebto = c.num_seq )

left join pedagio_frete h on (h.cod_empresa      = a.cod_empresa
                             and h.cod_fornecedor = a.cod_transpor
                             and h.num_nf_conhec  = a.num_conhec
                             and h.ser_nf_conhec  = a.ser_conhec
                             and h.ssr_nf_conhec  = a.ssr_conhec )

left join sup_par_frete i on (i.empresa          = a.cod_empresa
                             and i.transportadora = a.cod_transpor
                             and i.num_conhec     = a.num_conhec
                             and i.serie_conhec   = a.ser_conhec
                             and i.subserie_conhec = a.ssr_conhec
                             and i.parametro      = 'val_seguro_frt' )

join paises j on (j.cod_pais          = e.cod_pais )

where a.tip_frete      = 'C'  -- Compras

```

APÊNDICE D – RELATÓRIO ENTRADAS

```
--create or replace view eis_v_cst_entradas as
select a.cod_empresa,
       a.cod_operacao,
       a.dat_entrada_nf,
       b.cod_fiscal_item,
       a.cod_fornecedor,
       c.raz_social,
       c.cod_uni_feder,
       i.den_pais,
       a.num_nf,
       a.ser_nf,
       a.ies_especie_nf,
       a.cod_regist_entrada,
       a.num_aviso_rec,
       b.num_seq,
       b.cod_item,
       b.den_item,
       b.qtd_recebida,
       b.pre_unit_nf,
       b.val_liquido_item,
       b.val_contabil_item,
       b.val_desc_item,
       b.val_despesa_aces_i,
       b.val_devoluc,
       b.val_frete,
       b.val_acrescimos,
       b.val_enc_financ,
       b.val_compl_estoque,
       h.val_icms_st,
       b.val_ipi_decl_item+b.val_ipi_desp_aces val_ipi_decl_item,
       b.gru_ctr_desp_item,
       b.cod_tip_despesa,
       d.parametro_val      cst_pis,
       e.val_base_pis_d,
       e.pct_pis_item_d,
       e.val_pis_d,
       f.parametro_val      cst_cofins,
       e.val_base_cofins_d,
       e.pct_cofins_item_d,
       e.val_cofins_d,
       a.num_conhec,
       a.ser_conhec,
       a.ssr_conhec,
       a.cod_transpor,
       t.raz_social raz_transpor,
       u.parametro_booleano fnc_com_suspensao,
       b.cod_cla_fisc,
       b.val_ipi_calc_item,
       it.num_conta

from nf_sup a
left join aviso_rec b
  on (b.cod_empresa = a.cod_empresa and b.num_aviso_rec =
a.num_aviso_rec)

left join item_sup it on it.cod_empresa = b.cod_empresa
                    and it.cod_item   = b.cod_item

join eis_dimensao_dat dt
```



```

on (trunc(a.dat_entrada_nf) = dt.dat_refer)

left join fornecedor c
  on (c.cod_fornecedor = a.cod_fornecedor)

left join sup_par_ar d
  on (d.empresa = b.cod_empresa and d.aviso_recebto = b.num_aviso_rec and
      d.seq_aviso_recebto = b.num_seq and d.parametro IN ('cod_cst_PIS'))

left join sup_par_ar f
  on (f.empresa = b.cod_empresa and f.aviso_recebto = b.num_aviso_rec and
      f.seq_aviso_recebto = b.num_seq and
      f.parametro IN ('cod_cst_COFINS'))

left join ar_pis_cofins e
  on (e.cod_empresa = b.cod_empresa and e.num_aviso_rec = b.num_aviso_rec
and
    e.num_seq = b.num_seq)

left join aviso_rec_compl g
  on (g.cod_empresa = a.cod_empresa
and g.num_aviso_rec = a.num_aviso_rec
and g.ies_situacao = 'N' )

left join ar_subst_tribut h
  on (h.cod_empresa = b.cod_empresa
and h.num_aviso_rec = b.num_aviso_rec
and h.num_seq = b.num_seq)

left join paises i
  on (i.cod_pais = c.cod_pais )

left join fornecedor t
  on (t.cod_fornecedor = a.cod_transpor)

left join sup_par_fornecedor u on (u.empresa = 'SE'
and u.fornecedor = a.cod_fornecedor
and u.parametro = 'ies_susp_piscofins '
)

```

APÊNDICE E – RELATÓRIO ENTRADAS – TUNING

```
--create or replace view eis_v_cst_entradas_tuning as
select a.cod_empresa,
       a.cod_operacao,
       a.dat_entrada_nf,
       b.cod_fiscal_item,
       a.cod_fornecedor,
       c.raz_social,
       c.cod_uni_feder,
       i.den_pais,
       a.num_nf,
       a.ser_nf,
       a.ies_especie_nf,
       a.cod_regist_entrada,
       a.num_aviso_rec,
       b.num_seq,
       b.cod_item,
       b.den_item,
       b.qtd_recebida,
       b.pre_unit_nf,
       b.val_liquido_item,
       b.val_contabil_item,
       b.val_desc_item,
       b.val_despesa_aces_i,
       b.val_devoluc,
       b.val_frete,
       b.val_acrescimos,
       b.val_enc_financ,
       b.val_compl_estoque,
       h.val_icms_st,
       b.val_ipi_decl_item+b.val_ipi_desp_aces val_ipi_decl_item,
       b.gru_ctr_desp_item,
       b.cod_tip_despesa,
       d.parametro_val      cst_pis,
       e.val_base_pis_d,
       e.pct_pis_item_d,
       e.val_pis_d,
       f.parametro_val      cst_cofins,
       e.val_base_cofins_d,
       e.pct_cofins_item_d,
       e.val_cofins_d,
       a.num_conhec,
       a.ser_conhec,
       a.ssr_conhec,
       a.cod_transpor,
       t.raz_social raz_transpor,
       u.parametro_booleano fnc_com_suspensao,
       b.cod_cla_fisc,
       b.val_ipi_calc_item,
       it.num_conta

from nf_sup a
join aviso_rec b
  on (b.cod_empresa = a.cod_empresa and b.num_aviso_rec =
a.num_aviso_rec)

join item_sup it on it.cod_empresa = b.cod_empresa -->>
                  and it.cod_item      = b.cod_item

join fornecedor c -->>
```

```

        on (c.cod_fornecedor = a.cod_fornecedor)

join sup_par_ar d
    on (d.empresa = b.cod_empresa and d.aviso_recebto = b.num_aviso_rec and
        d.seq_aviso_recebto = b.num_seq and d.parametro IN ('cod_cst_PIS'))

join sup_par_ar f
    on (f.empresa = b.cod_empresa and f.aviso_recebto = b.num_aviso_rec and
        f.seq_aviso_recebto = b.num_seq and
        f.parametro IN ('cod_cst_COFINS'))

join ar_pis_cofins e
    on (e.cod_empresa = b.cod_empresa and e.num_aviso_rec = b.num_aviso_rec
and
    e.num_seq = b.num_seq)

join aviso_rec_compl g
    on (g.cod_empresa = a.cod_empresa
and g.num_aviso_rec = a.num_aviso_rec
and g.ies_situacao = 'N' )

left join ar_subst_tribut h
    on (h.cod_empresa = b.cod_empresa
and h.num_aviso_rec = b.num_aviso_rec
and h.num_seq = b.num_seq)

join paises i
    on (i.cod_pais = c.cod_pais )

join fornecedor t
    on (t.cod_fornecedor = a.cod_transpor)

left join sup_par_fornecedor u on (u.empresa = 'SE'
and u.fornecedor = a.cod_fornecedor
and u.parametro = 'ies_susp_piscofins '
)

```

APÊNDICE F – RELATÓRIO FATURAMENTO

```
--CREATE OR REPLACE VIEW EIS_V_FAT_VANILLA AS
SELECT 'NOTAS VENDAS' IES_TIPO,
       A.EMPRESA,
       YEAR(A.DAT_HOR_EMISSAO) ANO,
       MONTH(A.DAT_HOR_EMISSAO) MES,
       TRUNC(A.DAT_HOR_EMISSAO) DAT_HOR_EMISSAO,
       A.CLIENTE,
       D.NOM_CLIENTE,
       D.NUM_TELEX,
       E.DEN_TIP_CLI,
       F.COD_UNI_FEDER,
       F.DEN_CIDADE,
       H.DEN_PAIS,
       K.DEN_TIP_CARTEIRA,
       A.NATUREZA_OPERACAO COD_NAT_OPER,
       C.DEN_NAT_OPER,
       A.COND_PAGTO,
       M.DEN_CND_PGTO,
       A.NOTA_FISCAL,
       B.PEDIDO,
       N.PRZ_ENTREGA,
       J.COD_REPRES REPRESENTANTE,
       O PCT_COMISSAO,
       J.RAZ_SOCIAL,
       EIS_PKG_UTIL_LX.get_aen(L.COD_LIN_PROD) LIN_PROD1,
       EIS_PKG_UTIL_LX.get_aen(L.COD_LIN_PROD, L.COD_LIN_RECEI) LIN_PROD2,
       L.COD_ITEM ITEM,
       L.DEN_ITEM,
       L.COD_UNID_MED,
       l.cod_familia,
       fam.den_familia,
       B.QTD_ITEM,
       B.PRECO_UNIT_LIQUIDO,
       B.VAL_LIQUIDO_ITEM,
       IPI.VAL_TRIBUTO_TOT VAL_IPI,
       ICMS.VAL_TRIBUTO_TOT VAL_ICMS,
       PIS.VAL_TRIBUTO_TOT VAL_PIS,
       COFINS.VAL_TRIBUTO_TOT VAL_COFINS,
       A.TIP_FRETE,
       DECODE(A.TIP_FRETE, 1, 'CIF PAGO', 2, 'CIF COBRADO', 'FOB')
TIPO_FRETE,
       b.val_fret_consig + b.val_segr_consig + b.val_frete_cliente +
       b.val_seguro_cliente val_frete_rodov,
       ((b.val_desc_item + b.val_desc_merc + b.val_desc_contab +
       b.val_desc_duplicata) * -1) val_desconto,
       b.val_acresc_item + b.val_acre_merc + b.val_acre_duplicata
val_acrescimo,

       case
         when fat_dacre_item_nf.ind_desc_acre = 'D' then
           fat_dacre_item_nf.val_desc_acre * -1
         else
           fat_dacre_item_nf.val_desc_acre
         end val_desc_acre

FROM FAT_NF_MESTRE A

left JOIN FAT_NF_ITEM B
  ON A.EMPRESA = B.EMPRESA
```

```

        AND A.TRANS_NOTA_FISCAL = B.TRANS_NOTA_FISCAL

left JOIN NAT_OPERACAO C
    ON A.NATUREZA_OPERACAO = C.COD_NAT_OPER

JOIN CLIENTES D
    ON A.CLIENTE = D.COD_CLIENTE

JOIN TIPO_CLIENTE E
    ON D.COD_TIP_CLI = E.COD_TIP_CLI

JOIN CIDADES F
    ON D.COD_CIDADE = F.COD_CIDADE

JOIN UNI_FEDER G
    ON F.COD_UNI_FEDER = G.COD_UNI_FEDER

JOIN PAISES H
    ON G.COD_PAIS = H.COD_PAIS

LEFT JOIN PEDIDOS I
    ON B.EMPRESA = I.COD_EMPRESA
    AND B.PEDIDO = I.NUM_PEDIDO

LEFT JOIN REPRESENTANTE J
    ON I.COD_REPRES = J.COD_REPRES

LEFT JOIN TIPO_CARTEIRA K
    ON I.COD_TIP_CARTEIRA = K.COD_TIP_CARTEIRA

LEFT JOIN ITEM L
    ON B.EMPRESA = L.COD_EMPRESA
    AND B.ITEM = L.COD_ITEM

LEFT JOIN COND_PGTO M
    ON A.COND_PAGTO = M.COD_CND_PGTO

LEFT JOIN PED_ITENS N
    ON B.EMPRESA = N.COD_EMPRESA
    AND B.PEDIDO = N.NUM_PEDIDO
    AND B.SEQ_ITEM_PEDIDO = N.NUM_SEQUENCIA

LEFT JOIN FAT_NF_ITEM_FISC ICMS
    ON B.EMPRESA = ICMS.EMPRESA
    AND B.TRANS_NOTA_FISCAL = ICMS.TRANS_NOTA_FISCAL
    AND B.SEQ_ITEM_NF = ICMS.SEQ_ITEM_NF
    AND ICMS.TRIBUTO_BENEF = 'ICMS'

LEFT JOIN FAT_NF_ITEM_FISC IPI
    ON B.EMPRESA = IPI.EMPRESA
    AND B.TRANS_NOTA_FISCAL = IPI.TRANS_NOTA_FISCAL
    AND B.SEQ_ITEM_NF = IPI.SEQ_ITEM_NF
    AND IPI.TRIBUTO_BENEF = 'IPI'

LEFT JOIN FAT_NF_ITEM_FISC PIS
    ON B.EMPRESA = PIS.EMPRESA
    AND B.TRANS_NOTA_FISCAL = PIS.TRANS_NOTA_FISCAL
    AND B.SEQ_ITEM_NF = PIS.SEQ_ITEM_NF
    AND PIS.TRIBUTO_BENEF = 'PIS_REC'

LEFT JOIN FAT_NF_ITEM_FISC COFINS

```

```

    ON B.EMPRESA = COFINS.EMPRESA
    AND B.TRANS_NOTA_FISCAL = COFINS.TRANS_NOTA_FISCAL
    AND B.SEQ_ITEM_NF = COFINS.SEQ_ITEM_NF
    AND COFINS.TRIBUTO_BENEF = 'COFINS_REC'

left join familia fam
    ON fam.cod_empresa = l.cod_empresa
    AND fam.cod_familia = l.cod_familia

LEFT JOIN fat_dacre_item_nf
    ON fat_dacre_item_nf.EMPRESA = A.EMPRESA
    AND fat_dacre_item_nf.TRANS_NOTA_FISCAL = A.TRANS_NOTA_FISCAL
    AND fat_dacre_item_nf.SEQ_ITEM_NF = B.TRANS_NOTA_FISCAL

WHERE A.SIT_NOTA_FISCAL = 'N'
AND C.IES_ESTATISTICA <> 'N'

UNION

SELECT 'NOTAS ENTREGA FUTURA' IES_TIPO,
    A.EMPRESA,
    YEAR(A.DAT_HOR_EMISSAO) ANO,
    MONTH(A.DAT_HOR_EMISSAO) MES,
    TRUNC(A.DAT_HOR_EMISSAO) DAR_HOR_EMISSAO,
    A.CLIENTE,
    D.NOM_CLIENTE,
    D.NUM_TELEX,
    E.DEN_TIP_CLI,
    F.COD_UNI_FEDER,
    F.DEN_CIDADE,
    H.DEN_PAIS,
    K.DEN_TIP_CARTEIRA,
    A.NATUREZA_OPERACAO COD_NAT_OPER,
    C.DEN_NAT_OPER,
    A.COND_PAGTO,
    M.DEN_CND_PGTO,
    A.NOTA_FISCAL,
    B.PEDIDO,
    N.PRZ_ENTREGA,
    J.COD_REPRES REPRESENTANTE,
    0 PCT_COMISSAO,
    J.RAZ_SOCIAL,
    EIS_PKG_UTIL_LX.get_aen(L.COD_LIN_PROD) LIN_PROD1,
    EIS_PKG_UTIL_LX.get_aen(L.COD_LIN_PROD, L.COD_LIN_RECEI) LIN_PROD2,
    L.COD_ITEM ITEM,
    L.DEN_ITEM,
    L.COD_UNID_MED,
    l.cod_familia,
    fam.den_familia,
    B.QTD_ITEM,
    B.PRECO_UNIT LIQUIDO,
    B.VAL_LIQUIDO_ITEM,
    IPI.VAL_TRIBUTO_TOT VAL_IPI,
    ICMS.VAL_TRIBUTO_TOT VAL_ICMS,
    PIS.VAL_TRIBUTO_TOT VAL_PIS,
    COFINS.VAL_TRIBUTO_TOT VAL_COFINS,
    A.TIP_FRETE,
    DECODE(A.TIP_FRETE, 1, 'CIF PAGO', 2, 'CIF COBRADO', 'FOB')
TIPO_FRETE,
    b.val_fret_consigt + b.val_sogr_consigt + b.val_frete_cliente +
    b.val_seguro_cliente val_frete_rodov,

```

```

        ((b.val_desc_item + b.val_desc_merc + b.val_desc_contab +
        b.val_desc_duplicata) * -1) val_desconto,
        b.val_acresc_item + b.val_acre_merc + b.val_acre_duplicata
val_acrescimo,

```

```

    case
        when fat_dacre_item_nf.ind_desc_acre = 'D' then
            fat_dacre_item_nf.val_desc_acre * -1
        else
            fat_dacre_item_nf.val_desc_acre
    end val_desc_acre

```

```

FROM FAT_NF_MESTRE A
JOIN FAT_NF_ITEM B
    ON A.EMPRESA = B.EMPRESA
    AND A.TRANS_NOTA_FISCAL = B.TRANS_NOTA_FISCAL
JOIN NAT_OPERACAO C
    ON A.NATUREZA_OPERACAO = C.COD_NAT_OPER
JOIN CLIENTES D
    ON A.CLIENTE = D.COD_CLIENTE
JOIN TIPO_CLIENTE E
    ON D.COD_TIP_CLI = E.COD_TIP_CLI
JOIN CIDADES F
    ON D.COD_CIDADE = F.COD_CIDADE
JOIN UNI_FEDER G
    ON F.COD_UNI_FEDER = G.COD_UNI_FEDER
JOIN PAISES H
    ON G.COD_PAIS = H.COD_PAIS
LEFT JOIN PEDIDOS I
    ON B.EMPRESA = I.COD_EMPRESA
    AND B.PEDIDO = I.NUM_PEDIDO
LEFT JOIN REPRESENTANTE J
    ON I.COD_REPRES = J.COD_REPRES
LEFT JOIN TIPO_CARTEIRA K
    ON I.COD_TIP_CARTEIRA = K.COD_TIP_CARTEIRA
LEFT JOIN ITEM L
    ON B.EMPRESA = L.COD_EMPRESA
    AND B.ITEM = L.COD_ITEM
LEFT JOIN COND_PGTO M
    ON A.COND_PAGTO = M.COD_CND_PGTO
LEFT JOIN PED_ITENS N
    ON B.EMPRESA = N.COD_EMPRESA
    AND B.PEDIDO = N.NUM_PEDIDO
    AND B.SEQ_ITEM_PEDIDO = N.NUM_SEQUENCIA
LEFT JOIN FAT_NF_ITEM_FISC ICMS
    ON B.EMPRESA = ICMS.EMPRESA
    AND B.TRANS_NOTA_FISCAL = ICMS.TRANS_NOTA_FISCAL
    AND B.SEQ_ITEM_NF = ICMS.SEQ_ITEM_NF
    AND ICMS.TRIBUTO_BENEF = 'ICMS'
LEFT JOIN FAT_NF_ITEM_FISC IPI
    ON B.EMPRESA = IPI.EMPRESA
    AND B.TRANS_NOTA_FISCAL = IPI.TRANS_NOTA_FISCAL
    AND B.SEQ_ITEM_NF = IPI.SEQ_ITEM_NF
    AND IPI.TRIBUTO_BENEF = 'IPI'
LEFT JOIN FAT_NF_ITEM_FISC PIS
    ON B.EMPRESA = PIS.EMPRESA
    AND B.TRANS_NOTA_FISCAL = PIS.TRANS_NOTA_FISCAL
    AND B.SEQ_ITEM_NF = PIS.SEQ_ITEM_NF
    AND PIS.TRIBUTO_BENEF = 'PIS_REC'
LEFT JOIN FAT_NF_ITEM_FISC COFINS
    ON B.EMPRESA = COFINS.EMPRESA

```

```

AND B.TRANS_NOTA_FISCAL = COFINS.TRANS_NOTA_FISCAL
AND B.SEQ_ITEM_NF = COFINS.SEQ_ITEM_NF
AND COFINS.TRIBUTO_BENEF = 'COFINS_REC'
left join familia fam
  ON fam.cod_empresa = l.cod_empresa
  AND fam.cod_familia = l.cod_familia

LEFT JOIN fat_dacre_item_nf
  ON fat_dacre_item_nf.EMPRESA = A.EMPRESA
  AND fat_dacre_item_nf.TRANS_NOTA_FISCAL = A.TRANS_NOTA_FISCAL
  AND fat_dacre_item_nf.SEQ_ITEM_NF = B.TRANS_NOTA_FISCAL

WHERE A.SIT_NOTA_FISCAL = 'N'
AND C.COD_NAT_OPER IN (107, 108, 405, 406, 455)

UNION

SELECT 'NOTAS DE COMPLEMENTO' IES_TIPO,
  A.EMPRESA,
  YEAR(A.DAT_HOR_EMISSAO) ANO,
  MONTH(A.DAT_HOR_EMISSAO) MES,
  TRUNC(A.DAT_HOR_EMISSAO) DAR_HOR_EMISSAO,
  A.CLIENTE,
  D.NOM_CLIENTE,
  D.NUM_TELEX,
  E.DEN_TIP_CLI,
  F.COD_UNI_FEDER,
  F.DEN_CIDADE,
  H.DEN_PAIS,
  K.DEN_TIP_CARTEIRA,
  A.NATUREZA_OPERACAO COD_NAT_OPER,
  C.DEN_NAT_OPER,
  A.COND_PAGTO,
  M.DEN_CND_PGTO,
  A.NOTA_FISCAL,
  B.PEDIDO,
  N.PRZ_ENTREGA,
  J.COD_REPRES REPRESENTANTE,
  0 PCT_COMISSAO,
  J.RAZ_SOCIAL,
  EIS_PKG_UTIL_LX.get_aen(L.COD_LIN_PROD) LIN_PROD1,
  EIS_PKG_UTIL_LX.get_aen(L.COD_LIN_PROD, L.COD_LIN_RECEI) LIN_PROD2,
  L.COD_ITEM ITEM,
  L.DEN_ITEM,
  L.COD_UNID_MED,
  l.cod_familia,
  fam.den_familia,
  B.QTD_ITEM,
  B.PRECO_UNIT_LIQUIDO,
  B.VAL_LIQUIDO_ITEM,
  IPI.VAL_TRIBUTO_TOT VAL_IPI,
  ICMS.VAL_TRIBUTO_TOT VAL_ICMS,
  PIS.VAL_TRIBUTO_TOT VAL_PIS,
  COFINS.VAL_TRIBUTO_TOT VAL_COFINS,
  A.TIP_FRETE,
  DECODE(A.TIP_FRETE, 1, 'CIF PAGO', 2, 'CIF COBRADO', 'FOB')
TIPO_FRETE,
  b.val_fret_consig + b.val_segr_consig + b.val_frete_cliente +
  b.val_seguro_cliente val_frete_rodov,
  ((b.val_desc_item + b.val_desc_merc + b.val_desc_contab +

```



```

        b.val_desc_duplicata) * -1) val_desconto,
        b.val_acresc_item + b.val_acre_merc + b.val_acre_duplicata
val_acrescimo,

```

```

    case
        when fat_dacre_item_nf.ind_desc_acre = 'D' then
            fat_dacre_item_nf.val_desc_acre * -1
        else
            fat_dacre_item_nf.val_desc_acre
    end val_desc_acre

```

```

FROM FAT_NF_MESTRE A
JOIN FAT_NF_ITEM B
    ON A.EMPRESA = B.EMPRESA
    AND A.TRANS_NOTA_FISCAL = B.TRANS_NOTA_FISCAL
JOIN NAT_OPERACAO C
    ON A.NATUREZA_OPERACAO = C.COD_NAT_OPER
JOIN CLIENTES D
    ON A.CLIENTE = D.COD_CLIENTE
JOIN TIPO_CLIENTE E
    ON D.COD_TIP_CLI = E.COD_TIP_CLI
JOIN CIDADES F
    ON D.COD_CIDADE = F.COD_CIDADE
JOIN UNI_FEDER G
    ON F.COD_UNI_FEDER = G.COD_UNI_FEDER
JOIN PAISES H
    ON G.COD_PAIS = H.COD_PAIS
LEFT JOIN PEDIDOS I
    ON B.EMPRESA = I.COD_EMPRESA
    AND B.PEDIDO = I.NUM_PEDIDO
LEFT JOIN REPRESENTANTE J
    ON I.COD_REPRES = J.COD_REPRES
LEFT JOIN TIPO_CARTEIRA K
    ON I.COD_TIP_CARTEIRA = K.COD_TIP_CARTEIRA
LEFT JOIN ITEM L
    ON B.EMPRESA = L.COD_EMPRESA
    AND B.ITEM = L.COD_ITEM
LEFT JOIN COND_PGTO M
    ON A.COND_PAGTO = M.COD_CND_PGTO
LEFT JOIN PED_ITENS N
    ON B.EMPRESA = N.COD_EMPRESA
    AND B.PEDIDO = N.NUM_PEDIDO
    AND B.SEQ_ITEM_PEDIDO = N.NUM_SEQUENCIA
LEFT JOIN FAT_NF_ITEM_FISC_ICMS
    ON B.EMPRESA = ICMS.EMPRESA
    AND B.TRANS_NOTA_FISCAL = ICMS.TRANS_NOTA_FISCAL
    AND B.SEQ_ITEM_NF = ICMS.SEQ_ITEM_NF
    AND ICMS.TRIBUTO_BENEF = 'ICMS'
LEFT JOIN FAT_NF_ITEM_FISC_IPI
    ON B.EMPRESA = IPI.EMPRESA
    AND B.TRANS_NOTA_FISCAL = IPI.TRANS_NOTA_FISCAL
    AND B.SEQ_ITEM_NF = IPI.SEQ_ITEM_NF
    AND IPI.TRIBUTO_BENEF = 'IPI'
LEFT JOIN FAT_NF_ITEM_FISC_PIS
    ON B.EMPRESA = PIS.EMPRESA
    AND B.TRANS_NOTA_FISCAL = PIS.TRANS_NOTA_FISCAL
    AND B.SEQ_ITEM_NF = PIS.SEQ_ITEM_NF
    AND PIS.TRIBUTO_BENEF = 'PIS_REC'
LEFT JOIN FAT_NF_ITEM_FISC_COFINS
    ON B.EMPRESA = COFINS.EMPRESA
    AND B.TRANS_NOTA_FISCAL = COFINS.TRANS_NOTA_FISCAL

```

```

AND B.SEQ_ITEM_NF = COFINS.SEQ_ITEM_NF
AND COFINS.TRIBUTO_BENEF = 'COFINS_REC'
left join familia fam
  ON fam.cod_empresa = l.cod_empresa
  AND fam.cod_familia = l.cod_familia

LEFT JOIN fat_dacre_item_nf
  ON fat_dacre_item_nf.EMPRESA = A.EMPRESA
  AND fat_dacre_item_nf.TRANS_NOTA_FISCAL = A.TRANS_NOTA_FISCAL
  AND fat_dacre_item_nf.SEQ_ITEM_NF = B.TRANS_NOTA_FISCAL

WHERE A.SIT_NOTA_FISCAL = 'N'
AND C.COD_NAT_OPER IN (118, 115, 165, 226, 405, 406, 455)

```

union

```

SELECT 'DEVOLUCAO ',
  a.cod_empresa,
  year(a.dat_entrada_nf),
  month(a.dat_entrada_nf),
  a.dat_entrada_nf,
  a.cod_fornecedor,
  c.nom_cliente,
  C.num_telex,
  e.den_tip_cli,
  xx.cod_uni_feder,
  xx.den_cidade,
  pa.den_pais,
  q.den_tip_carteira,
  nf.natureza_operacao,
  g.den_nat_oper,
  a.cnd_pgto_nf,
  u.des_cnd_pgto,
  nf.nota_fiscal,
  s.ped_nf_fatura,
  NULL dat_entrega,
  v.representante,
  v.pct_comissao,
  s.raz_social,
  d.den_estr_linprod,
  e.den_estr_linprod,
  b.cod_item,
  f.den_item,
  f.cod_unid_med,
  f.cod_familia,
  fam.den_familia,
  case
    when sum(b.qtd_recebida * -1) = 0 or
         sum(b.qtd_recebida * -1) is null then
      sum(b.qtd_declarad_nf * -1)
    else
      sum(b.qtd_recebida * -1)
  end,
  sum(b.pre_unit_nf * -1),
  sum(b.val_liquido_item * -1),
  sum((b.val_ipi_calc_item + b.val_ipi_desp_aces) * -1) val_ipi,
  sum(b.val_icms_item_c * -1) val_icms,
  sum(j.val_pis * -1) val_pis,
  sum(j.val_cofins * -1) val_cofins,
  nf.tip_frete,

```

```

        case
        when nf.tip_frete = 1 then
            'CIF PAGO'
        else
            case
            when nf.tip_frete = 2 then
                'CIF COBRADO'
            else
                'FOB'
            end
        end,
        sum(nf.val_frete_rodov * -1) val_frete_rodov,
        0 val_desconto,
        0 val_acrescimo,
        0 val_desc_acre

from nf_sup a

join aviso_rec b
    on a.cod_empresa = b.cod_empresa
    and a.num_aviso_rec = b.num_aviso_rec

join clientes c
    on a.cod_fornecedor = c.cod_cliente

join cidades xx
    on c.cod_cidade = xx.cod_cidade

join uni_feder yy
    on yy.cod_uni_feder = xx.cod_uni_feder

join tipo_cliente e
    on c.cod_tip_cli = e.cod_tip_cli

join item f
    on b.cod_empresa = f.cod_empresa
    and b.cod_item = f.cod_item

left join obf_dvcli_piscofin j
    on j.empresa = b.cod_empresa
    and j.aviso_recebto = b.num_aviso_rec
    and j.seq_aviso_recebto = b.num_seq

left join paises pa
    on pa.cod_pais = yy.cod_pais

left join sup_nf_devol_cli s
    on b.cod_empresa = s.empresa
    and b.num_aviso_rec = s.aviso_recebto
    and b.num_seq = s.seq_aviso_recebto

join fat_nf_mestre nf
    on nf.empresa = s.empresa
    and nf.nota_fiscal = s.nota_fiscal_fatura
    and nf.serie_nota_fiscal = s.ser_nf_fatura

left join fat_nf_repr v
    on (v.empresa = nf.empresa and
        v.trans_nota_fiscal = nf.trans_nota_fiscal and
        v.seq_representante = 1)

```

```

join nat_operacao g
  on nf.natureza_operacao = g.cod_nat_oper
  and g.ies_estatistica in ('T', 'V')

join linha_prod d
  on d.cod_lin_prod = f.cod_lin_prod
  and d.cod_lin_recei = 0
  and d.cod_seg_merc = 0
  and d.cod_cla_uso = 0

join linha_prod e
  on e.cod_lin_prod = f.cod_lin_prod
  and e.cod_lin_prod = f.cod_lin_prod
  and e.cod_lin_recei = f.cod_lin_recei
  and e.cod_seg_merc = 0
  and e.cod_cla_uso = 0

join tipo_carteira q
  on q.cod_tip_carteira = nf.tip_carteira

left join representante s
  on s.cod_repres = v.representante

left join cond_pgto_cap u
  on u.cnd_pgto = a.cnd_pgto_nf

left join familia fam
  ON fam.cod_empresa = f.cod_empresa
  AND fam.cod_familia = f.cod_familia

WHERE a.ies_especie_nf = 'NFD'

GROUP BY a.cod_empresa,
  year(a.dat_entrada_nf),
  month(a.dat_entrada_nf),
  a.dat_entrada_nf,
  a.cod_fornecedor,
  c.nom_cliente,
  c.num_telex,
  e.den_tip_cli,
  xx.cod_uni_feder,
  xx.den_cidade,
  pa.den_pais,
  q.den_tip_carteira,
  nf.natureza_operacao,
  g.den_nat_oper,
  a.cnd_pgto_nf,
  u.des_cnd_pgto,
  nf.nota_fiscal,
  s.ped_nf_fatura,
  v.representante,
  v.pct_comissao,
  s.raz_social,
  d.den_estr_linprod,
  e.den_estr_linprod,
  b.cod_item,
  f.den_item,
  f.cod_unid_med,
  f.cod_familia,
  fam.den_familia,

```

```
nf.tip_frete,  
case  
  when nf.tip_frete = 1 then  
    'CIF PAGO'  
  else  
    case  
      when nf.tip_frete = 2 then  
        'CIF COBRADO'  
      else  
        'FOB'  
      end  
    end  
end;
```

APÊNDICE G – RELATÓRIO FATURAMENTO – TUNING

```
--CREATE OR REPLACE VIEW EIS_V_FAT_VANILLA_tuning AS
SELECT distinct 'NOTAS VENDAS' IES_TIPO,
    A.EMPRESA,
    trunc(a.dat_hor_emissao)    dat_hor_emissao,
    A.CLIENTE,
    D.NOM_CLIENTE,
    E.DEN_TIP_CLI,
    F.COD_UNI_FEDER,
    F.DEN_CIDADE,
    H.DEN_PAIS,
    A.NATUREZA_OPERACAO COD_NAT_OPER,
    C.DEN_NAT_OPER,
    A.COND_PAGTO,
    M.DEN_CND_PGTO,
    A.NOTA_FISCAL,
    B.PEDIDO,
    N.PRZ_ENTREGA,
    J.COD_REPRES REPRESENTANTE,
    0 PCT_COMISSAO,
    J.RAZ_SOCIAL,
    EIS_PKG_UTIL_LX.get_aen(L.COD_LIN_PROD) LIN_PROD1,
    EIS_PKG_UTIL_LX.get_aen(L.COD_LIN_PROD, L.COD_LIN_RECEI) LIN_PROD2,
    L.COD_ITEM ITEM,
    L.DEN_ITEM,
    L.COD_UNID_MED,
    l.cod_familia,
    fam.den_familia,
    B.QTD_ITEM,
    B.PRECO_UNIT_LIQUIDO,
    B.VAL_LIQUIDO_ITEM,
    IPI.VAL_TRIBUTO_TOT VAL_IPI,
    ICMS.VAL_TRIBUTO_TOT VAL_ICMS,
    PIS.VAL_TRIBUTO_TOT VAL_PIS,
    COFINS.VAL_TRIBUTO_TOT VAL_COFINS,
    A.TIP_FRETE,
    DECODE(A.TIP_FRETE, 1, 'CIF PAGO', 2, 'CIF COBRADO', 'FOB')
    TIPO_FRETE,
    b.val_fret_consig + b.val_segr_consig + b.val_frete_cliente +
    b.val_seguro_cliente val_frete_rodov,
    ((b.val_desc_item + b.val_desc_merc + b.val_desc_contab +
    b.val_desc_duplicata) * -1) val_desconto,
    b.val_acresc_item + b.val_acre_merc + b.val_acre_duplicata
    val_acrescimo,

    case
        when fat_dacre_item_nf.ind_desc_acre = 'D' then
            fat_dacre_item_nf.val_desc_acre * -1
        else
            fat_dacre_item_nf.val_desc_acre
        end val_desc_acre

FROM FAT_NF_MESTRE A

JOIN FAT_NF_ITEM B
    ON A.EMPRESA = B.EMPRESA
    AND A.TRANS_NOTA_FISCAL = B.TRANS_NOTA_FISCAL
    and a.sit_nota_fiscal='N'

JOIN NAT_OPERACAO C
```

```

    ON A.NATUREZA_OPERACAO = C.COD_NAT_OPER
and C.IES_ESTATISTICA <> 'N'
and a.sit_nota_fiscal='N'

JOIN CLIENTES D
    ON A.CLIENTE = D.COD_CLIENTE

JOIN TIPO_CLIENTE E
    ON D.COD_TIP_CLI = E.COD_TIP_CLI

JOIN CIDADES F
    ON D.COD_CIDADE = F.COD_CIDADE

JOIN UNI_FEDER G
    ON F.COD_UNI_FEDER = G.COD_UNI_FEDER

JOIN PAISES H
    ON G.COD_PAIS = H.COD_PAIS

LEFT JOIN PEDIDOS I
    ON B.EMPRESA = I.COD_EMPRESA
AND B.PEDIDO = I.NUM_PEDIDO

LEFT JOIN REPRESENTANTE J
    ON I.COD_REPRES = J.COD_REPRES

LEFT JOIN TIPO_CARTEIRA K
    ON I.COD_TIP_CARTEIRA = K.COD_TIP_CARTEIRA

LEFT JOIN ITEM L
    ON B.EMPRESA = L.COD_EMPRESA
AND B.ITEM = L.COD_ITEM

LEFT JOIN COND_PGTO M
    ON A.COND_PAGTO = M.COD_CND_PGTO

LEFT JOIN PED_ITENS N
    ON B.EMPRESA = N.COD_EMPRESA
AND B.PEDIDO = N.NUM_PEDIDO
AND B.SEQ_ITEM_PEDIDO = N.NUM_SEQUENCIA

LEFT JOIN FAT_NF_ITEM_FISC ICMS
    ON B.EMPRESA = ICMS.EMPRESA
AND B.TRANS_NOTA_FISCAL = ICMS.TRANS_NOTA_FISCAL
AND B.SEQ_ITEM_NF = ICMS.SEQ_ITEM_NF
AND ICMS.TRIBUTO_BENEF = 'ICMS'
and a.sit_nota_fiscal='N'
AND C.IES_ESTATISTICA <> 'N'

LEFT JOIN FAT_NF_ITEM_FISC IPI
    ON B.EMPRESA = IPI.EMPRESA

```

```

AND B.TRANS_NOTA_FISCAL = IPI.TRANS_NOTA_FISCAL
AND B.SEQ_ITEM_NF = IPI.SEQ_ITEM_NF
AND IPI.TRIBUTO_BENEF = 'IPI'
and a.sit_nota_fiscal='N'
AND C.IES_ESTATISTICA <> 'N'

LEFT JOIN FAT_NF_ITEM_FISC PIS
  ON B.EMPRESA = PIS.EMPRESA
AND B.TRANS_NOTA_FISCAL = PIS.TRANS_NOTA_FISCAL
AND B.SEQ_ITEM_NF = PIS.SEQ_ITEM_NF
AND PIS.TRIBUTO_BENEF = 'PIS_REC'
and a.sit_nota_fiscal='N'
AND C.IES_ESTATISTICA <> 'N'

LEFT JOIN FAT_NF_ITEM_FISC COFINS
  ON B.EMPRESA = COFINS.EMPRESA
AND B.TRANS_NOTA_FISCAL = COFINS.TRANS_NOTA_FISCAL
AND B.SEQ_ITEM_NF = COFINS.SEQ_ITEM_NF
AND COFINS.TRIBUTO_BENEF = 'COFINS_REC'
and a.sit_nota_fiscal='N'
AND C.IES_ESTATISTICA <> 'N'

left join familia fam
  ON fam.cod_empresa = l.cod_empresa
AND fam.cod_familia = l.cod_familia

LEFT JOIN fat_dacre_item_nf
  ON fat_dacre_item_nf.EMPRESA = A.EMPRESA
AND fat_dacre_item_nf.TRANS_NOTA_FISCAL = A.TRANS_NOTA_FISCAL
AND fat_dacre_item_nf.SEQ_ITEM_NF = B.TRANS_NOTA_FISCAL
and a.sit_nota_fiscal='N'
AND C.IES_ESTATISTICA <> 'N'

UNION ALL

SELECT distinct 'NOTAS ENTREGA FUTURA' IES_TIPO,
  A.EMPRESA,
  trunc(a.dat_hor_emissao) dat_hor_emissao,
  A.CLIENTE,
  D.NOM_CLIENTE,
  E.DEN_TIP_CLI,
  F.COD_UNI_FEDER,
  F.DEN_CIDADE,
  H.DEN_PAIS,
  A.NATUREZA_OPERACAO COD_NAT_OPER,
  C.DEN_NAT_OPER,
  A.COND_PAGTO,
  M.DEN_CND_PGTO,
  A.NOTA_FISCAL,
  B.PEDIDO,
  N.PRZ_ENTREGA,
  J.COD_REPRES REPRESENTANTE,
  0 PCT_COMISSAO,
  J.RAZ_SOCIAL,
  EIS_PKG_UTIL_LX.get_aen(L.COD_LIN_PROD) LIN_PROD1,
  EIS_PKG_UTIL_LX.get_aen(L.COD_LIN_PROD, L.COD_LIN_RECEI) LIN_PROD2,
  L.COD_ITEM ITEM,
  L.DEN_ITEM,
  L.COD_UNID_MED,
  l.cod_familia,

```



```

        fam.den_familia,
        B. QTD_ITEM,
        B. PREÇO_UNIT_LIQUIDO,
        B. VAL_LIQUIDO_ITEM,
        IPI.VAL_TRIBUTO_TOT VAL_IPI,
        ICMS.VAL_TRIBUTO_TOT VAL_ICMS,
        PIS.VAL_TRIBUTO_TOT VAL_PIS,
        COFINS.VAL_TRIBUTO_TOT VAL_COFINS,
        A.TIP_FRETE,
        DECODE(A.TIP_FRETE, 1, 'CIF PAGO', 2, 'CIF COBRADO', 'FOB')
TIPO_FRETE,
        b.val_fret_consig + b.val_segr_consig + b.val_frete_cliente +
        b.val_seguro_cliente val_frete_rodov,
        ((b.val_desc_item + b.val_desc_merc + b.val_desc_contab +
        b.val_desc_duplicata) * -1) val_desconto,
        b.val_acresc_item + b.val_acre_merc + b.val_acre_duplicata
val_acrescimo,

        case
            when fat_dacre_item_nf.ind_desc_acre = 'D' then
                fat_dacre_item_nf.val_desc_acre * -1
            else
                fat_dacre_item_nf.val_desc_acre
        end val_desc_acre

FROM FAT_NF_MESTRE A

JOIN FAT_NF_ITEM B
    ON A.EMPRESA = B.EMPRESA
AND A.TRANS_NOTA_FISCAL = B.TRANS_NOTA_FISCAL
and a.sit_nota_fiscal='N'

JOIN NAT_OPERACAO C
    ON A.NATUREZA_OPERACAO = C.COD_NAT_OPER
AND C.COD_NAT_OPER IN (107, 108, 405, 406, 455)

JOIN CLIENTES D
    ON A.CLIENTE = D.COD_CLIENTE

JOIN TIPO_CLIENTE E
    ON D.COD_TIP_CLI = E.COD_TIP_CLI

JOIN CIDADES F
    ON D.COD_CIDADE = F.COD_CIDADE

JOIN UNI_FEDER G
    ON F.COD_UNI_FEDER = G.COD_UNI_FEDER

JOIN PAISES H
    ON G.COD_PAIS = H.COD_PAIS

LEFT JOIN PEDIDOS I
    ON B.EMPRESA = I.COD_EMPRESA
AND B.PEDIDO = I.NUM_PEDIDO
AND I.COD_NAT_OPER IN (107, 108, 405, 406, 455)

LEFT JOIN REPRESENTANTE J
    ON I.COD_REPRES = J.COD_REPRES

```

```

LEFT JOIN TIPO_CARTEIRA K
  ON I.COD_TIP_CARTEIRA = K.COD_TIP_CARTEIRA

LEFT JOIN ITEM L
  ON B.EMPRESA = L.COD_EMPRESA
  AND B.ITEM = L.COD_ITEM

LEFT JOIN COND_PGTO M
  ON A.COND_PAGTO = M.COD_CND_PGTO

LEFT JOIN PED_ITENS N
  ON B.EMPRESA = N.COD_EMPRESA
  AND B.PEDIDO = N.NUM_PEDIDO
  AND B.SEQ_ITEM_PEDIDO = N.NUM_SEQUENCIA

LEFT JOIN FAT_NF_ITEM_FISC ICMS
  ON B.EMPRESA = ICMS.EMPRESA
  AND B.TRANS_NOTA_FISCAL = ICMS.TRANS_NOTA_FISCAL
  AND B.SEQ_ITEM_NF = ICMS.SEQ_ITEM_NF
  AND ICMS.TRIBUTO_BENEF = 'ICMS'

LEFT JOIN FAT_NF_ITEM_FISC IPI
  ON B.EMPRESA = IPI.EMPRESA
  AND B.TRANS_NOTA_FISCAL = IPI.TRANS_NOTA_FISCAL
  AND B.SEQ_ITEM_NF = IPI.SEQ_ITEM_NF
  AND IPI.TRIBUTO_BENEF = 'IPI'

LEFT JOIN FAT_NF_ITEM_FISC PIS
  ON B.EMPRESA = PIS.EMPRESA
  AND B.TRANS_NOTA_FISCAL = PIS.TRANS_NOTA_FISCAL
  AND B.SEQ_ITEM_NF = PIS.SEQ_ITEM_NF
  AND PIS.TRIBUTO_BENEF = 'PIS_REC'

LEFT JOIN FAT_NF_ITEM_FISC COFINS
  ON B.EMPRESA = COFINS.EMPRESA
  AND B.TRANS_NOTA_FISCAL = COFINS.TRANS_NOTA_FISCAL
  AND B.SEQ_ITEM_NF = COFINS.SEQ_ITEM_NF
  AND COFINS.TRIBUTO_BENEF = 'COFINS_REC'

left join familia fam
  ON fam.cod_empresa = l.cod_empresa
  AND fam.cod_familia = l.cod_familia
  and C.IES_ESTATISTICA <> 'N'
  and a.sit_nota_fiscal='N'

LEFT JOIN fat_dacre_item_nf
  ON fat_dacre_item_nf.EMPRESA = A.EMPRESA
  AND fat_dacre_item_nf.TRANS_NOTA_FISCAL = A.TRANS_NOTA_FISCAL
  AND fat_dacre_item_nf.SEQ_ITEM_NF = B.TRANS_NOTA_FISCAL
  and C.IES_ESTATISTICA <> 'N'
  and a.sit_nota_fiscal='N'

WHERE a.natureza_operacao IN (107, 108, 405, 406, 455)

UNION ALL

SELECT distinct 'NOTAS DE COMPLEMENTO' IES_TIPO,
  A.EMPRESA,

```

```

trunc(a.dat_hor_emissao)    dat_hor_emissao,
A.CLIENTE,
D.NOM_CLIENTE,
E.DEN_TIP_CLI,
F.COD_UNI_FEDER,
F.DEN_CIDADE,
H.DEN_PAIS,
A.NATUREZA_OPERACAO COD_NAT_OPER,
C.DEN_NAT_OPER,
A.COND_PAGTO,
M.DEN_CND_PGTO,
A.NOTA_FISCAL,
B.PEDIDO,
N.PRZ_ENTREGA,
J.COD_REPRES REPRESENTANTE,
0 PCT_COMISSAO,
J.RAZ_SOCIAL,
EIS_PKG_UTIL_LX.get_aen(L.COD_LIN_PROD) LIN_PROD1,
EIS_PKG_UTIL_LX.get_aen(L.COD_LIN_PROD, L.COD_LIN_RECEI) LIN_PROD2,
L.COD_ITEM ITEM,
L.DEN_ITEM,
L.COD_UNID_MED,
l.cod_familia,
fam.den_familia,
B.QTD_ITEM,
B.PRECO_UNIT LIQUIDO,
B.VAL_LIQUIDO_ITEM,
IPI.VAL_TRIBUTO_TOT VAL_IPI,
ICMS.VAL_TRIBUTO_TOT VAL_ICMS,
PIS.VAL_TRIBUTO_TOT VAL_PIS,
COFINS.VAL_TRIBUTO_TOT VAL_COFINS,
A.TIP_FRETE,
DECODE(A.TIP_FRETE, 1, 'CIF PAGO', 2, 'CIF COBRADO', 'FOB')
TIPO_FRETE,
b.val_fret_consig + b.val_segr_consig + b.val_frete_cliente +
b.val_seguro_cliente val_frete_rodov,
((b.val_desc_item + b.val_desc_merc + b.val_desc_contab +
b.val_desc_duplicata) * -1) val_desconto,
b.val_acresc_item + b.val_acre_merc + b.val_acre_duplicata
val_acrescimo,

case
when fat_dacre_item_nf.ind_desc_acre = 'D' then
fat_dacre_item_nf.val_desc_acre * -1
else
fat_dacre_item_nf.val_desc_acre
end val_desc_acre

FROM FAT_NF_MESTRE A

JOIN FAT_NF_ITEM B
ON A.EMPRESA = B.EMPRESA
and a.sit_nota_fiscal='N'
AND A.TRANS_NOTA_FISCAL = B.TRANS_NOTA_FISCAL

JOIN NAT_OPERACAO C
ON A.NATUREZA_OPERACAO = C.COD_NAT_OPER
AND C.COD_NAT_OPER IN (118, 115, 165, 226, 405, 406, 455)

JOIN CLIENTES D
ON A.CLIENTE = D.COD_CLIENTE

```

```

JOIN TIPO_CLIENTE E
  ON D.COD_TIP_CLI = E.COD_TIP_CLI

JOIN CIDADES F
  ON D.COD_CIDADE = F.COD_CIDADE

JOIN UNI_FEDER G
  ON F.COD_UNI_FEDER = G.COD_UNI_FEDER

JOIN PAISES H
  ON G.COD_PAIS = H.COD_PAIS

LEFT JOIN PEDIDOS I
  ON B.EMPRESA = I.COD_EMPRESA
  AND B.PEDIDO = I.NUM_PEDIDO
  AND I.COD_NAT_OPER IN (118, 115, 165, 226, 405, 406, 455)

LEFT JOIN REPRESENTANTE J
  ON I.COD_REPRES = J.COD_REPRES

LEFT JOIN TIPO_CARTEIRA K
  ON I.COD_TIP_CARTEIRA = K.COD_TIP_CARTEIRA

LEFT JOIN ITEM L
  ON B.EMPRESA = L.COD_EMPRESA
  AND B.ITEM = L.COD_ITEM

LEFT JOIN COND_PGTO M
  ON A.COND_PAGTO = M.COD_CND_PGTO

LEFT JOIN PED_ITENS N
  ON B.EMPRESA = N.COD_EMPRESA
  AND B.PEDIDO = N.NUM_PEDIDO
  AND B.SEQ_ITEM_PEDIDO = N.NUM_SEQUENCIA

LEFT JOIN FAT_NF_ITEM_FISC ICMS
  ON B.EMPRESA = ICMS.EMPRESA
  AND B.TRANS_NOTA_FISCAL = ICMS.TRANS_NOTA_FISCAL
  AND B.SEQ_ITEM_NF = ICMS.SEQ_ITEM_NF
  AND ICMS.TRIBUTO_BENEF = 'ICMS'

LEFT JOIN FAT_NF_ITEM_FISC IPI
  ON B.EMPRESA = IPI.EMPRESA
  AND B.TRANS_NOTA_FISCAL = IPI.TRANS_NOTA_FISCAL
  AND B.SEQ_ITEM_NF = IPI.SEQ_ITEM_NF
  AND IPI.TRIBUTO_BENEF = 'IPI'

LEFT JOIN FAT_NF_ITEM_FISC PIS
  ON B.EMPRESA = PIS.EMPRESA
  AND B.TRANS_NOTA_FISCAL = PIS.TRANS_NOTA_FISCAL
  AND B.SEQ_ITEM_NF = PIS.SEQ_ITEM_NF
  AND PIS.TRIBUTO_BENEF = 'PIS_REC'

LEFT JOIN FAT_NF_ITEM_FISC COFINS
  ON B.EMPRESA = COFINS.EMPRESA
  AND B.TRANS_NOTA_FISCAL = COFINS.TRANS_NOTA_FISCAL
  AND B.SEQ_ITEM_NF = COFINS.SEQ_ITEM_NF
  AND COFINS.TRIBUTO_BENEF = 'COFINS_REC'

left join familia fam

```

```

        ON fam.cod_empresa = l.cod_empresa
        AND fam.cod_familia = l.cod_familia

LEFT JOIN fat_dacre_item_nf
    ON fat_dacre_item_nf.EMPRESA = A.EMPRESA
    AND fat_dacre_item_nf.TRANS_NOTA_FISCAL = A.TRANS_NOTA_FISCAL
    AND fat_dacre_item_nf.SEQ_ITEM_NF = B.TRANS_NOTA_FISCAL
    and C.IES_ESTATISTICA <> 'N'
    and a.sit_nota_fiscal='N'

WHERE
AND A.natureza_operacao IN (118, 115, 165, 226, 405, 406, 455)

union all

SELECT distinct  'DEVOLUCAO ',
    a.cod_empresa,
    trunc(a.dat_entrada_nf) dat_entrada_nf,
    a.cod_fornecedor,
    c.nom_cliente,
    e.den_tip_cli,
    xx.cod_uni_feder,
    xx.den_cidade,
    pa.den_pais,
    nf.natureza_operacao,
    g.den_nat_oper,
    a.cnd_pgto_nf,
    u.des_cnd_pgto,
    nf.nota_fiscal,
    s.ped_nf_fatura,
    NULL dat_entrega,
    v.representante,
    v.pct_comissao,
    s.raz_social,
    d.den_estr_linprod,
    e.den_estr_linprod,
    b.cod_item,
    f.den_item,
    f.cod_unid_med,
    f.cod_familia,
    fam.den_familia,
    case
        when sum(b.qtd_recebida * -1) = 0 or
             sum(b.qtd_recebida * -1) is null then
            sum(b.qtd_declarad_nf * -1)
        else
            sum(b.qtd_recebida * -1)
    end,
    sum(b.pre_unit_nf * -1),
    sum(b.val_liquido_item * -1),
    sum((b.val_ipi_calc_item + b.val_ipi_desp_aces) * -1) val_ipi,
    sum(b.val_icms_item_c * -1) val_icms,
    sum(j.val_pis * -1) val_pis,
    sum(j.val_cofins * -1) val_cofins,
    nf.tip_frete,
    case
        when nf.tip_frete = 1 then
            'CIF PAGO'
        else

```

```

        case
        when nf.tip_frete = 2 then
            'CIF COBRADO'
        else
            'FOB'
        end
    end,
    sum(nf.val_frete_rodov * -1) val_frete_rodov,
    0 val_desconto,
    0 val_acrescimo,
    0 val_desc_acre

from nf_sup a

join aviso_rec b
    on a.cod_empresa = b.cod_empresa
    and a.num_aviso_rec = b.num_aviso_rec
    and a.ies_especie_nf = 'NFD'

join clientes c
    on a.cod_fornecedor = c.cod_cliente

join cidades xx
    on c.cod_cidade = xx.cod_cidade

join uni_feder yy
    on yy.cod_uni_feder = xx.cod_uni_feder

join tipo_cliente e
    on c.cod_tip_cli = e.cod_tip_cli

join item f
    on b.cod_empresa = f.cod_empresa
    and b.cod_item = f.cod_item

left join obf_dvcli_piscofin j
    on j.empresa = b.cod_empresa
    and j.aviso_recebto = b.num_aviso_rec
    and j.seq_aviso_recebto = b.num_seq

left join paises pa
    on pa.cod_pais = yy.cod_pais

left join sup_nf_devol_cli s
    on b.cod_empresa = s.empresa
    and b.num_aviso_rec = s.aviso_recebto
    and b.num_seq = s.seq_aviso_recebto
    and a.ies_especie_nf = 'NFD'

join fat_nf_mestre nf
    on nf.empresa = s.empresa
    and nf.nota_fiscal = s.nota_fiscal_fatura
    and nf.serie_nota_fiscal = s.ser_nf_fatura
    and nf.sit_nota_fiscal='N'

left join fat_nf_repr v
    on (v.empresa = nf.empresa and
        v.trans_nota_fiscal = nf.trans_nota_fiscal and
        v.seq_representante = 1)
    and nf.sit_nota_fiscal='N'

```

```

join nat_operacao g
  on nf.natureza_operacao = g.cod_nat_oper
  and g.ies_estatistica in ('T', 'V')

join linha_prod d
  on d.cod_lin_prod = f.cod_lin_prod
  and d.cod_lin_recei = 0
  and d.cod_seg_merc = 0
  and d.cod_cla_uso = 0

join linha_prod e
  on e.cod_lin_prod = f.cod_lin_prod
  and e.cod_lin_prod = f.cod_lin_prod
  and e.cod_lin_recei = f.cod_lin_recei
  and e.cod_seg_merc = 0
  and e.cod_cla_uso = 0

join tipo_carteira q
  on q.cod_tip_carteira = nf.tip_carteira

left join representante s
  on s.cod_repres = v.representante

left join cond_pgto_cap u
  on u.cnd_pgto = a.cnd_pgto_nf

left join familia fam
  ON fam.cod_empresa = f.cod_empresa
  AND fam.cod_familia = f.cod_familia

WHERE
a.ies_especie_nf = 'NFD'

GROUP BY a.cod_empresa,
  year(a.dat_entrada_nf),
  month(a.dat_entrada_nf),
  a.dat_entrada_nf,
  a.cod_fornecedor,
  c.nom_cliente,
  c.num_telex,
  e.den_tip_cli,
  xx.cod_uni_feder,
  xx.den_cidade,
  pa.den_pais,
  q.den_tip_carteira,
  nf.natureza_operacao,
  g.den_nat_oper,
  a.cnd_pgto_nf,
  u.des_cnd_pgto,
  nf.nota_fiscal,
  s.ped_nf_fatura,
  v.representante,
  v.pct_comissao,
  s.raz_social,
  d.den_estr_linprod,
  e.den_estr_linprod,
  b.cod_item,
  f.den_item,
  f.cod_unid_med,
  f.cod_familia,
  fam.den_familia,

```

```
nf.tip_frete,  
case  
  when nf.tip_frete = 1 then  
    'CIF PAGO'  
  else  
    case  
      when nf.tip_frete = 2 then  
        'CIF COBRADO'  
      else  
        'FOB'  
      end  
    end  
end;  
;
```


APÊNDICE H – RELATÓRIO MOVIMENTAÇÃO ESTOQUE

--CREATE OR REPLACE VIEW EIS_V_SUP0710 AS

```
SELECT A.COD_EMPRESA,
       A.NUM_TRANSAC,
       A.COD_ITEM,
       B.DEN_ITEM,
       EIS_PKG_UTIL_LX.get_aen(B.COD_LIN_PROD) LIN_PROD,
       EIS_PKG_UTIL_LX.get_aen(B.COD_LIN_PROD,B.COD_LIN_RECEI) LIN_RECEI,

       EIS_PKG_UTIL_LX.get_aen(B.COD_LIN_PROD,B.COD_LIN_RECEI,B.COD_SEG_MERC)
       SEG_MERC,

       EIS_PKG_UTIL_LX.get_aen(B.COD_LIN_PROD,B.COD_LIN_RECEI,B.COD_SEG_MERC,B.COD
       _CLA_USO) CLA_USO,
       A.DAT_MOVTO,
       A.COD_OPERACAO,
       D.DEN_OPERACAO,
       A.NUM_DOCUM,
       A.NUM_SEQ,
       DECODE(A.IES_TIP_MOVTO, 'N', 'NORMAL',
              'R', 'REVERSAO',
              'L', 'NORMAL',
              'NAO TRATADO') IES_TIP_MOVTO,

       case when a.ies_tip_movto = 'R'
       then A.QTD_MOVTO * -1
       else A.QTD_MOVTO
       end qtd_movto,
       A.CUS_UNIT_MOVTO_P,
       case when a.ies_tip_movto = 'R'
       then A.CUS_TOT_MOVTO_P * -1
       else A.CUS_TOT_MOVTO_P
       end cus_tot_movto_p,
       A.CUS_UNIT_MOVTO_F,
       case when a.ies_tip_movto = 'R'
       then A.CUS_TOT_MOVTO_F * -1
       else A.CUS_TOT_MOVTO_F
       end cus_tot_movto_f,
       A.NUM_CONTA,
       A.NUM_SECAO_REQUIS,
       A.COD_LOCAL_EST_ORIG,
       A.NUM_LOTE_ORIG,
       A.COD_LOCAL_EST_DEST,
       A.NUM_LOTE_DEST,
       A.IES_SIT_EST_ORIG,
       A.IES_SIT_EST_DEST,
       A.COD_TURNNO,
       OBS.TEX_OBSERV,
       FAM.DEN_FAMILIA
FROM ESTOQUE_TRANS A

LEFT JOIN ITEM B
  ON A.COD_EMPRESA = B.COD_EMPRESA
 AND A.COD_ITEM = B.COD_ITEM

LEFT JOIN EMPRESA C
  ON A.COD_EMPRESA = C.COD_EMPRESA

LEFT JOIN ESTOQUE_OPERAC D
  ON A.COD_EMPRESA = D.COD_EMPRESA
 AND A.COD_OPERACAO = D.COD_OPERACAO
```

```
LEFT JOIN LOCAL LOCAL_ORIG
  ON A.COD_LOCAL_EST_ORIG = LOCAL_ORIG.COD_LOCAL
  AND A.COD_EMPRESA = LOCAL_ORIG.COD_EMPRESA

LEFT JOIN LOCAL LOCAL_DEST
  ON A.COD_LOCAL_EST_DEST = LOCAL_DEST.COD_LOCAL
  AND A.COD_EMPRESA = LOCAL_DEST.COD_EMPRESA

LEFT JOIN ESTOQUE_OBS OBS
  ON A.COD_EMPRESA = OBS.COD_EMPRESA
  AND A.NUM_TRANSAC = OBS.NUM_TRANSAC

LEFT JOIN FAMILIA FAM
  ON B.COD_FAMILIA = FAM.COD_FAMILIA
  AND B.COD_EMPRESA = FAM.COD_EMPRESA
```

APÊNDICE I – RELATÓRIO MOVIMENTAÇÃO ESTOQUE – TUNING

```
--CREATE OR REPLACE VIEW EIS_V_SUP0710_TUNING AS
SELECT A.COD_EMPRESA,
       A.NUM_TRANSAC,
       A.COD_ITEM,
       B.DEN_ITEM,
       EIS_PKG_UTIL_LX.get_aen(B.COD_LIN_PROD) LIN_PROD,
       EIS_PKG_UTIL_LX.get_aen(B.COD_LIN_PROD,B.COD_LIN_RECEI) LIN_RECEI,

EIS_PKG_UTIL_LX.get_aen(B.COD_LIN_PROD,B.COD_LIN_RECEI,B.COD_SEG_MERC)
SEG_MERC,

EIS_PKG_UTIL_LX.get_aen(B.COD_LIN_PROD,B.COD_LIN_RECEI,B.COD_SEG_MERC,B.COD
_CLA_USO) CLA_USO,
       A.DAT_MOVTO,
       A.COD_OPERACAO,
       D.DEN_OPERACAO,
       A.NUM_DOCUM,
       A.NUM_SEQ,
       DECODE(A.IES_TIP_MOVTO, 'N', 'NORMAL',
              'R', 'REVERSAO',
              'L', 'NORMAL',
              'NAO TRATADO') IES_TIP_MOVTO,

       case when a.ies_tip_movto = 'R'
         then A.QTD_MOVTO * -1
         else A.QTD_MOVTO
       end qtd_movto,
       A.CUS_UNIT_MOVTO_P,
       case when a.ies_tip_movto = 'R'
         then A.CUS_TOT_MOVTO_P * -1
         else A.CUS_TOT_MOVTO_P
       end cus_tot_movto_p,
       A.CUS_UNIT_MOVTO_F,
       case when a.ies_tip_movto = 'R'
         then A.CUS_TOT_MOVTO_F * -1
         else A.CUS_TOT_MOVTO_F
       end cus_tot_movto_f,
       A.NUM_CONTA,
       A.NUM_SECAO_REQUIS,
       A.COD_LOCAL_EST_ORIG,
       A.NUM_LOTE_ORIG,
       A.COD_LOCAL_EST_DEST,
       A.NUM_LOTE_DEST,
       A.IES_SIT_EST_ORIG,
       A.IES_SIT_EST_DEST,
       A.COD_TURNNO,
       OBS.TEX_OBSERV,
       FAM.DEN_FAMILIA
FROM ESTOQUE_TRANS A

JOIN ITEM B
  ON A.COD_EMPRESA = B.COD_EMPRESA
 AND A.COD_ITEM = B.COD_ITEM
 and b.ies_situacao='A'

JOIN ESTOQUE_OPERAC D
  ON A.COD_EMPRESA = D.COD_EMPRESA
 AND A.COD_OPERACAO = D.COD_OPERACAO

JOIN ESTOQUE_OBS OBS
```

```
ON A.COD_EMPRESA = OBS.COD_EMPRESA  
AND A.NUM_TRANSAC = OBS.NUM_TRANSAC
```

```
JOIN FAMILIA FAM  
ON B.COD_FAMILIA = FAM.COD_FAMILIA  
AND B.COD_EMPRESA = FAM.COD_EMPRESA
```