

Todos os direitos autorais reservados pela **TOTVS S.A.**

Proibida a reprodução total ou parcial, bem como a armazenagem em sistema de recuperação e a transmissão, de qualquer modo ou por qualquer outro meio, seja este eletrônico, mecânico, de fotocópia, de gravação, ou outros, sem prévia autorização por escrito da proprietária.

O desrespeito a essa proibição configura em apropriação indevida dos direitos autorais e patrimoniais da TOTVS.

Conforme artigos 122 e 130 da LEI no. 5.988 de 14 de Dezembro de 1973.

Consultas SQL e TOTVS

Inteligência de Negócios

RM – Versão 12

Habilitar o participante nas funcionalidades do ambiente de forma sistemática e prática, abordando diversas opções oferecidas pelo sistema, tais como: Extração de dados, criação de cubos, criação de gráficos painel de metas e passagem de parâmetros.



Sumário

1. Objetivo.....	3
2. Família de comandos SQL	3
3. Comando SELECT	3
4. Funções de Agrupamento.....	6
4.1. Função COUNT	6
4.2. Função SUM.....	7
4.3. Função AVG	7
4.4. Função MAX.....	7
4.5. Função MIN	8
5. Cláusula WHERE	8
6. Cláusula TOP	8
7. Operadores.....	9
8. Cláusula GROUP BY	12
9. Cláusula GROUP BY	13
10. Cláusula ORDER BY	13
11. Cláusula HAVING	14
12. Cláusula UNION	15
13. Alias de Coluna (Apelidos).....	15
14. Consultas com Junção (duas ou mais tabelas – INNER JOIN, ligação)	18
15. Alias de Tabela (Apelido).....	20
16. SubQuery.....	21
17. OUTER JOINS.....	21
18. Ferramenta Visões de Dados (Consulta SQL).....	24
18.1. Editor Consulta SQL	24
18.2. Recursos.....	28
18.3. Inclusão de consultas SQL em anexos.....	32
18.4. Exemplos de Aplicação das Sentenças	34
19. TOTVS Inteligência de Negócios	36
19.1. Categoria	36
19.2. Cubo	36
19.3. Roteiro para criação de cubos	37
19.4. Criação de gráfico.....	39
20. Painel de Metas	42
21. Conclusão.....	50

1. Objetivo

Linguagem SQL é uma linguagem estruturada de consulta e manipulação dos elementos no banco de dados. Um padrão internacional inicialmente desenvolvido pela IBM, agora coordenado pela ANSI-standard, sendo que sua última publicação foi em 1992.

Tem como características:

- Padronizada para os bancos de dados relacionais.
- Cada comando é uma descrição do que se deseja obter; quem executa o comando é o gerenciador do banco de dados.

Atualmente é uma linguagem muito utilizada em vários processos do aplicativo TOTVS RM e por este motivo vamos estudá-la, bem como sua aplicação no sistema e a sua utilização no módulo Inteligência de Negócios.

2. Família de comandos SQL

DDL - Linguagem de Definição de Dados

Comandos de manipulação da estrutura do banco de dados (create, alter, drop, rename).

DML - Linguagem de Manipulação de Dados

Comandos de manipulação dos registros (dados) do banco de dados (insert, select, update, delete).

DCL - Linguagem de Controle de Dados

Comandos de manipulação de acesso a estrutura/registros do banco de dados (grant, revoke).

A família de comandos DML está associada diretamente ao dia-a-dia do Técnico de Suporte, principalmente, o comando SELECT. Tendo em vista que os aplicativos do RM têm a função de inserir dados (insert), deletar dados (delete) e modificar dados (update), não é aconselhável executarmos tais procedimentos através do software do banco de dados. No entanto, temos total liberdade de nos aprofundarmos na exploração dos recursos, que o comando SELECT nos oferece para extração de dados, convertendo-os em uma rica fonte de consulta.

3. Comando SELECT

Seleciona linhas e colunas de uma tabela;

Na estrutura do comando SELECT, cada parte do comando é chamada de Cláusula.

Cláusula SELECT: Lista colunas (campos) para visualização;

Cláusula FROM: Tabela(s) de procura;

Cláusula WHERE: Lista de condições;

Cláusula GROUP BY: Colunas para serem agregadas;

Cláusula HAVING: Especifica as Condições do uso de agrupamento (funções);

Cláusula ORDER BY: Lista de coluna(s) para ordenação [ASC (ascendente)] DESC (descendente)]. Para efeitos didáticos dividiremos o comando SELECT em dois tipos de consultas:

- 1) Consultas simples (em uma única tabela).
- 2) Consultas com junção (em duas ou mais tabelas).

SINTAXE

SELECT (Lista de Colunas);

Lista de Colunas

São as colunas (Campos) a serem selecionadas na tabela, podendo ser separadas por vírgula. Caso você deseje ver todos os campos da tabelas use o caractere * (asterisco).

Argumentos

DISTINCT

Define a visualização somente dos campos com valores distintos, ou seja, sem repetição;

ALL

Define que todos os registros serão selecionados, este argumento é o default, assim sendo, não é necessário escrevê-lo na sentença de consulta.

Cláusula FROM

Tabelas com os dados a serem pesquisados;

Lista de Tabelas

É a lista de tabelas, cujos registros serão selecionados. Se desejar selecionar mais de uma tabela ou mais de um campo, separe-os por vírgula.

Executando sentenças de consultas SQL

O usuário terá à disposição em todos os aplicativos RM, um ícone para acesso ao editor de consultas. No exemplo, utilizaremos o Aplicativo TOTVS Folha de Pagamento, no entanto poderá ser acessado de qualquer aplicativo.

Clique no botão correspondente na barra de ferramentas do aplicativo ou acesse o UTILITÁRIOS / CONSULTAS SQL. Pela nova MDI acesse GESTÃO / VISÕES DE DADOS.

Será apresentada a janela de visão de consultas. Para inserir uma nova consulta, basta clicar no ícone “Novo”.

Como exemplo, utilizaremos inicialmente a consulta de seleção de funcionários:

```
SELECT NOME, SALARIO FROM PFUNC
```

Campos selecionados: Nome e salário.

Tabela a ser pesquisada: PFUNC, tabela de funcionários do aplicativo TOTVS Folha de Pagamento.

Para verificar o resultado da consulta, o usuário deve pressionar o botão Executar. Será apresentado o resultado da consulta SQL confeccionada.

Algumas sentenças serão muito utilizadas para localização das tabelas e colunas do RM.

Exemplos:

```
SELECT * FROM GSISTEMA
```

Lista todos os dados dos sistemas RM.

NOME DO SISTEMA	CÓDIGO DO SISTEMA	DESCRICAO
RM CHRONUS	A	TOTVS Automação de Ponto
RM TESTIS	B	TOTVS Avaliação e Pesquisa
RM SALDUS	C	TOTVS Gestão Contábil
RM LIBER	D	TOTVS Gestão Fiscal
RM CLASSIS - E	E	Ensino Básico
RM FLUXUS	F	TOTVS Gestão Financeira
RM BIS	G	TOTVS Inteligência de Negócios
RM AGILIS	H	TOTVS Gestão do Relacionamento com o Cliente
RM BONUM	I	TOTVS Gestão Patrimonial
RM PLANOS	J	Gestão de Planos de Saúde
RM FACTOR	K	TOTVS Planejamento e Controle da Produção
RM BIBLIOS	L	TOTVS Gestão Bibliotecária
RM SOLUM	M	TOTVS Construção e Projetos
RM OFFICINA	N	TOTVS Manutenção
RM Saude/Janus	O	Gestão de Hospitais, Clínicas e Centros de Diagnóstico
RM LABORE	P	TOTVS Folha de Pagamento
RM SSO	R	TOTVS Segurança e Saúde Ocupacional
RM CLASSIS NET	S	TOTVS Educacional
RM NUCLEUS	T	TOTVS Gestão de Estoque, Compras e Faturamento
RM CLASSIS - U	U	Ensino Superior
RM VITAE	V	TOTVS Gestão de Pessoas
RM PORTAL	W	TOTVS Gestão de Conteúdos
RM SGI	X	TOTVS Incorporação

SELECT * FROM GCAMPOS

Ideal para localização das tabelas e colunas do RM.

TABELA	COLUNA	DESCRICAO	RELATORIO	APLICACOES
PFUNC	ISENTOIRRF	Isento IRRF	1	P;G;
PFUNC	JORNADA	Não Mais Utilizado pelo Sistema	0	P;N;M;K;G;
PFUNC	JORNADAMENSAL	Jornada Mensal de Trabalho	1	P;G;N;M;K;
PFUNC	LOCALTRABCODMUNICPIO	Local de Trabalho (Código Município)	1	P;G;

SELECT * FROM GCAMPOS WHERE DESCRICAO = 'FUNCIONARIOS'

SELECT * FROM GCAMPOS WHERE TABELA = 'PFUNC' AND DESCRICAO = 'SALARIO'

Importante/Saiba Mais

- A coluna RELATORIO: 1 é para campos da base visíveis no relatório e 0 para campos da base não aceitos no relatório.

SELECT * FROM GLINKSREL

Lista todos os relacionamentos das tabelas.

MASTERTABLE ▲ ▼	CHILDTABLE	MASTERFIELD	CHILDFIELD	RECCREAT...	RECCREAT...	RECMODIF...	RECMODIF...
PSECAO	PFSUBSTITUICAO	CODCOLIGADA,CODIGO	CODCOLIGADA,CODSECAO				
PSECAO	PFUNC	CODCOLIGADA,CODIGO	CODCOLIGADA,CODSECAO				
PSECAO	PGPSDESONERACAO	CODCOLIGADA,CODIGO	CODCOLIGADA,CODSECAO				
PSECAO	PGPSLIMINAR	CODCOLIGADA,CODIGO	CODCOLIGADA,CODSECAO				

4. Funções de Agrupamento

Funções que retornam resultados baseados em grupos de linhas. Por padrão todas as linhas de uma tabela são tratadas como um grupo.

As funções que serão estudadas neste curso são:

- 1) COUNT (contar);
- 2) AVG (média);
- 3) MAX (máximo);
- 4) MIN (mínimo);
- 5) SUM (somar)

Importante/Saiba Mais

- As funções de agrupamento ignoram valores nulos. A única exceção é a função COUNT, que em alguns tipos de aplicação poderão considerar também os valores nulos.
- Usando o DISTINCT, a função só atua nos valores diferentes.

4.1. Função COUNT

Retorna o número de itens de um grupo.

Exemplos:

Utilizando Tabelas do TOTVS Gestão de Estoques, Compras e Faturamento

```
SELECT COUNT (*) FROM TPRODUTO
```

Lista a quantidade de produtos existentes na tabela.

```
SELECT COUNT (DISTINCT PRECO1) FROM TPRODUTODEF
```

Lista a quantidade de produtos que não tenham preços repetidos.

```
SELECT COUNT(CHAPA) FROM PFUNC
```

Lista a quantidade de funcionários.

4.2. Função SUM

Soma todos os valores da coluna selecionada, os valores nulo são ignorados.

Exemplos:

Utilizando Tabelas do TOTVS Folha de Pagamento

```
SELECT SUM (SALARIO) FROM PFUNC
```

Lista o total (soma) de salários de todos os funcionários da tabela.

```
SELECT SUM (DISTINCT SALARIO) FROM PFUNC
```

Lista o total (soma) de salários não repetidos de todos os funcionários da tabela.

```
SELECT CODSECAO, SUM(SALARIO) FROM PFUNC GROUP BY CODSECAO
```

Lista a soma de salários agrupada por Seção.

4.3. Função AVG

Calcula a média aritmética dos valores de uma coluna, os valores nulos são ignorados.

Exemplos:

```
SELECT AVG(SALARIO) FROM PFUNC
```

Lista a média de salários. Primeiramente ele soma os salários de cada funcionário e depois divide o total encontrado pela quantidade de registros.

```
SELECT AVG (DISTINCT SALARIO) FROM PFUNC
```

Lista a média dos salários não repetidos da tabela.

```
SELECT CODFILIAL, AVG(SALARIO) FROM PFUNC GROUP BY CODFILIAL
```

Lista a média de salários agrupada por Filial.

4.4. Função MAX

Lista o maior valor de uma coluna.

Exemplos:

```
SELECT MAX(SALARIO) FROM PFUNC
```

Lista o maior salário da tabela.

```
SELECT CODFILIAL, MAX(SALARIO)  
FROM PFUNC  
GROUP BY CODFILIAL
```

Lista o maior salário da tabela agrupado por Filial.

4.5. Função MIN

Lista o menor valor de uma coluna, valor Nulo é ignorado.

Exemplo:

```
SELECT MIN(SALARIO) FROM PFUNC
```

Lista o menor salário da tabela.

```
SELECT CODSECAO, MIN(SALARIO)
FROM PFUNC
GROUP BY CODSECAO
```

Lista o menor salário da tabela agrupado por Seção.

5. Cláusula WHERE

Determina uma condição que os registros devem satisfazer, para serem selecionados em uma ou mais tabelas.

Exemplos:

```
SELECT NOME, SALARIO
FROM PFUNC
WHERE SALARIO <=500.00
```

Lista todos os nomes e salários de funcionários com salário menor ou igual 500,00 reais.

Utilizando Tabelas do TOTVS Gestão Financeira.

```
SELECT IDLAN, CODCFO, DATAVENCIMENTO, VALORORIGINAL
FROM FLAN
WHERE DATAVENCIMENTO>='01/01/2004'
```

6. Cláusula TOP

A cláusula TOP permite controlar o número de linhas a serem devolvidas ou que são afetadas por uma consulta. Percebemos melhor a utilização dessa cláusula quando desejamos retornar um subconjunto de linhas de um conjunto de resultados.

Exemplos:

```
SELECT TOP 10 NOME, DATADEMISSAO FROM PFUNC ORDER BY DATADEMISSAO DESC
```

Lista os dez funcionários com data de demissão em ordem decrescente.

```
SELECT TOP 10 PERCENT NOME, SALARIO FROM PFUNC ORDER BY SALARIO DESC
```

Lista dez por cento de todos os funcionários com os maiores salários em ordem decrescente.

7. Operadores

Há diversos tipos de operadores. Quando mais de um tipo de operador está presente em uma expressão, eles são avaliados segundo uma ordem de precedência, descrita na lista a seguir:

1. (+) Positivo, - Negativo, ~ NOT
2. (*) Multiplicação, (/) Divisão, (%) Módulo
3. (+) ADD e Concatenação, (-) Subtração
4. =, <, >, >=, <=, <!, >!, < (Comparação)
5. NOT
6. AND
7. ALL, BETWEEN, IN, LIKE, OR

(+) ADD:

Exemplo:

```
SELECT NOME, SALARIO + 400  
FROM PFUNC
```

Seleciona o nome dos funcionários, adicionando 400 reais no salário de todos.

CONCATENAÇÃO:

Exemplo:

```
SELECT NOME, RUA+ ',' + NUMERO  
FROM PPESSOA
```

Seleciona o nome dos funcionários, unindo as colunas Rua e Número.

DIVISÃO

Exemplo:

```
SELECT NOME, SALARIO / 20  
FROM PFUNC
```

Seleciona o nome e o salário dos funcionários dividido por 20.

A Tabela abaixo mostra os operadores aceitos em comparações:

OPERADOR	COMPARAÇÃO
Igualdade	=
Desigualdade	< >
Maior que	>
Menor que	<
Maior ou igual a	> =

Menor ou igual a	< =
Não maior que	! >
Não menor que	! <
Não igual a	! =

Exemplos de consulta com os Operadores de Comparação:**Igual a (=)**

```
SELECT NOME, SALARIO  
FROM PFUNC  
WHERE SALARIO = 580
```

Lista todos os nomes e salários dos funcionários, com salário igual a R\$ 580,00.

Diferente de (<>)

```
SELECT NOME, CODSECAO  
FROM PFUNC  
WHERE CODSECAO <> '01.01'
```

Lista todos os nomes e códigos de seção dos funcionários, exceto os da seção 01.01.

Maior que (>)

```
SELECT NOME, SALARIO  
FROM PFUNC  
WHERE SALARIO >500
```

Lista todos os nomes e salários dos funcionários, com salário maior que R\$ 500,00.

Maior ou igual a (>=)

```
SELECT NOME, SALARIO  
FROM PFUNC  
WHERE SALARIO >=500
```

Lista todos os nomes e salários dos funcionários, com salário maior ou igual a R\$ 500,00.

Menor que (<)

```
SELECT NOME, SALARIO  
FROM PFUNC  
WHERE SALARIO <500
```

Lista todos os nomes e salários dos funcionários, com salário menor que R\$ 500,00.

Menor ou igual a (<=)

```
SELECT NOME, SALARIO  
FROM PFUNC  
WHERE SALARIO <=500
```

Lista todos os nomes e salários dos funcionários, com salário menor ou igual a R\$ 500,00.

IN: Compara um valor com uma lista de valores.

```
SELECT NOME, SALARIO  
FROM PFUNC  
WHERE SALARIO IN (500, 700, 800, 900)
```

Lista todos os nomes e salários dos funcionários, com salários iguais a R\$ 500,00, R\$ 700,00, R\$ 800,00 e R\$ 900,00.

BETWEEN: Permite procurar dados dentro de um intervalo.

```
SELECT NOME, SALARIO  
FROM PFUNC  
WHERE SALARIO BETWEEN 1000 AND 1500
```

Lista todos os nomes e salários dos funcionários que tem salário entre R\$ 1000,00 e R\$ 1500,00.

```
SELECT CHAPA,NOME,DATAADMISSAO  
FROM PFUNC  
WHERE DATAADMISSAO BETWEEN '01/01/2014' AND '12/31/2014'
```

Lista todas chapas, nomes e data de admissão dos funcionários que foram admitidos entre 01/01/2014 a 31/12/2014.

LIKE: Permite procurar strings (parte do contexto) que atendam determinado padrão de pesquisa.

%(Percentual) - Substitui por um ou mais caracteres. _(Underscore) - Substitui por um único caractere.

Exemplos:

```
SELECT NOME  
FROM PFUNC  
WHERE NOME LIKE 'M%'
```

Lista todos os nomes de funcionários iniciados com a letra 'M'.

```
SELECT NOME  
FROM PFUNC  
WHERE NOME LIKE '%O'
```

Lista todos os nomes de funcionários terminados com a letra 'O'.

```
SELECT NOME  
FROM PFUNC  
WHERE NOME LIKE 'J_A%'
```

Lista todos os nomes de funcionários iniciados com a letra 'J', 1 caractere qualquer seguido da letra 'A' e, terminados com quaisquer combinações de nomes.

```
SELECT NOME  
FROM PFUNC  
WHERE NOME NOT LIKE 'MARIA_%'
```

Lista todos os nomes de funcionários exceto as 'MARIAS'.

IS NULL: Testa valores que são nulos.

```
SELECT NOME, DATADEMISSAO  
FROM PFUNC
```

WHERE DATADEMISSAO IS NULL

Lista os nomes de funcionários que tenham data de demissão nula (sem dado).

NOT: Inverte o resultado de uma condição.

```
SELECT NOME, DATADEMISSAO
FROM PFUNC
WHERE DATADEMISSAO IS NOT NULL
```

Podemos negar expressões com os seguintes operadores:

<>, NOT, IS NOT NULL, NOT EXISTS, NOT LIKE, NOT IN.

8. Cláusula GROUP BY

Aprenderemos neste artigo a trabalhar com funções de data no SQL Server. As principais são: GETDATE, DATEPART, DATEADD e DATEDIFF.

Um detalhe importante é que as funções de data trabalham referenciando unidades de data. As mais comuns são:

YEAR(ano)

Exemplo:

```
SELECT NUMERODOCUMENTO, DATAVENCIMENTO, VALORORIGINAL
FROM FLAN
WHERE YEAR (DATAVENCIMENTO) = 2013
```

Lista o número do documento, data de vencimento e valor original dos lançamentos financeiros vencidos em 2013.

MONTH(mês)

Exemplo:

```
SELECT NUMERODOCUMENTO, DATAVENCIMENTO, VALORORIGINAL
FROM FLAN
WHERE YEAR (DATAVENCIMENTO) = 2013 AND MONTH (DATAVENCIMENTO) = 1
```

Lista o número do documento, data de vencimento e valor original dos lançamentos financeiros vencidos em janeiro de 2013.

DAY(dia)

Exemplo:

```
SELECT MONTH(DTNASCIMENTO) MÊS, DAY(DTNASCIMENTO) DIA
FROM PPESSOA
```

Lista o mês e o dia de aniversário das pessoas (cadastro de pessoas).

GETDATE ()

A função GETDATE retorna a data e a hora atuais do sistema. Podemos digitar o seguinte comando para obter a data e hora atuais:

```
SELECT GETDATE ()
```

DATEPART (unidade, data)

A função DATEPART retorna a parte especificada de uma data como um inteiro. Observe os exemplos:

```
SELECT DATEPART (YEAR, '02/01/2004') Resposta: 2004
```

DATEADD (unidade, numero_unid, data)

A função DATEADD retorna uma nova data através da soma do número de unidades especificadas pelo valor unidade a uma data. Observe os exemplos:

```
SELECT DATEADD ( DAY ,6, '02/01/2004' ) Resposta: 2004-02-07
```

```
SELECT DATEADD ( MONTH ,6, '02/01/2004' ) Resposta: 2004-08-01
```

```
SELECT DATEADD ( YEAR ,6, '02/01/2004' ) Resposta: 2010-02-01
```

DATEDIFF (unidade, data1, data2)

A função DATEDIFF calcula a diferença entre as datas data2 e data1, retornando o resultado como um inteiro, cuja unidade é definida pelo valor unidade. Observe os exemplos:

```
SELECT DATEDIFF ( DAY , '02/01/2004' , '05/25/2004' )  
Resposta: 114 (dias)
```

```
SELECT DATEDIFF ( MONTH , '02/01/2004' , '05/25/2004' )  
Resposta: 3 (meses)
```

```
SELECT DATEDIFF ( YEAR , '02/01/2004' , '05/25/2006' )  
Resposta: 2 (anos)
```

9. Cláusula GROUP BY

Agrupar linhas de uma tabela dentro de grupos menores. É utilizada em conjunto com as funções de grupo.

Exemplo:

```
SELECT CODSECAO, AVG(SALARIO)  
FROM PFUNC  
GROUP BY CODSECAO  
Lista a média salarial agrupada por Seção.
```

```
SELECT CODFILIAL, COUNT(CHAPA)  
FROM PFUNC  
GROUP BY CODFILIAL  
Lista a quantidade de funcionários agrupada por Filial.
```

10. Cláusula ORDER BY

Define a ordem das linhas do resultado de uma pesquisa. Deve ser a última cláusula do SELECT. Utiliza-se ASC (ordem crescente) e DESC (ordem decrescente). O padrão de ordenação é ASC.

Ordem default:

- 1º) Valores numéricos menores;
- 2º) Valores de datas mais antigas;
- 3º) Caracteres em ordem alfabética;

Para alterar a ordem default: Colocar DESC depois do nome da coluna na cláusula ORDER BY.

Importante/Saiba Mais

- Os valores nulos são mostrados por último para sequências ascendentes.

```
SELECT NOME, SALARIO  
FROM PFUNC  
ORDER BY NOME ASC
```

Lista todos os registros da tabela pfunc, apresentando somente as colunas com os nomes (em ordem alfabética) e salários dos funcionários.

```
SELECT NOME, SALARIO  
FROM PFUNC  
ORDER BY NOME DESC
```

Lista todos os registros da tabela pfunc, apresentando somente as colunas com os nomes (em ordem decrescente) e salários dos funcionários.

```
SELECT CODFILIAL, CHAPA, NOME, DATAADMISSAO  
FROM PFUNC  
ORDER BY DATAADMISSAO
```

Lista todos os registros da tabela pfunc, apresentando somente as colunas com código da filial, as chapas, os nomes e data de admissão (em ordem crescente) dos funcionários.

11. Cláusula HAVING

Usada para selecionar os grupos que serão mostrados, com base nas informações das funções de grupo.

Exemplos:

```
SELECT PSECAO.CODIGO, SUM(SALARIO), MAX(SALARIO), COUNT(CHAPA)  
FROM PSECAO, PFUNC  
WHERE PSECAO.CODIGO=PFUNC.CODSECAO  
GROUP BY PSECAO.CODIGO  
HAVING MIN(SALARIO)>1000
```

Mostra o código da seção, a soma dos salários, o maior salário e o número de funcionários, desde que o menor salário seja superior a 1000.

```
SELECT CODFILIAL, COUNT(CHAPA)  
FROM PFUNC  
GROUP BY CODFILIAL  
HAVING COUNT(CHAPA) > 20
```

Mostra o código da filial, a quantidade de funcionários, desde que o número de funcionários seja superior a 20.

12. Cláusula UNION

Usada para unir duas ou mais consultas (SELECT).

Exemplos:

```
SELECT CHAPA, NOME, LOCALNASCIMENTO
FROM PFDEPEND
WHERE LOCALNASCIMENTO='BETIM'
UNION
SELECT CHAPA, NOME, LOCALNASCIMENTO
FROM PFDEPEND
WHERE LOCALNASCIMENTO='SALVADOR'
```

Lista o nome de todos os dependentes dos funcionários, que nasceram em Betim e todos os nascidos em Salvador.

```
SELECT CHAPA, NOME, LOCALNASCIMENTO
FROM PFDEPEND
WHERE LOCALNASCIMENTO='BETIM'
OR LOCALNASCIMENTO='SALVADOR'
```

Essa sentença também irá trazer o mesmo resultado, porém utilizando a opção: OR.

Outro exemplo da cláusula Union:

```
SELECT NOME, SALARIO
FROM PFUNC
WHERE SALARIO = ( SELECT MIN(SALARIO)FROM PFUNC)
UNION
SELECT NOME, SALARIO
FROM PFUNC
WHERE SALARIO = ( SELECT MAX(SALARIO)FROM PFUNC)
```

13. Alias de Coluna (Apelidos)

O Alias de coluna serve para nomear (apelidar) uma coluna em uma determinada consulta. É muito útil para utilização de nomes mais sugestivos, no momento da visualização da consulta.

Exemplos:

```
SELECT NOME, DATAADMISSAO ADMISSAO
FROM PFUNC
Na sentença acima apelidamos a coluna DATAADMISSAO.
```

```
SELECT MAX(SALARIO) AS MAIOR_SALARIO
FROM PFUNC
Na sentença acima apelidamos a coluna MAIOR_SALARIO.
```

```
SELECT MIN(SALARIO) 'MENOR SALARIO'
```

FROM PFUNC

Na sentença acima apelidamos a coluna MENOR SALARIO.

Exercício

Utilizando a ferramenta SQL Server Management Studio, realize a montagem das sentenças abaixo:

1. Selecione os nomes, em ordem alfabética, de todos os funcionários da filial 1, bem como o código da filial.
2. Selecione os nomes e salários (estes em ordem decrescente) dos funcionários, cujos salários não estejam entre 400,00 e 600,00 Reais, que foram admitidos em maio de 2000, que trabalham na filial 1 e que possuam a função 00018.
3. Selecione os nomes dos funcionários da Filial 1, cujo nome contenha a letra N.
4. Liste os nomes e os salários dos funcionários da coligada 1, cujo nome contenha a letra M e o salário seja superior a 600.
5. Selecione os nomes, as chapas, os salários e a data de admissão dos funcionários cujos salários não estejam entre 400,00 e 600,00 reais, que foram admitidos em maio de 2000 e que trabalham na filial 1.
6. Selecione a quantidade de funcionários agrupados por seção, que pertençam à coligada 1.
7. Mostre a média salarial, o maior e o menor salário de cada filial.
8. Mostre os códigos das seções e a média salarial das mesmas, mas somente das que possuem a média salarial maior que 1200.
9. Selecione o nome e data de admissão dos 10 funcionários mais antigos da empresa.
10. Informe a filial, o número de funcionários de cada filial, a soma dos salários, o maior salário e o menor salário, "apelidando" as colunas: Filial, Total, Maior e Menor.
11. Quantos funcionários existem em cada sindicato relacionado à empresa?
12. Selecione o código da seção, a quantidade de funcionários, sendo o código da situação igual a A (ATIVO) e, somente da seção com mais de dois funcionários.
13. Informe os nomes dos funcionários que trabalham na seção 01.01, que possuam função 00011 ou 00012 e que ganham salário maior que 600, em ordem decrescente.
14. Selecione o nome, data de admissão e o código da coligada dos funcionários, que foram admitidos em maio de 2000 e cujo o nome se inicie com a letra 'J'.
15. Liste o nome, salário e data de admissão de todos os funcionários com menos de 7 anos de empresa. Ordene o resultado pela data de admissão.

14. Consultas com Junção (duas ou mais tabelas – INNER JOIN, ligação)

A sintaxe é praticamente a mesma. A diferença está na utilização da cláusula WHERE, necessária para se fazer a ligação entre as tabelas. É imprescindível que as colunas de ligação sejam realmente idênticas (conteúdo). É na junção de tabelas, que a teoria dos bancos de dados relacionais, se apóia para se diferenciar dos não-relacionais.

A ligação entre tabelas é necessária sempre que for preciso selecionar um “valor” de uma coluna, de uma tabela “Pai” ou de uma tabela “filha” em relação a tabela principal de sua sentença SQL.


Exemplos:

```
SELECT PFUNC.NOME, PCODSITUACAO.DESCRICAO
FROM PFUNC INNER JOIN PCODSITUACAO
ON PFUNC.CODSITUACAO = PCODSITUACAO.CODCLIENTE
```

Neste exemplo, a tabela principal é a PFUNC (funcionários da empresa) e não temos a descrição da situação fazendo parte de sua estrutura. A descrição da situação está na tabela PCODSITUACAO (situações dos funcionários). Sendo assim, ligamos as colunas em que os conteúdos sejam idênticos (fazem parte das chaves primárias e estrangeiras das tabelas envolvidas). A tabela PCODSITUACAO é filha da tabela PFUNC, pois, a coluna CODCLIENTE é chave primária em PFUNC, e, conseqüentemente, é chave estrangeira em PCODSITUACAO.

```
SELECT *
FROM GLINKSREL
WHERE MASTERTABLE = 'PFUNC'
AND CHILDTABLE = 'PCODSITUACAO'
```

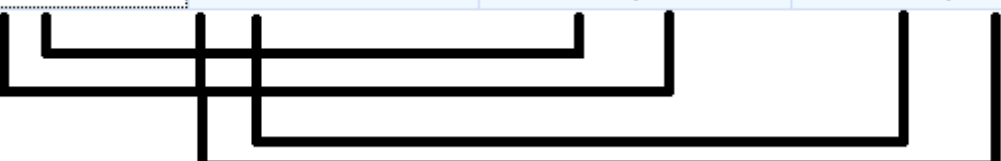
MASTERTABLE ▲ ▼	CHILDTABLE	MASTERFIELD	CHILDFIELD
PFUNC	PCODSITUACAO	CODSITUACAO	CODCLIENTE



```
SELECT PFUNC.NOME, PSECAO.DESCRICAO
FROM PFUNC INNER JOIN PSECAO
ON PFUNC.CODCOLIGADA = PSECAO.CODCOLIGADA
AND PFUNC.CODSECAO = PSECAO.CODIGO
```

```
SELECT *
FROM GLINKSREL
WHERE MASTERTABLE = 'PFUNC'
AND CHILDTABLE = 'PSECAO'
```

MASTERTABLE ▲ ▼	CHILDTABLE	MASTERFIELD	CHILDFIELD
PFUNC	PSECAO	CODCOLIGADA,CODSECAO	CODCOLIGADA,CODIGO



Neste exemplo, a tabela principal é a PFUNC (funcionários da empresa) e não temos a descrição da seção fazendo parte de sua estrutura. A descrição da seção está na tabela PSECAO (Seções da empresa). Sendo assim, ligamos as colunas em que os conteúdos sejam idênticos (fazem parte das chaves primárias e estrangeiras das tabelas envolvidas). A tabela PSECAO é filha da tabela PFUNC, pois, a coluna CODSECAO é chave primária em PSECAO, e, conseqüentemente, é chave estrangeira em PFUNC.

Exemplos:

```
SELECT GCOLIGADA.NOME, PFUNC.CHAPA, PFUNC.NOME
FROM GCOLIGADA, PFUNC
WHERE GCOLIGADA.CODCOLIGADA=PFUNC.CODCOLIGADA
```

Lista o nome da coligada e as chapas dos funcionários.

```
SELECT GBANCO.NOME, PFUNC.CHAPA, PFUNC.NOME
FROM GBANCO, PFUNC
WHERE GBANCO.NUMBANCO=PFUNC.CODBANCOPAGTO
```

Lista o nome do banco de pagamento e as chapas dos funcionários.

```
SELECT GBANCO.NOME, PFUNC.CHAPA, PFUNC.NOME
FROM GBANCO, PFUNC
WHERE GBANCO.NUMBANCO=PFUNC.CODBANCOFGTS
```

Lista o nome do banco de FGTS e as chapas dos funcionários.

```
SELECT PCODSITUACAO.DESCRICAO, PFUNC.CHAPA, PFUNC.NOME
FROM PCODSITUACAO, PFUNC
WHERE PCODSITUACAO.CODCLIENTE=PFUNC.CODSITUACAO
AND PCODSITUACAO.DESCRICAO LIKE 'AF%'
```

Lista o nome da situação (somente as que contém a descrição afastamento) e as chapas dos funcionários.

Outro exemplo usando a opção INNER JOIN:

```
SELECT PFUNC.NOME, PFUNCAO.NOME
FROM PFUNC INNER JOIN PFUNCAO
ON PFUNC.CODCOLIGADA= PFUNCAO.CODCOLIGADA
AND PFUNC.CODFUNCAO= PFUNCAO.CODIGO
```

Lista os nomes dos funcionários e os nomes das suas funções.

```
SELECT PFUNC.NOME, PFUNCAO.NOME,PSECAO.DESCRICAO
FROM PFUNC INNER JOIN PFUNCAO
ON PFUNC.CODCOLIGADA= PFUNCAO.CODCOLIGADA
AND PFUNC.CODFUNCAO= PFUNCAO.CODIGO
INNER JOIN PSECAO
ON PFUNC.CODCOLIGADA = PSECAO.CODCOLIGADA
```

```
AND PFUNC.CODSECAO = PSECAO.CODIGO
```

Lista os nomes dos funcionários, os nomes das funções dos funcionários e a descrição da seção dos funcionários.

```
SELECT PCARGO.NOME, PFUNC.CHAPA, PFUNC.NOME
FROM PCARGO, PFUNC, PFUNCAO
WHERE PCARGO.CODCOLIGADA=PFUNCAO.CODCOLIGADA
AND PCARGO.CODIGO=PFUNCAO.CARGO
AND PFUNCAO.CODCOLIGADA=PFUNC.CODCOLIGADA
AND PFUNCAO.CODIGO=PFUNC.CODFUNCAO
```

Lista o nome do cargo e as chapas dos funcionários. Observe que não existe ligação direta da tabela pcargo com pfunc. Foi necessário utilizarmos a tabela pfuncao para compor o resultado.

```
SELECT PCARGO.NOME, AVG(PFUNC.SALARIO)
FROM PCARGO INNER JOIN PFUNCAO
ON PCARGO.CODCOLIGADA = PFUNCAO.CODCOLIGADA
AND PCARGO.CODIGO = PFUNCAO.CARGO
INNER JOIN PFUNC
ON PFUNC.CODCOLIGADA = PFUNCAO.CODCOLIGADA
AND PFUNC.CODFUNCAO = PFUNCAO.CODIGO
GROUP BY PCARGO.NOME
HAVING AVG(PFUNC.SALARIO) BETWEEN 500 AND 1500
ORDER BY AVG(PFUNC.SALARIO) DESC
```

Lista os cargos que possuem salários com média salarial entre 500 e 1500 reais, e mostra as médias dos mesmos, ordenando-os em ordem decrescente.

```
SELECT PFUNC.NOME, PFUNC.DATAADMISSAO, PCODSITUACAO.DESCRICAO
FROM PFUNC INNER JOIN PCODSITUACAO
ON PFUNC.CODSITUACAO = PCODSITUACAO.CODCLIENTE
WHERE PCODSITUACAO.DESCRICAO LIKE 'ATI%'
AND PFUNC.DATAADMISSAO >= '05/05/2008'
ORDER BY PFUNC.DATAADMISSAO
```

Lista os nome, data de admissão dos funcionários, e situação dos funcionários, retornando os que estão situação ATIVA, data de admissão maior ou igual a 05/05/2008, ordenando-as em ordem crescente.

15. Alias de Tabela (Apelido)

O Alias de tabela é utilizado para nomear (apelidar) uma tabela em uma determinada consulta.

Exemplos:

```
SELECT G.NOME, P.NOME
FROM PFUNC P, GFILIAL G
WHERE P.CODCOLIGADA = G.CODCOLIGADA
AND P.CODFILIAL = G.CODFILIAL
```

16. SubQuery

Em uma cláusula SELECT é possível chamar outra cláusula SELECT, envolvendo as mesmas tabelas da cláusula inicial ou não. A estas cláusulas internas damos o nome de SubQuery, ou subsentenças.

Exemplo:

Listar os funcionários que possuem o salário maior que a média de salários da Filial 2.

```
SELECT NOME, SALARIO
FROM PFUNC
WHERE SALARIO > (SELECT AVG(SALARIO) FROM PFUNC WHERE CODFILIAL=2)
```

Listar as filiais que o maior salário seja superior a média dos salários dos funcionários das Filiais.

```
SELECT CODFILIAL, SUM(PFUNC.SALARIO), COUNT(PFUNC.CHAPA), AVG(PFUNC.SALARIO)
FROM PFUNC
GROUP BY CODFILIAL
HAVING MAX(PFUNC.SALARIO) > (SELECT AVG(PFUNC.SALARIO)
                             FROM PFUNC)
```

Este recurso é muito utilizado em sentenças mais elaboradas que envolvam tabelas sem relacionamento, ou correção de dados envolvendo mais de uma tabela, dentre outros.

Exercício

Utilizando a ferramenta SQL Server Management Studio, realize a montagem das sentenças abaixo:

1. Selecione o nome dos funcionários que possuem o salário, superior a todos os funcionários da seção 'COM - Depto Comercial'.
2. Encontre todos os funcionários que possuem o cargo 'Gerência' e, que ganhem mais que os funcionários da filial 'RM Sistemas - Filial RJ'.
3. Mostre o código de seção, a soma do salários, o maior salário, o número de funcionários e a média salarial, onde o maior salário seja superior a média dos salários dos funcionários.

17. OUTER JOINS

As consultas feitas anteriormente são realizados com base em INNER JOINS, essas consultas desprezam, ou melhor, não listam as linhas que não tenham um dado correspondente na outra tabela, assim se formos relacionar a tabela de bancos e funcionários, só virão os funcionários que tenham pelo menos um banco informado e os nomes dos bancos que tenham pelo menos um funcionário associado.

As consultas com OUTER JOINS produzem resultados contendo as colunas selecionadas de cada linha em uma tabela, mesmo que esta não tenha relação com um subconjunto de linhas de uma outra tabela. A sintaxe para uma consulta com OUTER JOIN é:

```
SELECT <lista_colunas>
FROM <tabela_esquerda> { LEFT | RIGHT | FULL } [OUTER] JOIN <tabela_direita>
[ ON <condições>]
[ WHERE <condições>]
```

Assim, ao utilizarmos a cláusula OUTER JOIN no exemplo acima, ela listará os bancos e os funcionários independente deles terem relação entre si. A cláusula LEFT, RIGHT e FULL determinará qual tabela será listada na íntegra, ou se ambas no caso do FULL.

Utilizamos consultas com OUTER JOINS quando queremos comparar todos os dados de uma tabela, com o subconjunto dos dados relacionados na outra tabela.

É necessário especificar o tipo de junção (JOIN) a ser executado. Existem três possibilidades:

- Um **LEFT OUTER JOIN** ou **LEFT JOIN** recupera todas as linhas da tabela à esquerda da junção e, recupera aquelas linhas da tabela à direita, que satisfazem às condições especificadas na cláusula ON.

```
SELECT PFUNC.NOME, GBANCO.NOME BANCO
FROM PFUNC LEFT OUTER JOIN GBANCO
ON PFUNC.CODBANCOPAGTO = GBANCO.NUMBANCO
```

NOME	BANCO
PEDRO PAULO AUTONOMO	NULL
MARIANA LORENA AUTONOMA	NULL
LUANA PRO-LABORE	NULL
MESTRE DOS MAGOS PRO-LABORE	NULL
WEMERSON HENRIQUE MARTINS ANTUNES	BANCO DO BRASIL
SOPHIA MARIANA LARA	BANCO DO BRASIL
BILLY JEAN	BANCO DO BRASIL

Lista todos os nomes dos funcionários e os nomes dos bancos de depósito de pagamento, somente para aqueles que possuem contas de pagamento cadastradas.

- Um **RIGHT OUTER JOIN** ou **RIGHT JOIN** recupera todas as linhas da tabela à direita da junção e, recupera aquelas linhas da tabela à esquerda, que satisfazem às condições especificadas na cláusula ON.

```
SELECT PFUNC.NOME, GBANCO.NOME BANCO
FROM PFUNC RIGHT JOIN GBANCO
ON PFUNC.CODBANCOPAGTO=GBANCO.NUMBANCO
```

NOME	BANCO
NULL	AJUSTE CONVERSAO
NULL	AJUSTE CONVERSAO
NULL	AJUSTE CONVERSAO
NULL	AJUSTE CONVERSAO
JOAO DA SILVA JORDAM	BANCO DO BRASIL
MARIA CONCEICAO DE ASSIS	BANCO DO BRASIL

Lista todos os nomes dos bancos de depósito de pagamento e os nomes dos funcionários, somente para aqueles que possuem funcionários cadastrados.

- Um **FULL OUTER JOIN** ou **FULL JOIN** recupera todas as linhas de ambas as tabelas, independentemente, das condições específica das na cláusula ON.

```
SELECT PFUNC.NOME, GBANCO.NOME BANCO
FROM PFUNC FULL OUTER JOIN GBANCO
ON PFUNC.CODBANCOPAGTO=GBANCO.NUMBANCO
```

NOME	BANCO
PEDRO PAULO AUTONOMO	NULL
MARIANA LORENA AUTONOMA	NULL
LUANA PRO-LABORE	NULL
MESTRE DOS MAGOS PRO-LABORE	NULL
NULL	AJUSTE CONVERSAO
NULL	AJUSTE CONVERSAO
NULL	AJUSTE CONVERSAO
NULL	AJUSTE CONVERSAO

Lista todos os nomes dos funcionários e os nomes dos bancos de depósito de pagamento, independentemente, das condições específica das na cláusula ON.

Nos SGBDs do ORACLE e SQL SERVER, temos outras maneiras de especificar um OUTER JOIN na cláusula WHERE, que poderá utilizar o (+) e o (*), respectivamente. Vejamos como ficariam o exemplo 1 desta nova maneira:

Em Oracle:

```
SELECT P.NOME, B.NOME
FROM PFUNC, GBANCO
WHERE P.CODBANCOPAGTO (+)= B.NUMBANCO
```

Em SQL Server

```
SELECT P.NOME, B.NOME
FROM PFUNC P, GBANCO B
WHERE P.CODBANCOPAGTO *= B.NUMBANCO
```


Exercício

Utilizando a ferramenta SQL Server Management Studio, realize a montagem das sentenças abaixo:

1. Listar os nomes, as chapas e os nomes dos bancos de FGTS de todos funcionários da coligada 1, mesmo para aqueles que não tenham a conta de FGTS cadastrada.
2. Listar os nomes dos funcionários e os nomes de todos os sindicatos, independente deles possuírem funcionários filiados ou não.
3. Liste os nomes, as chapas, os salários e a descrição da situação dos funcionários. Mesmo que exista uma situação que não esteja associada a um funcionário.
4. Liste os nomes de todas as funções, e naquelas que possuírem funcionários, mostre o total de salário gasto com cada uma, em ordem salarial do maior para o menor.

18. Ferramenta Visões de Dados (Consulta SQL)

Embora seja destinada somente às consultas, esta importante ferramenta é mais um grande recurso disponível com exclusividade nos aplicativos RM, que permitirá a você cadastrar e executar sentenças de Consulta SQL, acessando diretamente o Banco de Dados de seu aplicativo.

Uma Consulta SQL é uma sentença (montada ou escrita) em linguagem própria para a realização de uma pesquisa (consulta) diretamente no banco de dados, buscando informações (dados) que estejam devidamente gravados (populados) nas diversas tabelas (cadastros) relativos ao sistema.

18.1. Editor Consulta SQL

O editor permite a edição/alteração de uma consulta SQL. Para isto selecione os dados necessários nas pastas desta janela:

Consultas SQL: RELATORIO - LISTA DE FUNCIONARIOS - 03.9.0507

1001/1001 Anexos Processos

Identificação

Código: RELATORIO Título: LISTA DE FUNCIONARIOS Aplicação: RM LABORE

☐ Visível a todas Coligadas ☐ Não considerar segurança por linha ☐ Não considerar segurança por coluna

Executar Configurar Editor

Formatar Dicionário

Editar **Opções**

1 SELECT * FROM PFUNC

Banco de dados externo: Versão: Disponível para

☒ Execuções em Visões ☒ Relatórios ☐ RM Portal

☒ Filtros ☐ Menu

Data da Última Alteração: 05/11/2015 15:11 Usuário Responsável: mestre

OK Cancelar Salvar

Editor de Consultas SQL

Tabelas / Campos Filtros / Condições Agrupar / Ordenar

Tabelas Disponíveis:

Tabela	Descrição
AABONFUN	Abonos Lançados para os Funcionários
AABONFUNAM	Abonos Períodos Anteriores
AABONFUTURO	Tabela de Abonos Futuros
AABONO	Abonos
AABONOCALCULADO	Abono Calculado

Adiciona Remove

Tabelas Selecionadas:

Tabela
PFUNC

Campos Disponíveis:

Tabela	Nome	Descrição
PFUNC	*	Todos os Campos
PFUNC	AJUDACUSTO	Valor da Ajuda de Custo
PFUNC	ANOCOMPTRA...	Ano da Competência du...
PFUNC	ANTIGACARTT...	Antiga Carteira de Traba...
PFUNC	ANTIGACHAPA	Antiga Chapa do Funcio...

Adiciona

Campos Selecionados:

*

OK Cancelar Ajuda

Tabelas / Campos – Sentença

Nesta pasta serão informadas (selecionadas) as tabelas das quais os dados serão apurados para a consulta. Das tabelas selecionadas serão definidos os campos que se deseja consultar no banco de dados.

Corresponde ao comando SELECT de uma sentença SQL.

Deverão ser selecionadas as tabelas desejadas para a montagem e edição da sentença a ser executada.

Tabelas Disponíveis

Selecione a tabela entre as tabelas disponíveis, observando seu nome e sua respectiva descrição no banco de dados. Clique sobre o botão Adiciona.

Após pressionar este botão, a tabela selecionada será transferida automaticamente para o espaço abaixo Tabelas Selecionadas, para a montagem da sentença SQL. Ou, dê um duplo clique no nome da tabela desejada.

Tabelas Selecionadas

Informa as tabelas que serão utilizadas na consulta SQL.

Campos Disponíveis

Uma vez selecionada(s) a(s) tabela(s), estarão também disponíveis os campos a serem pesquisados.

Selecione o campo entre os campos disponíveis, observando seu nome e sua respectiva descrição no banco de dados. Clique sobre o botão Adiciona.

Após pressionar este botão, o campo selecionado será transferido automaticamente para o espaço abaixo Campos Selecionados, para a montagem da sentença SQL. Ou, dê um duplo clique no nome do campo desejado.

Observação

Para selecionar mais de um campo, mantenha a tecla CTRL pressionada ao selecionar os campos.

Campos Selecionados

Este espaço apresentará todos os campos já selecionados para a sentença. Da mesma forma como ocorre com o espaço de Tabelas Selecionadas, poderá ser utilizado para que você digite cada campo desejado para a sentença.

Observação

Para a inclusão de campos na área de Campos Selecionados, você poderá ou digitar cada campo desejado, ou utilizar os mesmos passos informados para a seleção e transferência de tabelas.

Filtros / Condições – Sentença

Nesta pasta serão informados os critérios de seleção dos dados a consultar. Os filtros poderão classificar previamente, datas, códigos, descrições ou outros dados que você desejar, conforme critérios definidos nos filtros.

Corresponde aos comandos WHERE [CAMPO] =...; WHERE [CAMPO] >= ...; WHERE [CAMPO] > ..., etc, de uma sentença SQL.

Opção para inclusão de campos-filtro para a seleção de dados da tabela.

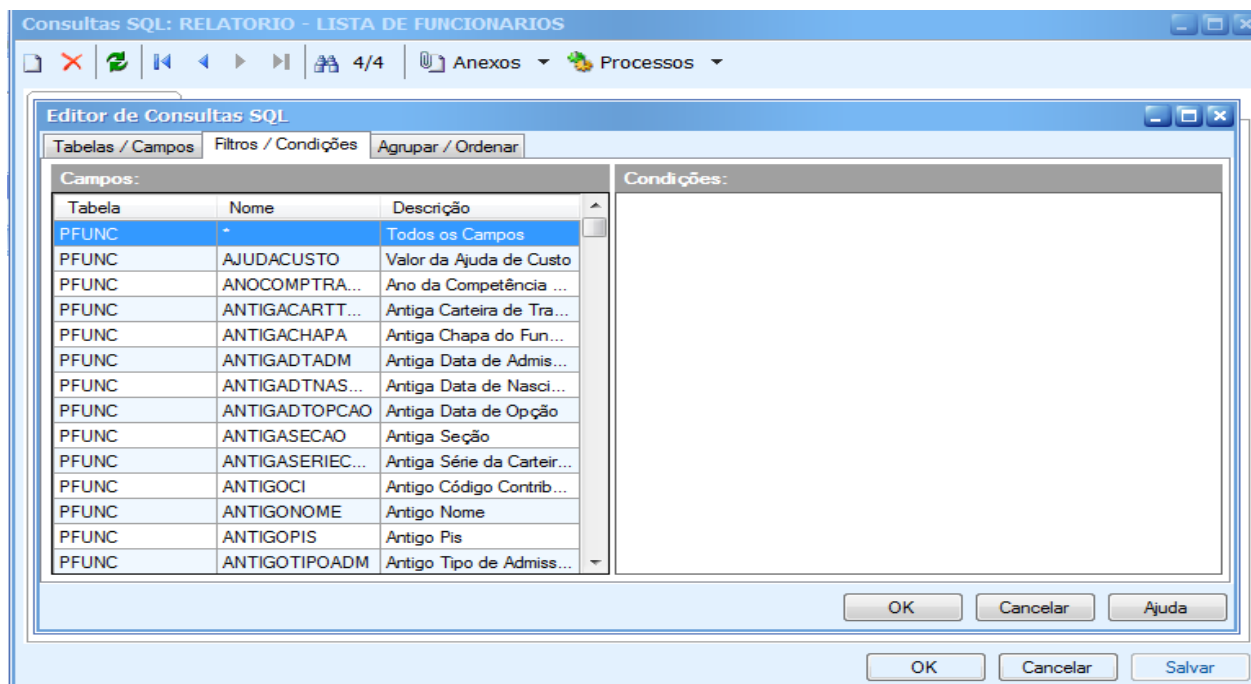
Campos

Verifique neste espaço todos os campos da tabela disponíveis para serem utilizados como filtros de dados. Selecione o campo entre os campos disponíveis, observando seu nome e sua respectiva função no banco de dados.

Selecione cada campo desejado, clicando duas vezes sobre ele. O campo selecionado será transferido automaticamente para o espaço à direita denominado Condições, para a montagem da sentença SQL. Os campos deverão ser selecionados e transferidos já na sequência correta de montagem da sentença.

Condições

Este espaço receberá os campos selecionados na coluna à esquerda para a montagem do filtro. Entretanto, você poderá, se desejar, digitar o filtro independentemente de se ter selecionado previamente os campos desejados.



Agrupar / Ordenar – Sentença

Nesta seção, poderão ser definidas as formas de agrupamento de dados como, por exemplo, agrupar por data, por código, etc.

Além do agrupamento, da mesma forma, os dados resultantes do processamento da sentença poderão ser ordenados por data, código, ordem alfabética ou por outro critério que você definir.

Corresponde aos comandos ORDER BY ou GROUP BY de uma sentença SQL.

Recurso para o agrupamento e ordenação dos dados resultantes da execução da sentença.

Campos

Selecione os campos disponíveis nesta coluna para serem agrupados/ordenados.

Para selecionar e transferir um ou mais campos, proceda da seguinte forma:

Selecione o campo entre os campos disponíveis, observando seu nome e sua respectiva função no banco de dados.

Clique sobre o ícone que representa uma seta única para a direita.

Após pressionar este ícone, o campo selecionado será transferido automaticamente para o espaço abaixo Agrupar por ou no espaço abaixo Ordenar por, dependendo de qual seta for pressionada.

Observação

Para selecionar mais de um campo, mantenha a tecla CTRL pressionada ao selecionar os campos.

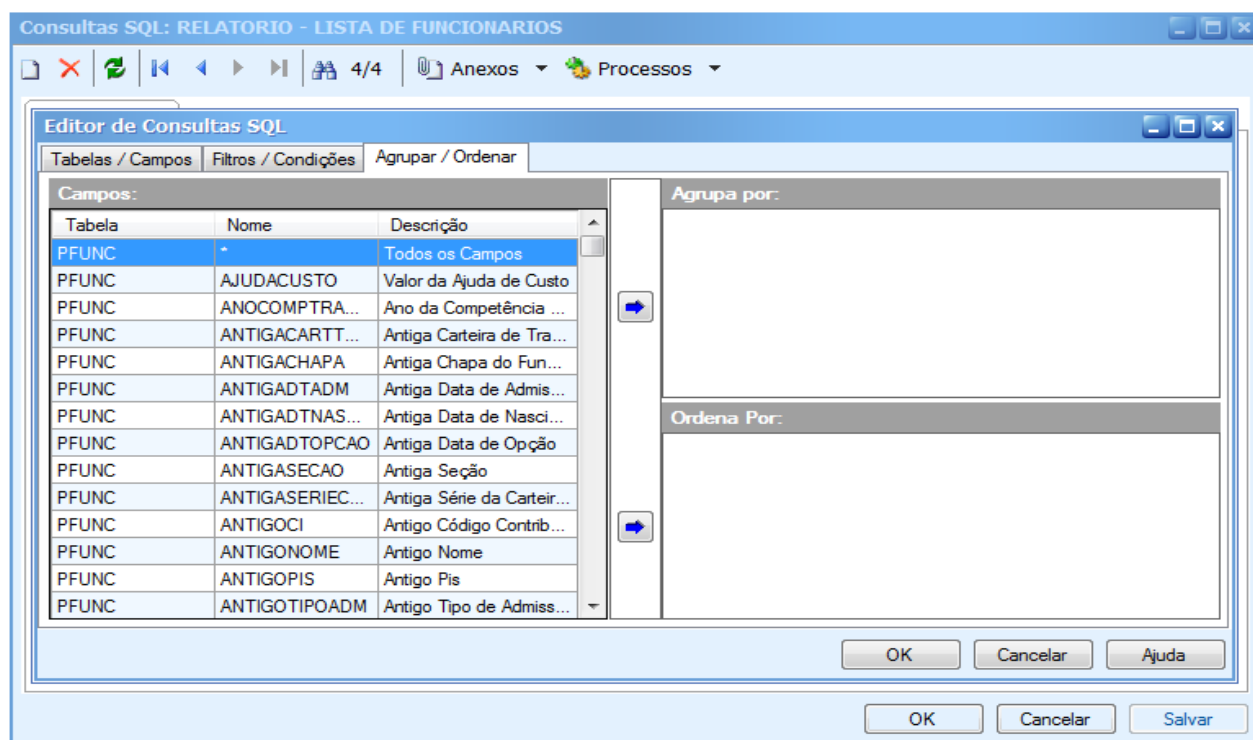
Agrupar por

Espaço que apresenta os campos selecionados para agrupamento dos dados da sentença. Poderá ser também utilizado para que você digite cada campo ao invés de selecionar e transferir através das setas de transferência.

O agrupamento consiste na junção de dados para serem apresentados em grupos, em bloco.

Ordenar por

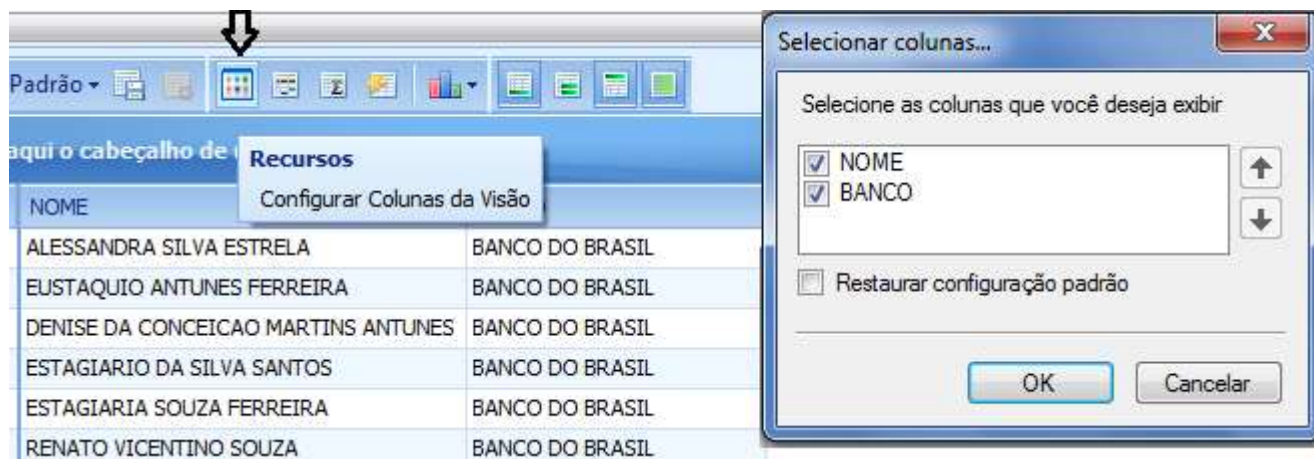
Apresentam os campos de ordenação selecionados. Poderá ser utilizado também para a digitação dos campos de agrupamento desejados, ao invés de se proceder a seleção e transferência através das setas de transferência.



18.2. Recursos

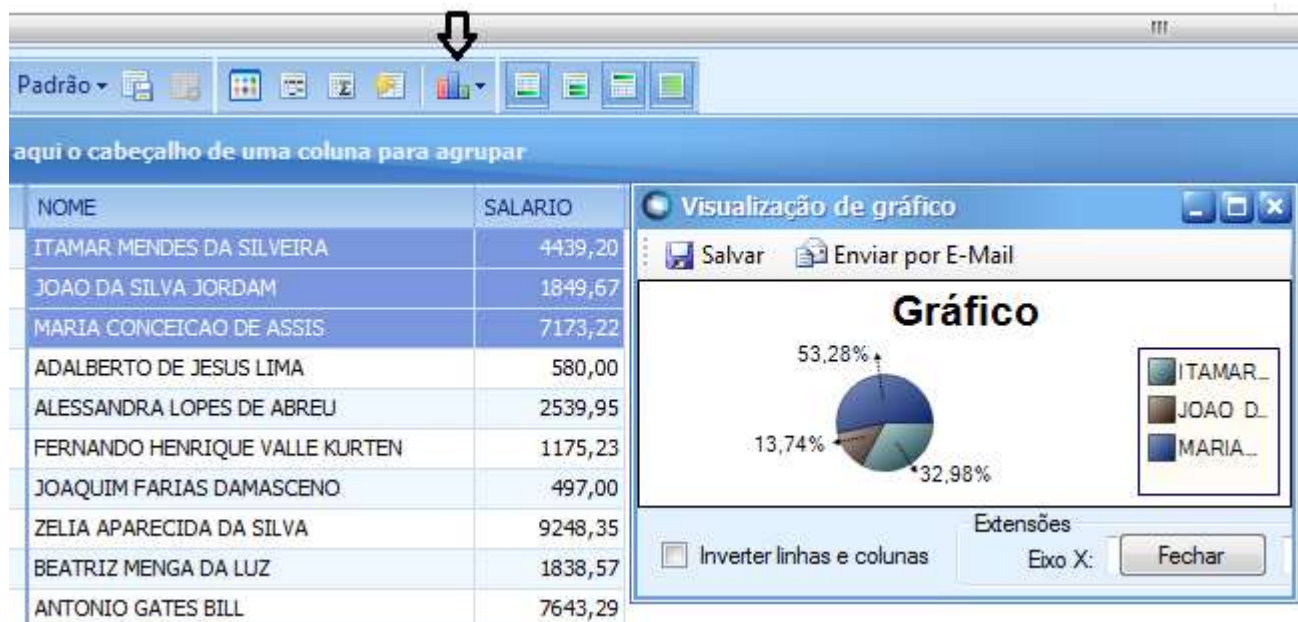
Configurar Colunas da Visão

Esse recurso permite selecionar quais campos devem ser exibidos no resultado e qual a ordem de exibição



Gerar Gráfico Instantâneo da Visão

Esta opção permite gerar gráficos através do resultado da consulta SQL executada.

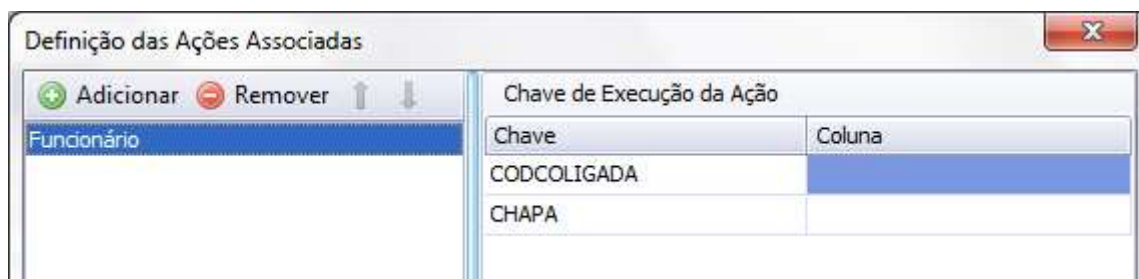


Definição de Actions Associada a visão

Essa opção permite que seja destacada cada célula da coluna e que se possa executar uma Action a partir da grid de execução da sentença SQL.

Para que uma Action seja executada, deve ser feita uma comparação entre o(s) campo(s) chave(s) da tabela com o(s) campo(s) resultantes da Consulta SQL.

Na grid de action deve ser inserida a Action que se deseja abrir. A partir daí visualizaremos a grid Campos da Chave Primária, onde serão exibidos os campos chaves da Action escolhida. Na coluna chamada de "coluna", devemos escolher o campo resultante da consulta SQL que será comparado com o da tabela.



The screenshot shows a dialog box titled 'Definição das Ações Associadas'. It has a toolbar with '+ Adicionar' (Add), '- Remover' (Remove), and up/down arrows. Below the toolbar is a list box containing 'Funcionário'. To the right is a table titled 'Chave de Execução da Ação' (Action Execution Key) with two columns: 'Chave' (Key) and 'Coluna' (Column). The table has two rows: 'CODCOLIGADA' and 'CHAPA'.

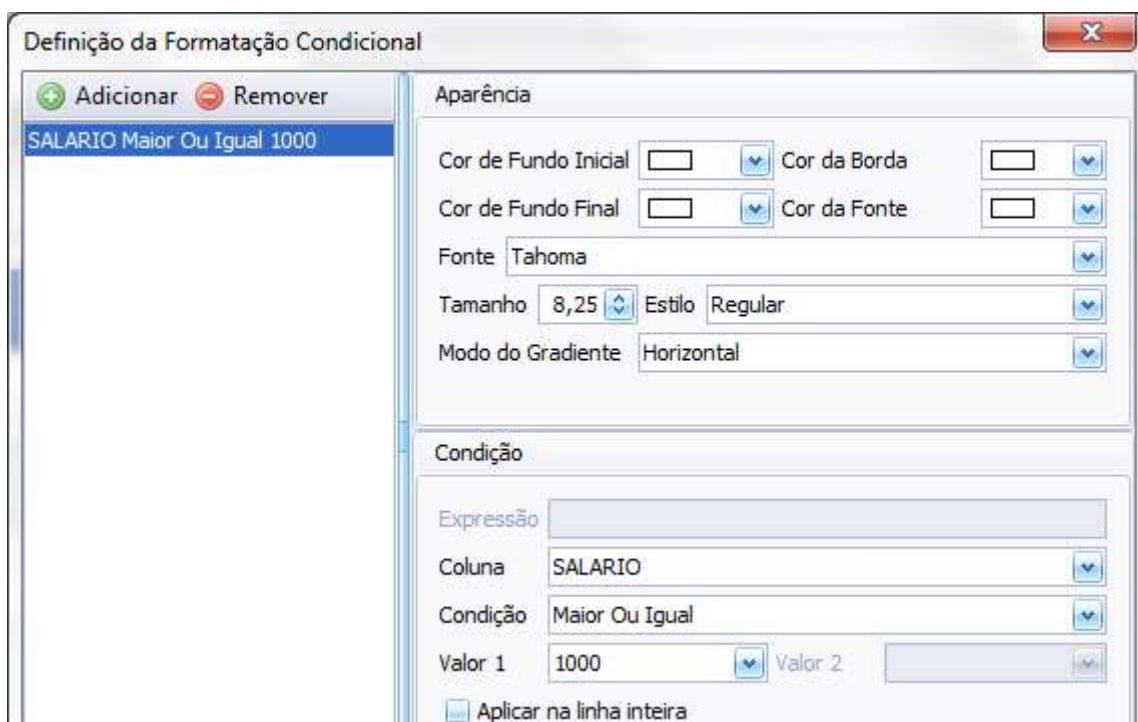
Chave	Coluna
CODCOLIGADA	
CHAPA	

Definições de Formatação Condicional da Visão

Esse recurso permite colorir as células da grid de execução da sentença SQL baseado em uma série de comparações realizadas pelo usuário.

Para realizar essa operação, selecione um capo na grid de campos e acesse a formatação condicional.

No campo valor deverá ser inserido um valor para fazer a comparação. Caso a comparação seja verdadeira, a célula será preenchida com as cores que devem ser escolhidas nos campos Fundo e Fonte dessa mesma Grid.



A janela "Definição da Formatação Condicional" possui uma aba "Aparência" e uma aba "Condição". Na aba "Aparência", há campos para "Cor de Fundo Inicial", "Cor de Fundo Final", "Cor da Borda", "Cor da Fonte", "Fonte" (Tahoma), "Tamanho" (8,25), "Estilo" (Regular) e "Modo do Gradiente" (Horizontal). Na aba "Condição", há campos para "Expressão", "Coluna" (SALARIO), "Condição" (Maior Ou Igual), "Valor 1" (1000) e "Valor 2". Há também uma opção "Aplicar na linha inteira".

Definição das Colunas Calculadas da Visão

Esse recurso permite criar colunas na visão. Os dados dessa coluna será o resultado de uma expressão criada pelo usuário dentro da ferramenta.

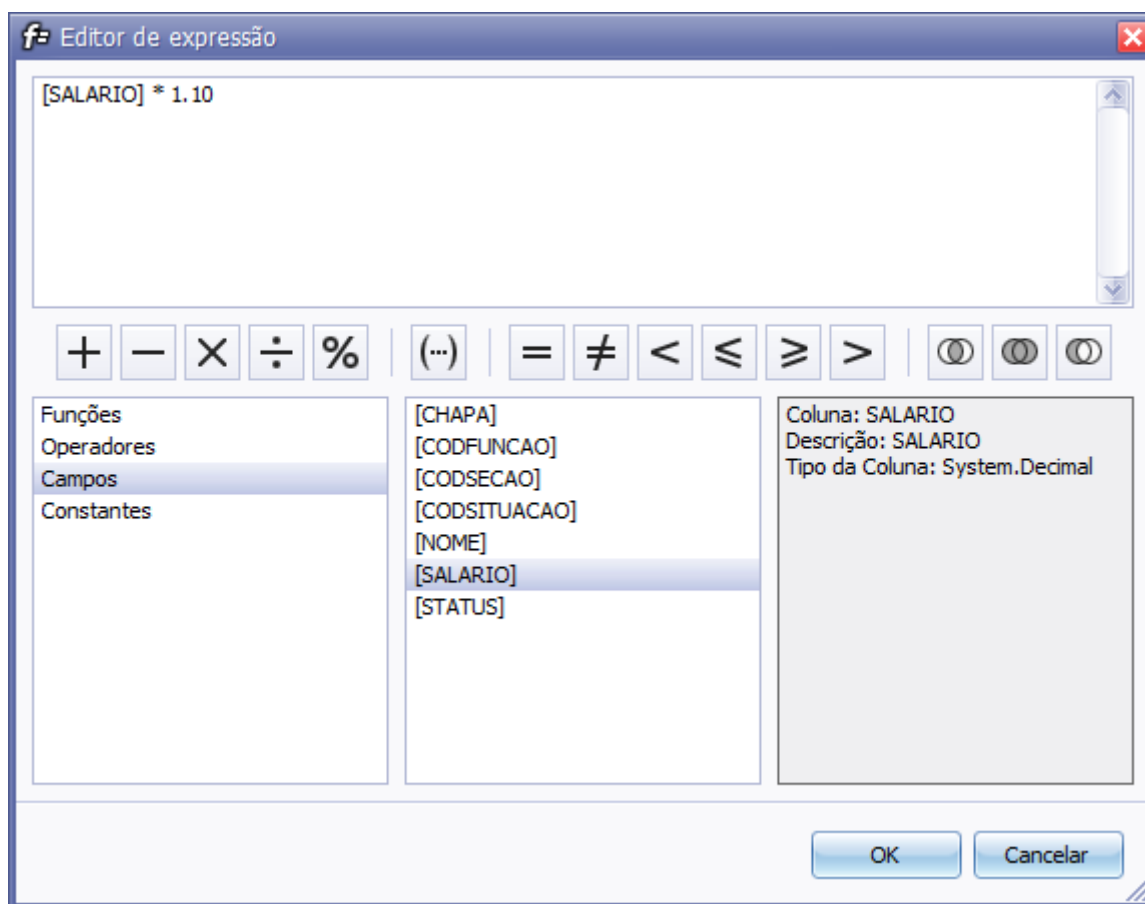
Na Grid Definição das Colunas Calculadas devemos clicar no botão Adicionar.



A janela "Definição das Colunas Calculadas" possui uma aba "Adicionar" e uma aba "Remover". Há uma tabela com as seguintes colunas: "Nome", "Título", "Tipo de Dado" e "...".

Para criar uma nova coluna na visão precisamos preencher os dados do cadastro acima.

Após informar o nome, título e tipo de dado, é necessário clicar em “expressão”, no final do cadastro, representado pelo símbolo “...”.



O editor de expressão se abrirá e o usuário poderá criar a expressão que definirá o dado que aparecerá na nova coluna.

No exemplo acima, o resultado da expressão será o valor do salário do funcionário vezes o índice 1.10.

Passagem de Parâmetros

Permite ao usuário informar o valor de uma condição, no momento da execução da consulta, sem que se altere a consulta.

Exemplo:

```
SELECT CHAPA, NOME, DATAADMISSAO  
FROM PFUNC  
WHERE CHAPA =: CHAPA
```

O parâmetro poderá ser utilizado nos relatórios, juntamente com a inserção da opção Parâmetros ou Fórmula.

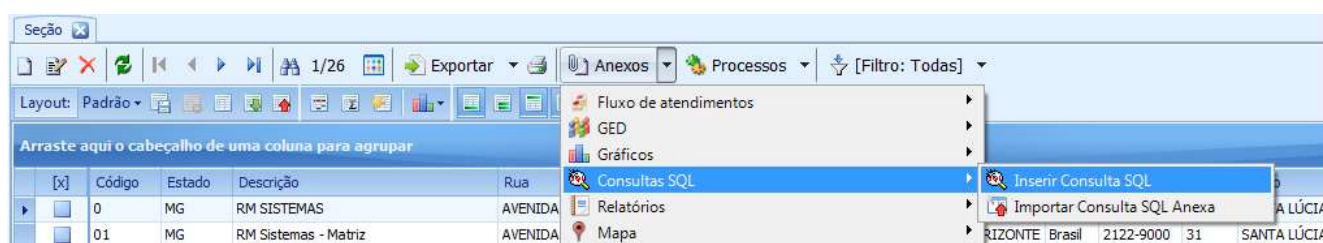
Importante/Saiba Mais

- Este recurso somente será possível através do aplicativo

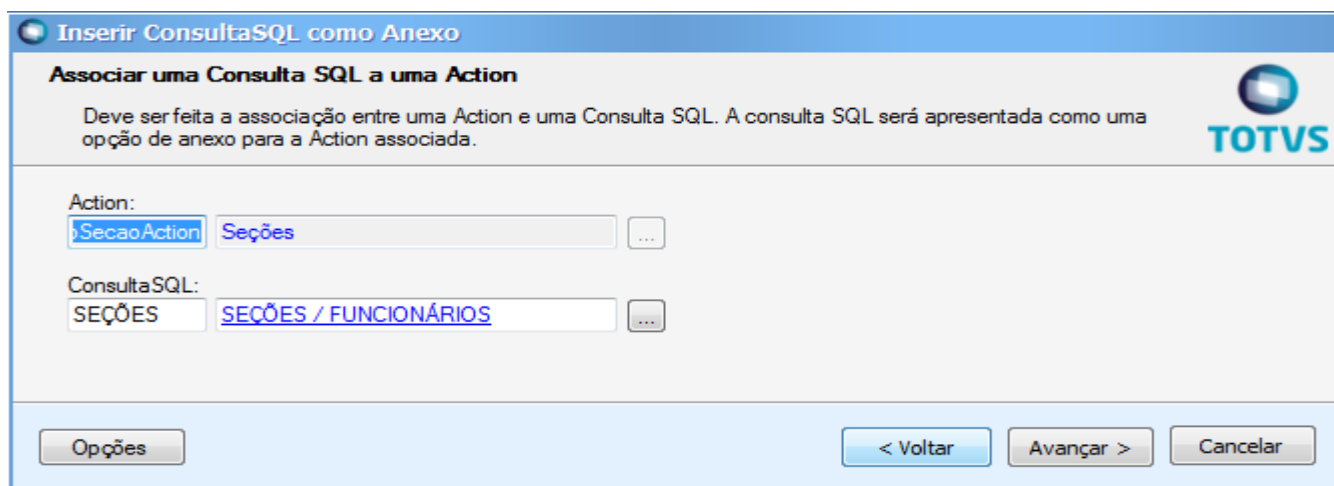
18.3. Inclusão de consultas SQL em anexos

```
SELECT CHAPA,NOME,CODSITUACAO SITUAÇÃO,SALARIO
FROM PFUNC (NOLOCK)
WHERE CODSECAO =:CODSECAO
AND CODCOLIGADA =:CODCOLIGADA
AND CODSITUACAO <> 'D'
```

Utilizaremos o cadastro de Seções dos funcionários para retornar os dados em anexo da consulta. Ao acessar o cadastro da Seção clique em: Anexos à Consultas SQL à Inserir consultas SQL



Selecione a consulta SQL que será utilizada no anexo.



Na próxima tela, insira o valor do parâmetro correspondente a consulta. Esta consulta SQL utiliza os parâmetros "SECAO" e "COLIGADA" deve-se neste caso informar o campo chave da tabela para este parâmetro que é o campo CODSECAO e CODCOLIGADA.

Consultas SQL e TOTVS Inteligência de Negócios



Inserir ConsultaSQL como Anexo

Definição de Parâmetros

Associe para cada parâmetro da consulta, uma coluna de dados da Visão.

Nome	Valor	VALORFIXO
CODSECAO	[CODIGO]	
CODCOLIGADA	[CODCOLIGADA]	

Opções < Voltar Avançar > Cancelar

Nesta tela é possível definir quais permissões de usuários para geração da consulta SQL. No nosso caso utilizaremos todos, porém é possível determinar qual usuário / perfil terá acesso.

Inserir ConsultaSQL como Anexo

Definição de Grupo de Visualização

Você deve selecionar qual o grupo de usuário visualizará esta Consulta SQL como Anexo.

Grupo de usuário

☐ Todos ☐ Perfis/Usuários

Opções < Voltar Avançar > Cancelar

Executando a consulta em anexo.

Ao clicar em: Anexos à Consultas SQL, será apresentada a consulta parametrizada.

Para execução clique sobre a consulta.

Seção

Layout: Padrão

Arraste aqui o cabeçalho de uma coluna para agrupar

[x]	Código	Estado	Descrição	Rua
	01.01	MG	DST - Depto Suporte Técnico	AVENID

Anexos Processos [Filtro: Todas]

- Fluxo de atendimentos
- GED
- Gráficos
- Consultas SQL
- Relatórios
- Mapa

SEÇÕES / FUNCIONÁRIOS

- Inserir Consulta SQL
- Importar Consulta SQL Anexa

Será executada a consulta SQL retornando os dados referentes a cada registro (Seção)

A cada seleção de Seção observe que a consulta SQL será recalculada de acordo com cada contexto existente.

SEÇÕES / FUNCIONÁRIOS						
Q-CHAPA	NOME	SITUAÇÃO	SALARIO			
00014	PEDRO DE ALCANTARA	A	1039,57			
00015	LEOPOLDO DOS SANTOS DE OLIVEIRA NETO	A	3629,05			
00049	VITOR PLAVIO CARDOSO	A	1238,92			
00095	LIANA CARLA LOES	P	1322,29			
00086	ROSANA DOS REIS SILVA	A	3900,00			

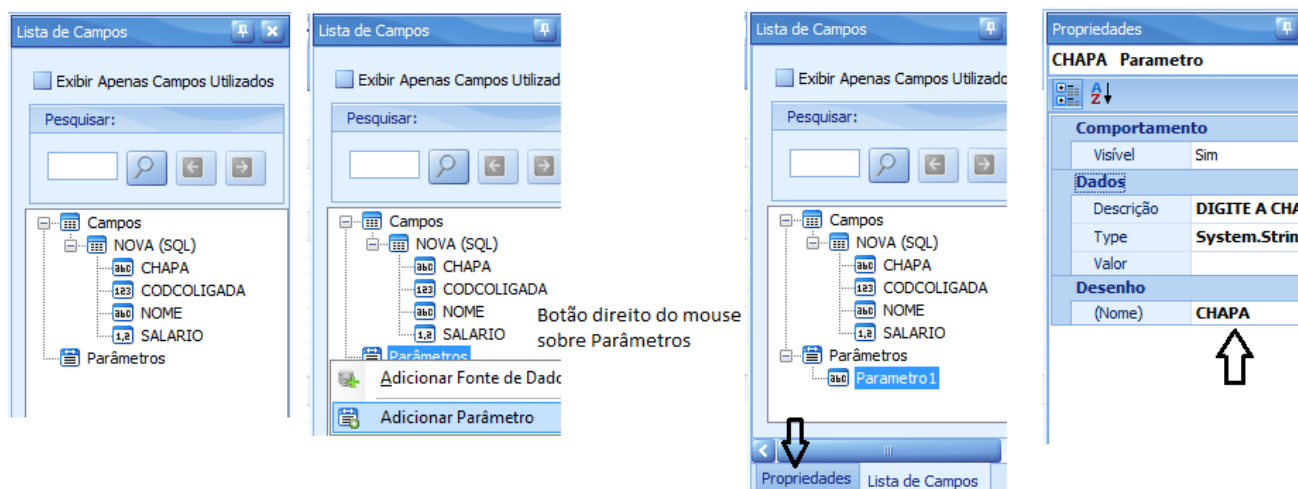
18.4. Exemplos de Aplicação das Sentenças

Gerador

O gerador de relatórios irá acessar diretamente o banco de dados, lendo e armazenando dados nas tabelas do sistema, conforme o relatório exigir.



É possível trabalhar com sentenças SQL no Gerador de Relatórios, unindo também a opção de passagem de parâmetro.

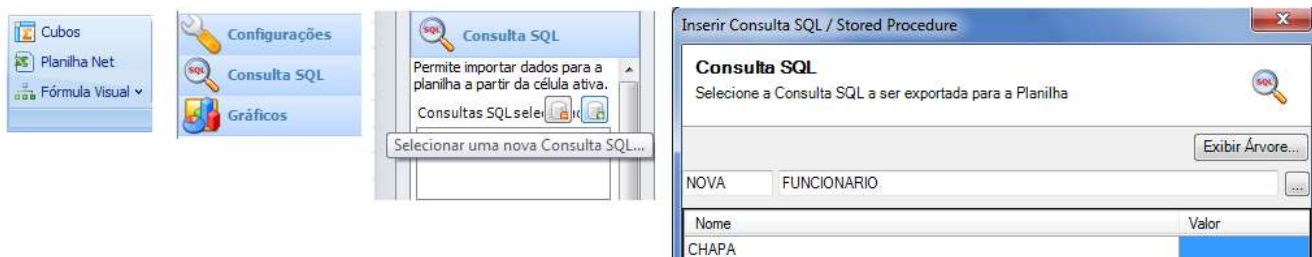


Planilha

Planilha é um arquivo que contém dados do usuário. Pode conter várias pastas, para que o usuário possa organizar vários tipos de relatórios num único arquivo.

A planilha do RM possui uma estrutura bastante semelhante à estrutura das planilhas do MS Excel.

Pode transformar-se em ferramenta de grande utilidade, conforme necessidade. Possui ampla aplicabilidade em cálculos, listagens e controles diversos que poderão estar relacionados ou não com suas tarefas diárias no sistema.



Exercício

Utilizando a ferramenta Consultas SQL, realize a montagem das sentenças abaixo:

Sentenças do TOTVS Gestão Financeira.

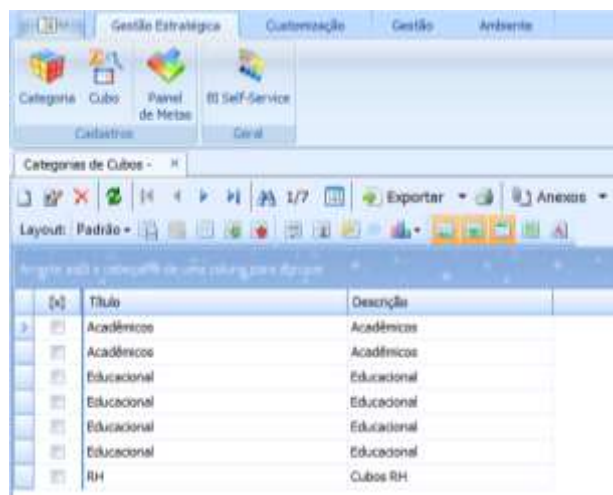
1. Criar uma sentença que retorne o saldo instantâneo de uma conta caixa, que deverá ser fornecida pelo usuário no momento da execução da sentença. A sentença ainda deverá listar os campos: código da conta caixa, descrição e saldo instantâneo.
2. Selecionar o código do cliente, o nome fantasia, a data de vencimento do lançamento e valor original. Use parâmetros para solicitar a digitação do ano e mês da data de vencimento a ser listada.
3. Retornar o nome do cliente ou fornecedor, estado e telefone. Utilize o recurso de parâmetros para solicitar a digitação do estado a ser listado.

Sentenças do TOTVS Gestão de Estoque, Compras e Faturamento.

1. Criar uma sentença que retorne o código da coligada, código da filial, nome do fornecedor, número do movimento, data de emissão e valor bruto do movimento. Utilize parâmetros para solicitar a digitação do número do movimento.
2. Selecionar os clientes que não possuem movimentação, a partir de uma data fornecida pelo usuário. Utilize parâmetros para solicitar a digitação das datas. Os campos a serem listados são: Código do cliente, nome fantasia do cliente, rua, número, complemento, bairro, cidade, cep e telefone.
3. Selecione os itens do movimento, a partir de uma data fornecida pelo usuário, apresentando os seguintes dados: código do produto, nome fantasia do produto, quantidade, unidade, preço unitário e data de emissão do movimento. Utilize o recurso de parâmetros, para solicitar a digitação da data de emissão dos itens do movimento.

19. TOTVS Inteligência de Negócios

19.1. Categoria

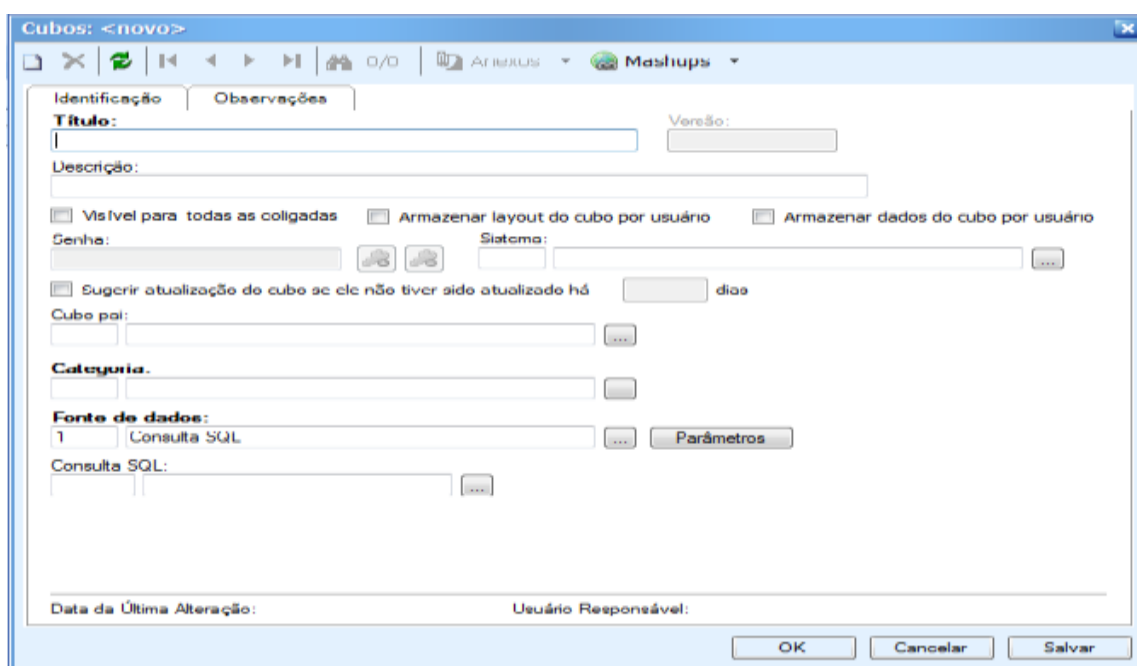


É preciso criar categorias para os cubos para organizá-los de acordo com os perfis de usuário. Isso significa que apenas alguns perfis de usuário terão permissão aos cubos. Para tal, um usuário precisa estar associado àquele perfil para visualizar os cubos.

O acesso do cadastro é através do módulo Inteligência de Negócios / Aba Gestão Estratégica / Categoria.

19.2. Cubo

Nesta janela você poderá fazer a manutenção de cubos em .NET. É possível criar cubos nesta visão e sua configuração deve ser feita pelo Assistente de Atualização dos cubos.



Cubos: <novo>

Identificação Observações

Título: _____ **Versão:** _____

Descrição: _____

☐ Visível para todas as coligadas ☐ Armazenar layout do cubo por usuário ☐ Armazenar dados do cubo por usuário

Senha: _____ **Sistema:** _____

☐ Sugerir atualização do cubo se ele não tiver sido atualizado há _____ dias

Cubo pai: _____

Categoria: _____

Fonte de dados: 1 Consulta SQL _____ **Parâmetros**

Consulta SQL: _____

Data da Última Alteração: _____ **Usuário Responsável:** _____

OK Cancelar Salvar

Veja algumas configurações:

Armazenar layout do cubo por usuário.

Permite salvar um layout diferente do cubo para cada usuário.

Armazenar dados do cubo por usuário

Permite salvar os dados do cubo de forma diferenciada para cada usuário.

Senha

Permite definir uma senha ao cubo. Clique na chavezinha com o “+” para incluir uma senha

Sistema

Os cubos poderão ser agrupados por sistema. Os cubos só serão exibidos neste sistema.

Cubo Pai

Serve para ajustar os cubos em forma de árvore, como o Gerador de relatórios.

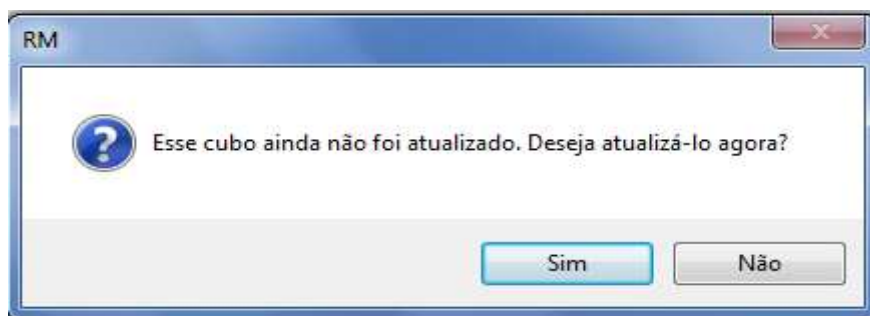
Categoria

Categoria do cubo com os perfis que terão permissão a ele.

Consulta SQL

Indique qual a consulta SQL utilizada para a confecção do cubo.

Clique duas vezes sobre o Cubo .NET criado anteriormente, clique em sim para que o cubo seja atualizado



19.3. Roteiro para criação de cubos

- 1) Deve-se criar uma categoria de cubos acessando o módulo Inteligência de Negócios / Aba Gestão Estratégica / Categoria.
- 2) Associar a categoria a um perfil de usuário e dar as permissões necessárias para a formatação do cubo.
- 3) Acessar o módulo Inteligência de Negócios / Aba Gestão Estratégica / Cubo. Inclua um novo Cubo, associe a uma categoria de cubo.
- 4) Associe uma consulta SQL ao cubo. No nosso exemplo utilizaremos uma sentença simples, que deve estar cadastrada no sistema:

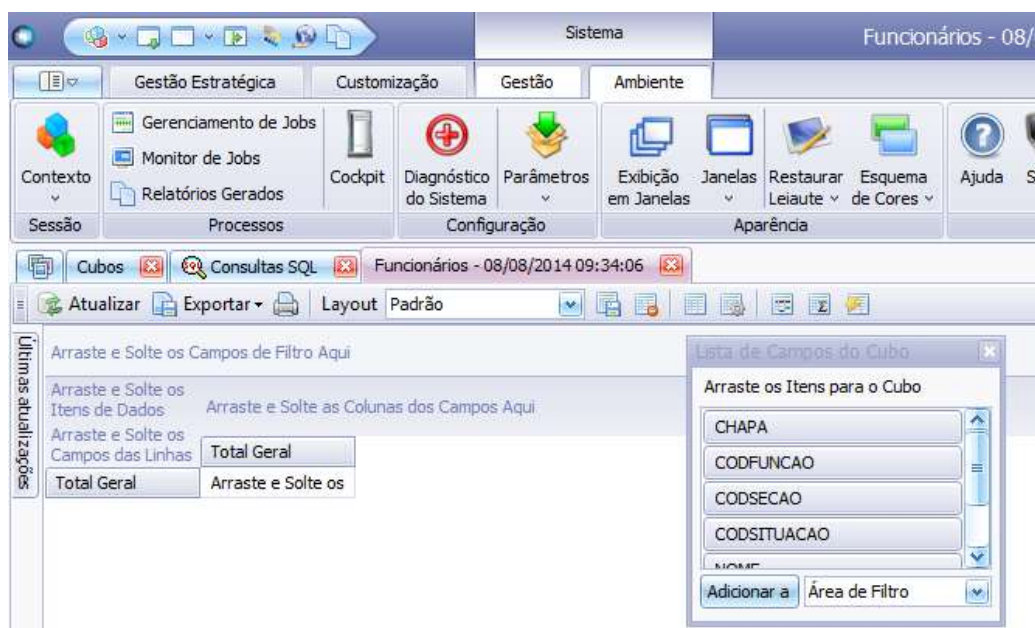
```
SELECT CHAPA, NOME, CODSECAO, CODFUNCAO, CODSITUACAO, SALARIO
```


FROM PFUNC

Depois se selecionada a consulta clique em OK.

5) Clique duas vezes no cubo criado para abrir sua edição. O cubo será mostrado com a lista de campos disponíveis para a criação do cubo. Esta lista é baseada na consulta SQL associada ao cubo.

6) Será necessário então determinar onde os campos serão visualizados no cubo.



Realize as seguintes ações levando em consideração o cenário acima:

- Arraste para a área Itens de Dados os seguintes campos: SALARIO
- Arraste para a área Campos das Linhas os seguintes campos: CODSECAO, NOME
- Arraste para a área Colunas dos Campos os seguintes campos: CODSITUACAO
- Arraste para a área Campos de Filtro os seguintes campos: CHAPA, CODFUNCAO

7) O próximo passo é formatar os campos do Cubo. Clique no botão “Formatar campos”.

Nesta etapa, configure o texto a ser apresentado em cada dimensão (campo), e no caso das dimensões das áreas de dados, configure a máscara dos valores a serem apresentados.

Texto

Informe uma descrição para o campo selecionado.

Totalizar Valores

Marque esta opção para que esse campo seja totalizado.

Máscara

Selecione uma máscara para o campo selecionado. Só estará disponível para “Campos de Dados”, isto é, para campos que foram organizados na área de dados.

Será possível selecionar os seguintes tipos de máscaras:

Inteiro (#.##0): Visualiza o valor sem casas decimais e com separador de milhar.

Decimal (#.##0,00): Visualiza o valor com N casas decimais e com separador de milhar.

Monetário (R\$ #.##0,00): Visualiza o valor com a unidade monetária corrente, com N casas decimais e separador de milhar;

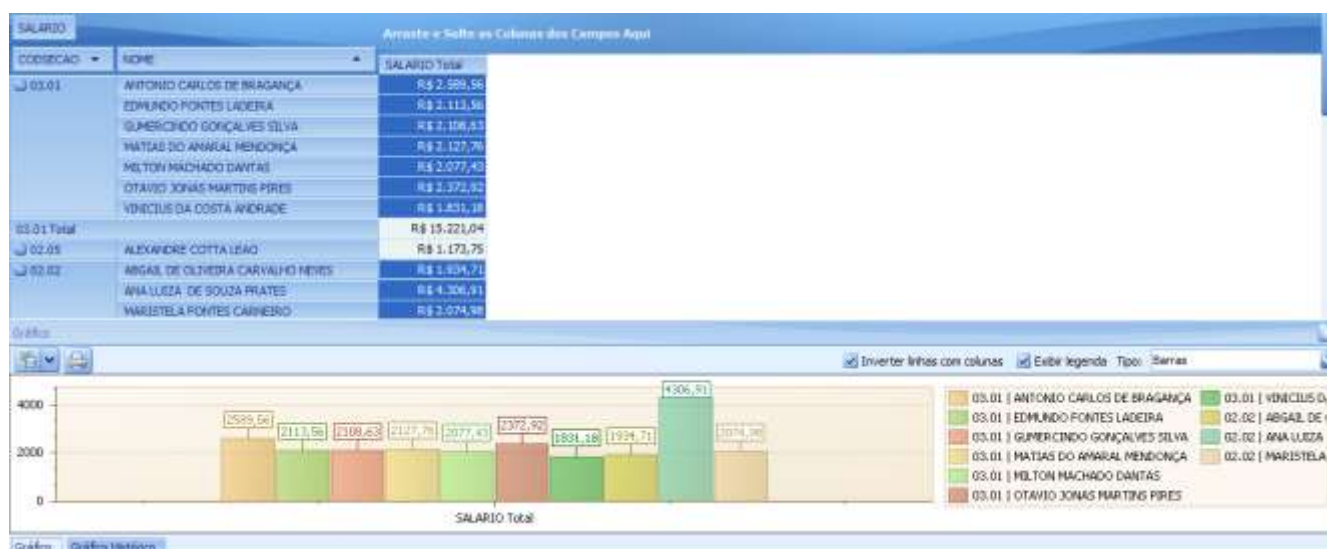
Percentual (0.00%): Visualiza o percentual correspondente ao valor total.

Importante/Saiba Mais

A determinação do número de casas decimais só terá efeito sobre campos da base de dados que suportarem a mesma quantidade de casas decimais escolhida. Com isso se o usuário escolher uma quantidade de casas decimais acima da quantidade suportada pelo campo, esta será preenchida com o algarismo zero.

19.4. Criação de gráfico

Será possível gerar o gráfico selecionando os campos correspondentes e após clicar em Gráfico.



Exercício

Utilizando a ferramenta TOTVS Inteligência de Negócios, crie os seguintes cubos abaixo:

1. Criar uma consulta SQL que retorne os seguintes dados:

Tabela FLAN (Lançamentos Financeiros)

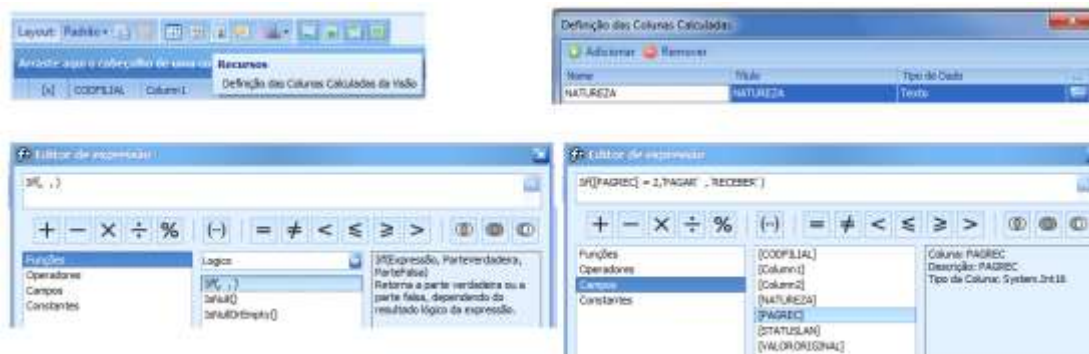
Campos: Filial, Ano da data de vencimento, Mês da data de vencimento, Valor Original, Pagar ou Receber e Status do Lançamento

Obs: crie um filtro que retorne apenas os lançamentos não vencidos.

Baseado na SQL criada, monte um cubo que retorne os lançamentos por Ano, Mês e Filial, conforme a figura abaixo.

VALOR ORIGINAL	COOPFILIAL	NATUREZA								
ANO	MES	1	PAGAR	RECEBER	1 Total	2	PAGAR	RECEBER	2 Total	Total Geral
2015	1		R\$ 2.000,00	R\$ 1.500,00	R\$ 3.500,00		R\$ 3.000,00		R\$ 3.000,00	R\$ 6.500,00
	2		R\$ 1.800,00	R\$ 2.250,00	R\$ 4.050,00		R\$ 2.900,00		R\$ 2.900,00	R\$ 6.950,00
	3		R\$ 2.300,00	R\$ 1.950,00	R\$ 4.250,00		R\$ 1.500,00	R\$ 2.000,00	R\$ 3.500,00	R\$ 7.750,00
	11		R\$ 2.600,00	R\$ 4.000,00	R\$ 6.600,00		R\$ 3.450,00	R\$ 4.500,00	R\$ 7.950,00	R\$ 14.550,00
2015 Total			R\$ 8.900,00	R\$ 9.700,00	R\$ 18.600,00		R\$ 7.850,00	R\$ 9.500,00	R\$ 17.350,00	R\$ 35.950,00

As tabelas PAGAR e RECEBER de acordo com Recursos Definições das Colunas Calculadas da Visão



2. Crie um cubo que permita a visualização de todos os lançamentos financeiros vencidos e em aberto. Os resultados devem ser retornados por cliente/fornecedor.
3. Crie um gráfico de uma coluna de cada exemplo acima e salve em formato PNG (Imagem).

Exercício

Utilizando a ferramenta TOTVS Inteligência de Negócios

O cubo será criado com base na consulta SQL abaixo:

```
SELECT GCOLIGADA.NOMEFANTASIA COLIGADA, GFILIAL.NOMEFANTASIA FILIAL,
PSECAO.DESCRICAO SEÇÃO, PFUNC.NOME FUNCIONÁRIOS,
PFUNCAO.NOME FUNÇÃO, PCARGO.NOME CARGO, PSINDIC.NOME SINDICATO,
PCODSITUACAO.DESCRICAO SITUAÇÃO, GBANCO.NOME BANCO, PFUNC.SALARIO
FROM GCOLIGADA, GFILIAL, PSECAO, PFUNC, PFUNCAO, PCARGO, PSINDIC,
PCODSITUACAO,GBANCO
WHERE GCOLIGADA.CODCOLIGADA = GFILIAL.CODCOLIGADA
AND GCOLIGADA.CODCOLIGADA = PFUNC.CODCOLIGADA
AND GFILIAL.CODFILIAL = PFUNC.CODFILIAL
AND GFILIAL.CODCOLIGADA = PFUNC.CODCOLIGADA
AND PFUNC.CODSECAO = PSECAO.CODIGO
AND PFUNC.CODCOLIGADA = PSECAO.CODCOLIGADA
AND PFUNC.CODSINDICATO = PSINDIC.CODIGO
AND PFUNC.CODCOLIGADA = PSINDIC.CODCOLIGADA
AND PFUNCAO.CODCOLIGADA = PFUNC.CODCOLIGADA
AND PFUNCAO.CODIGO = PFUNC.CODFUNCAO
AND PFUNCAO.CODCOLIGADA = PCARGO.CODCOLIGADA
AND PFUNCAO.CARGO = PCARGO.CODIGO
AND PCODSITUACAO.CODCLIENTE = PFUNC.CODSITUACAO
AND GBANCO.NUMBANCO = PFUNC.CODBANCOPAGTO
```

Baseado na SQL acima, monte o cubo que retorne os dados da figura abaixo:

COLIGADA	FILIAL	FUNÇÃO	SITUAÇÃO									
SALARIO												
SEÇÃO	FUNCIONÁRIOS	CARGO	SINDICATO	BANCO	AFIL Trab...	AF Previst...	Ativo	Desativado	Despesa Oc...	Licença M...	Total Usar	
ACD - Depto de...	ANTONIO SERA JUNIOR	Professor	SINDICADOS	CADIA ECONOMICA FEDERAL			R\$ 3.000,00				R\$ 3.000,00	
	XUAREZ DA SILVA PAVANELLO	Professor	SINDICADOS	BANCO DO BRASIL			R\$ 3.000,00				R\$ 3.000,00	
	ROSANA DOS REIS SILVA	Professor	SINDICADOS	BANCO HSBC				R\$ 2.500,00			R\$ 2.500,00	
	SUELI MARIA DOS SANTOS FERR...	Professor	SINDICADOS	CADIA ECONOMICA FEDERAL			R\$ 1.800,00				R\$ 1.800,00	
ACD - Depto de Administração Total												
ACD - Depto de...	CRISTIANE MARINER DE SOUSA	Professor	SINDICADOS	BANCO DO BRASIL			R\$ 2.500,00				R\$ 2.500,00	
	PATRICIA LUIZA RAHOES	Professor	SINDICADOS	BANCO DO BRASIL			R\$ 2.500,00				R\$ 2.500,00	
ACD - Depto de Sistemas de Informação Total												
							R\$ 5.000,00				R\$ 5.000,00	

Retorne os salários maiores ou iguais a R\$ 3000,00 na cor da fonte vermelha.

20. Painel de Metas

O Painel de Metas é um recurso, com base em Gráficos Medidores, onde se pode definir metas a serem alcançadas. É importante citar que o gráfico medidor é gerado com base em uma consulta SQL que retorna apenas um valor. A definição das metas é feita através de formulas. Essas fórmulas podem ser desde uma expressão simples até uma expressão complexa.

Na edição do painel de metas temos:

Nome

Entre com o nome que irá identificar a meta.

Descrição

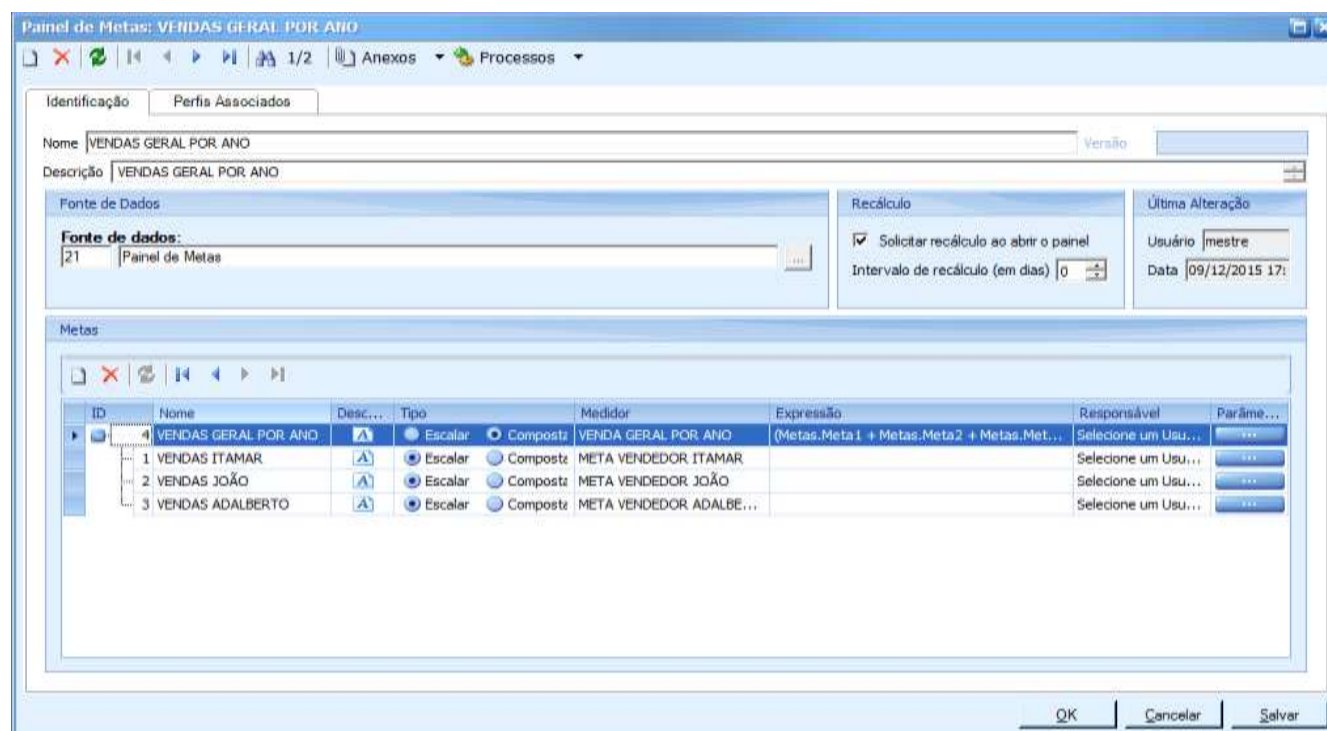
Entre com a descrição da meta.

Recálculo

Marque a opção solicitar recálculo ao abrir o painel e defina qual a periodicidade em dias que o recálculo será realizado. Para formular as metas é necessário incluir os registros na Grid. A grid possui a funcionalidade de hierarquizar os registros. Isso permite que um registro seja filho direto de outro registro.

Para fazer a hierarquia de registros, clique no seletor de linha do registro que deseja que ele se torne filho e arraste para o registro que será pai do outro. Dessa forma o registro pai irá se tornar do tipo composta, habilitando assim a coluna expressão. Essa coluna Expressão para uma meta que seja Composta é de preenchimento obrigatório.

Na coluna expressão existe um ícone com reticências, clicando nele será exibido nosso editor de expressão. Esse editor se encontrar em outras partes do sistema. Nele pode-se incluir inúmeros cálculos matemáticos para definir uma meta.



ID	Nome	Desc...	Tipo	Medidor	Expressão	Responsável	Parâmetros
	VENDAS GERAL POR ANO		Escalar	VENDA GERAL POR ANO	(Metas.Meta1 + Metas.Meta2 + Metas.Met...	Selecione um Usu...	...
1	VENDAS ITAMAR		Escalar	META VENDEDOR ITAMAR		Selecione um Usu...	...
2	VENDAS JOÃO		Escalar	META VENDEDOR JOÃO		Selecione um Usu...	...
3	VENDAS ADALBERTO		Escalar	META VENDEDOR ADALBE...		Selecione um Usu...	...

Visualização

A visualização do painel de metas será feita através de duplo-clique no registro desejado na visão de painéis de metas. A visualização é feita de forma hierárquica nos moldes da edição do painel.

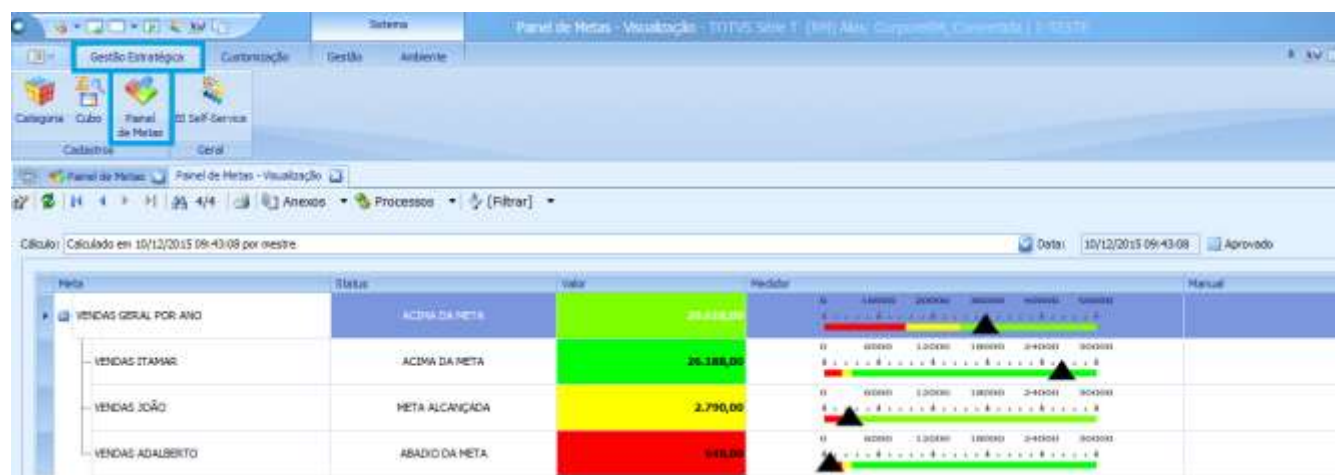
São exibidas as metas, seus status e valores, e um gráfico (medidor) indicando a posição da meta em relação à sua escala de forma visual. A cor de fundo do valor da meta será a cor da escala onde a meta se encontra.

Nessa tela os valores das metas poderão ser alterados manualmente, de forma a simular um cenário. Assim que o valor for alterado será solicitado o recálculo do painel de metas, de forma que as metas compostas sejam recalculadas utilizando o valor digitado.

Para isso dê um duplo clique na meta listada, será exibida uma tela para inserção de valores. Por default, vem preenchido com o valor atual. Após fazer a alteração, clique em ok para confirmar e acesse o item de Processo e o item Calcular Painel de Metas.

Nessa mesma edição da meta em sua visualização é possível definir que a meta esteja aprovada, marcando o check aprovado. Com o check aprovado, este cálculo não pode mais ser excluído.

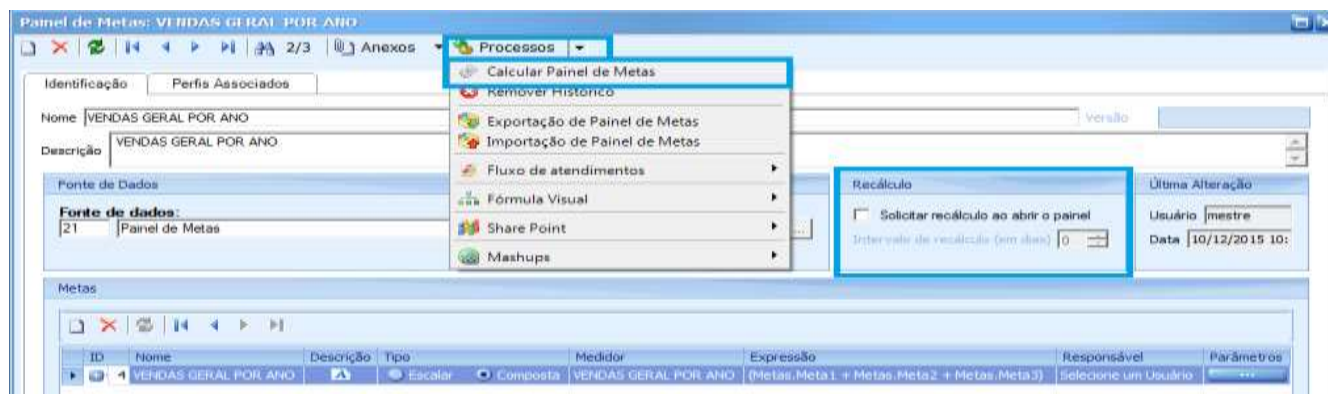
Utilize o campo “Observação” para colocar detalhes genéricos sobre a meta. Além disso, use o “campo plano de ação” para definir ação caso as metas sejam atingidas ou não. Este campo será de suma importância, já que dele sairão as tarefas a serem executadas.



Atenção

Esse cálculo irá substituir o cálculo gerado anteriormente, já que ocorreu a execução de cálculo de meta além da alteração dos valores manualmente.

Além disso, cuidado ao excluir um painel, porque se você o fizer, todos seus recálculos serão excluídos.



Exercício

Primeiramente, precisaremos de uma consulta SQL, a qual somente modificará o Código do Funcionário/Vendedor. O Painel de Metas é um 'agrupador' de gráficos de Gauge (medidores). Neste caso, cada 'medidor' será um Funcionário. Por isso a necessidade de criar uma consulta SQL para cada Funcionário. Assim, pode ser utilizada a consulta abaixo:

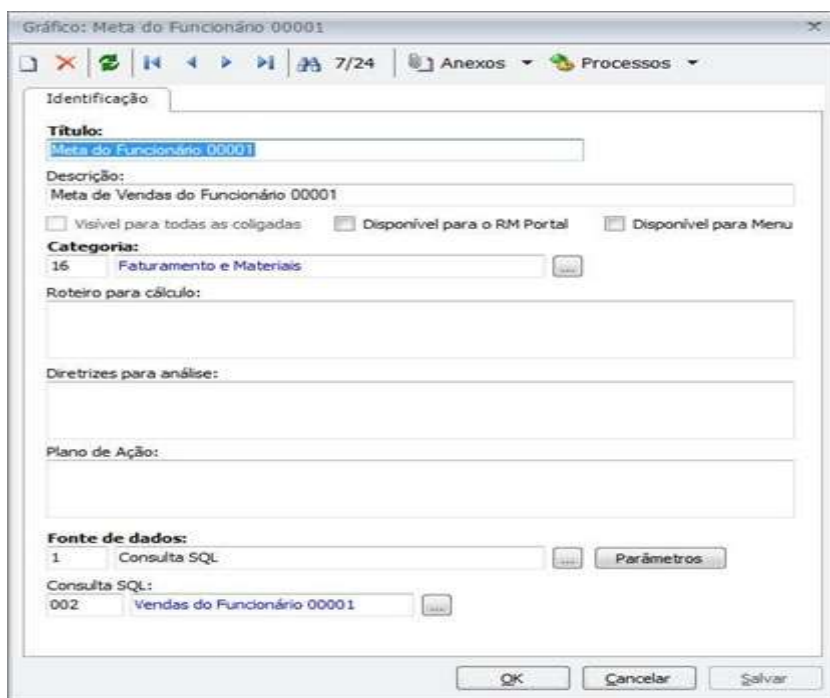
```
SELECT TMOV.CODVEN1, TVEN.NOME, SUM(TMOV.VALORBRUTO) AS
VALORVENDAS
FROM TMOV,TVEN (NOLOCK)
WHERE TMOV.CODVEN1 IS NOT NULL AND TMOV.CODCOLIGADA = 1
AND TMOV.CODCOLIGADA = TVEN.CODCOLIGADA
AND TMOV.CODVEN1 = TVEN.CODVEN
/* Selecione Movimentos de Vendas */
AND TMOV.CODTMV LIKE '2.2.____' AND TMOV.STATUS <> 'C'
/* Determine aqui o período do gráfico */
AND TMOV.DATAMOVIMENTO BETWEEN '01/01/2011' AND '12/31/2012'
/* Determine aqui o Vendedor */
AND TMOV.CODVEN1 = 'XXXXX'
GROUP BY TMOV.CODVEN1, TVEN.NOME
```

Vendedores: 00001 ; 00002 ; 00004

Criação dos Gráficos 'Medidores'

Após criar todas as consultas, temos que criar gráficos de Gauge para cada Funcionário, pois o Painel de Metas vai exibi-los juntos, para demonstrar o desempenho global da empresa. Assim, o gráfico deverá seguir o seguinte passo-a-passo:

Cadastro do Gráfico:



Cadastro do Gráfico:



Configuração do Gráfico:

Assistente de Configuração de Gráfico

Medidor

Defina propriedades relativas ao medidor

Selecione a coluna cujo valor será exibido:

VALORVENDAS

Escala
 Valor inicial: 0
 Valor final: 30000

Frequência
 Labels: 3000
 Marcador maior: 3000
 Marcador menor: 1500

Faixas de valor

Título	Início	Fim	Cor	
ABAIXO DA META	0	2000	Red	Configurar Ação
META ALCANÇADA	2000	3000	Yellow	Configurar Ação
ACIMA DA META	3000	30000	Green	Configurar Ação

☐ Pré-visualizar

Neste momento, podem ser definidas metas individuais para cada Funcionário/Vendedor. Por exemplo: o Funcionário/Vendedor que atende ao mercado estrangeiro possui uma meta mais elevada que o Funcionário/Vendedor que comercializa para um determinado bairro, por exemplo.

Depois de criados todos os medidores de cada Funcionário/Vendedor, basta criar um gráfico específico para o Painel de Metas, conforme abaixo:

Cadastro do Gráfico:

Gráfico: VENDA GERAL POR ANO

2/385 Anexos Processos

Identificação

Título: VENDAS GERAL POR ANO **Versão:**

Descrição: VENDAS GERAL POR ANO

☐ Visível para todas as coligadas ☐ Disponível para o RM Portal ☐ Disponível para Menu

Categoria: 1 Treinamento

Tipo de Gráfico:
☒ Padrão
☐ Drill Down

Roteiro para cálculo:

Diretrizes para análise:

Plano de Ação:

Fonte de dados: 4 PAINEL DE METAS **Parâmetros**

OK Cancelar Salvar

Assistente de Configuração de Gráfico

Medidor

Defina propriedades relativas ao medidor

Selecione a coluna cujo valor será exibido:
 Resultado da Meta

Escala
 Valor inicial: 0
 Valor final: 50000

Frequência
 Labels: 5000
 Marcador maior: 5000
 Marcador menor: 2500

Faixas de valor

Título	Início	Fim	Cor	Ação
ABAIXO DA META	0	15000	Red	Configurar Ação
META ALCANÇADA	15000	25000	Yellow	Configurar Ação
ACIMA DA META	25000	50000	Green	Configurar Ação

Opções ☐ Pré-visualizar

< Voltar Avançar > Cancelar

Este medidor deve ser configurado para que apresente os valores máximos possíveis pela somatória de todos os Funcionários/Vendedores. Caso o valor máximo do gráfico for menor que o valor máximo das vendas de todos os vendedores, o medidor vai apresentar problemas.

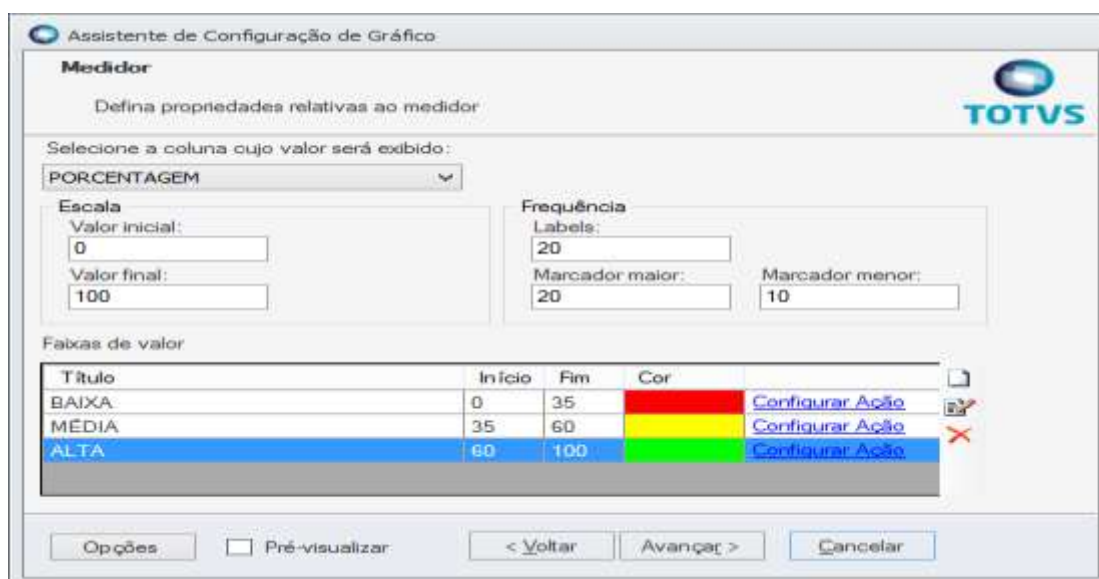
Painel de Metas

Ao criar um novo, deverão ser adicionados primeiramente todos os medidores dos Funcionários/Vendedores e, após incluir todos eles, incluir o medidor do Painel de Metas, que buscará o Resultado da Meta. A diferença deste para os demais é o preenchimento do campo 'Expressão' que deverá somar todos os medidores dos Funcionários/Vendedores, e o tipo, que será 'Composta', conforme abaixo:

Exercício

Primeiramente, precisaremos de uma consulta SQL, a qual retornará o percentual de Usuários Ativos. Assim, pode ser utilizada a consulta abaixo:

```
SELECT COUNT(CODUSUARIO) * 100 / (SELECT COUNT(CODUSUARIO) FROM
GUSUARIO) PORCENTAGEM
FROM GUSUARIO
WHERE GUSUARIO.STATUS = '1'
```



Assistente de Configuração de Gráfico

Medidor

Defina propriedades relativas ao medidor:

Selecione a coluna cujo valor será exibido:
PORCENTAGEM

Escala

Valor inicial: 0
 Valor final: 100

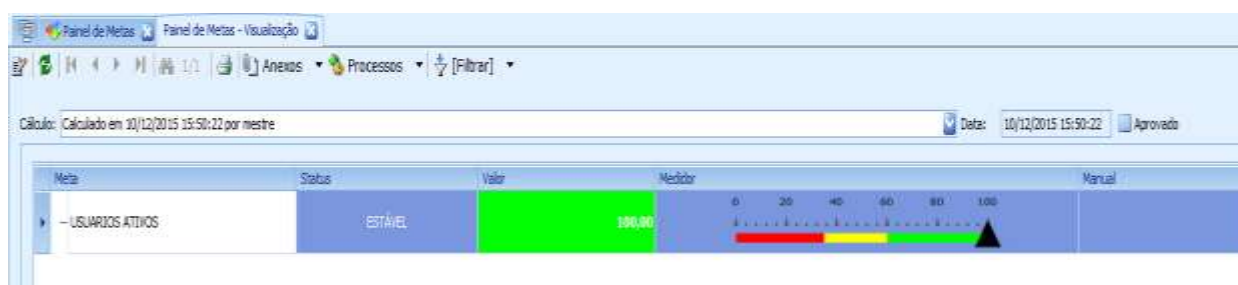
Frequência

Labels: 20
 Marcador maior: 20
 Marcador menor: 10

Faixas de valor

Título	Início	Fim	Cor	Ação
BAIXA	0	35	Red	Configurar Ação
MÉDIA	35	60	Yel	Configurar Ação
ALTA	60	100	Grn	Configurar Ação

Opções: ☐ Pré-visualizar



Exercício

Inclua a consulta SQL, a qual retornará a soma de horas extras pagas aos funcionários. Assim, pode ser utilizada a consulta abaixo:

Horas Extra 50%

```
SELECT SUM(VLOR) VALOR
FROM PFFINANC
WHERE CODEVENTO = 0082
AND MESCOMP BETWEEN 1 AND 12
AND ANOCOMP = XXXX
```

Horas Extra 100%

```
SELECT SUM(VLOR) VALOR
FROM PFFINANC
WHERE CODEVENTO = 0083
AND MESCOMP BETWEEN 1 AND 12
AND ANOCOMP = XXXX
```



Assistente de Configuração de Gráfico

Medidor

Defina propriedades relativas ao medidor

Selecione a coluna cujo valor será exibido:

VALOR

Escala

Valor inicial: 0

Valor final: 10000

Frequência

Label: 1000

Marcador maior: 1000

Marcador menor: 500

Faixas de valor

Título	Início	Fim	Cor	Ação
BAIXA	0	3000	Verde	Configurar Ação
MÉDIA	3000	6000	Amarelo	Configurar Ação
ALTA	6000	10000	Vermelho	Configurar Ação

Opções: ☐ Pré-visualizar

< Voltar Avançar > Cancelar



21. Conclusão

A ferramenta Consultas SQL e Inteligência de Negócios permitem ao usuário a extração de dados do sistema com facilidade e montagem de relatórios gerenciais com rapidez.