

Volume

1

INTERBASE

Operação, Manutenção e Utilização do Banco de Dados.

Interbase V 5.1.1

Marcos Antonio S. A. Ortega

O que é Interbase






Interbase é um banco de dados Cliente/Servidor relacional que compatível com SQL-ANSI-92 , e foi desenvolvido para ser um banco de dados independente de plataformas e de sistemas operacionais.

Este banco de dados, dispensa maiores estruturas dentro da empresa, (DBA / Preparação), onde basta instalar o software e usar – lo, sem a interferência freqüente de profissionais, especializados na manutenção do banco de dados de produção.

Acompanhando, isso tudo ele ainda dispensa o uso de super-servidores , usando pouco espaço em disco para sua instalação e utilizando pouca memória em situações normais de uso. Por isso a plataforma necessária para a sua instalação e utilização pode ser reduzida diminuindo consideravelmente os custos do projeto.

Tendo inicialmente o nome de Groton,, este produto veio sofrendo varias alterações até finalmente em 1986 receber o nome de Interbase iniciando na versão 2.0. Sua origem é compartilhada com a do Sybase, já este se tornou a boa ferramenta que é hoje graças ao investimento de milhões de dólares, pela empresa que adquiriu de seus fundadores.

Borland Adquiriu a empresa Aston-State e ...

-  1994: [Interbase 4.](#)
-  1996: Interbase 4.2.
-  1997: Interbase 5.
-  1999: Interbase 5.6.
-  2000: Interbase 6 - kinobi.

Assim conquistou alguns clientes

- Motorola
- Nokia
- MCI

- Bear Stearns
- The Money Store
- The US Army

- NASA
- Boeing

Limites Teoricamente Máximos do Interbase

I corresponded with a customer in Brazil who claimed to have a little bit less than 200 gigabytes in one database. Of course it had over 90 files. He said that it was growing rapidly, so I suppose it could certainly be over 250GB by now. InterBase's theoretical maximum is 32TB I believe. As for number of users, I believe our tech support team helped a customer using InterBase 4.0 on a huge AIX server with about 600 concurrent users. It took some very careful design in their applications to be as gentle as possible when it came to demands on the database server.

[Claudio Valderrama C.](#)

Tamanho Maximo de Um Banco: 32TB Usando múltiplos arquivos, A maior plataforma conhecida pela InterBase e um banco Brasileiro de aproximadamente acima de 200GB.

Tamanho Maximo de Um Arquivo: 4GB em muitas plataformas e 2GB em algumas

Nº Maximo de Tabelas: 64KB Tabelas

Tamanho Maximo de uma Tabela: 32TB

Nº Maximo de Linhas por tabela: 4GB Linhas

Tamanho Maximo de Cada Linha: 64KB

Nº Maximo de Colunas por tabela: Depende dos tipos de dados em questão (Exemplo: 16,384 INTEGER (4 byte) values per row.)

Nº Maximo de Índices por Tabela: 64KB indexes

Nº Máximo de Indices por Banco de Dados: 4GB

Tipos de Dados

Name	Size	Range/Precision	Description
Varchar(n)	n chars	1 to 32767 bytes	Variable length char or text string
Smallint	16 bits	-2 ¹⁵ to 2 ¹⁵ -1	Signed short (word)
Integer	32 bits	-2 ³¹ to 2 ³¹ -1	Signed long (longword)
Float	32 bits	3.4 x 10 ⁻³⁸ to 3.4 x 10 ³⁸	7 digit precision
Double Precision	64 bits	1.7 x 10 ⁻³⁰⁸ to 1.7 x 10 ³⁰⁸	15 digit precision
*Timestamp	64 bits	1 Jan 100 a.d. to 28 Feb 32768 a.d.	Includes time and date
**Date	32 bits	1 Jan 100 a.d. to 29 Feb 32768 a.d.	
*Time	32 bits	0:00 AM to 23:59.9999 PM	
Blob	<32GB		Stores data of variable indeterminate size
***Numeric (<i>precision, scale</i>)	Variable (16, 32, or 64)	specifies exactly <i>precision</i> digits of precision	Example: Numeric(10,3) holds numbers accurately in the following format: ppppppp.sss
***Decimal (<i>precision, scale</i>)	Variable (16, 32, or 64)	specifies at least <i>precision</i> digits of precision	Example: Decimal(10,3) holds numbers accurately in the following format: ppppppp.sss

- • New to InterBase 6.0
 - ** Date datatype holds both date and time information in versions prior to 6.0
 - *** precisions greater than 9 are stored as exact numerics in 6.0 and non-exact double precision in previous versions

Interbase Server Manager

Ferramenta de uso para a administração e monitoração do banco de dados, inicialmente é um aplicativo para windows, mas pode gerenciar qualquer banco de dados em qualquer servidor.

O Server manager é capaz de resolver as seguintes tarefas:

1. 1. Gerenciar a segurança e permissionamento de usuários;
2. 2. Operações de Backup e Restore;
3. 3. Visualização de Estatísticas tanto do banco de dados quanto do servidor;
4. 4. Realizar tarefas administrativas como:
 - a. a. Checar a integridade do banco de dados
 - b. b. “Limpeza” do banco de dados
 - c. c. Recuperação de transações que não foram terminadas.

Criação de usuários

Visão Geral

A segurança do interbase é centralizada a cada instalação de servidores, ou seja, cada servidor terá seus usuários, que poderão se conectar a ele, o que não significa que poderão acessar os bancos de dados que estão sob seu controle.

Inicialmente um usuário deve ser capaz de acessar o servidor Ib, para posteriormente suas permissões sejam definidas, para cada objeto de cada banco de dados.

Dentro de cada database, também existe o conceito de **ROLES**, ou seja são grupos de usuários, que possuem o mesmo permissionamento ente as tabelas , dentro do banco de dados.

Etapas para a Criação de Usuários

Executem o Server Manager

Efetuem login com o Usuário máster do servidor
Menu Tasks / User Security

** A partir deste momento a usuária Alexandra, já é capaz de se conectar ao banco de dados.

Operações de Backup

Visão Geral

Operações de backup criam uma copia do banco de dados em qualquer mídia disponível no servidor ou na rede <<desde que seja um disco >>, e protegem assim os banco de dados de qualquer falha inesperada, ou da perda de alguma informação.

As operações de backup para o Ib estão repletas de outras características que não simplesmente a função de criar uma copia do banco, algumas delas estão listadas.

1. 1. Realiza uma verdadeira “limpeza”, no banco de dados, removendo registros que foram desabilitados, ou até mesmo alterados;
2. 2. Recupera os espaços ocupados pelos mecanismos de busca, como também os espaços usados pelos registros não mais ativos;
3. 3. Possibilita a alteração da paginação do banco de dados, e / ou possibilita a criação de múltiplos arquivos em discos ou não para um único banco;
4. 4. Possibilita a migração de plataformas, ou upgrade de versões <restrições de tipos de dados diferentes entre eles >.

Um arquivo de backup tem diversas características, são elas;

1. 1. Não possui os índices, por isso é sempre menor que o banco original;
2. 2. Backup de banco de dados que estão em múltiplos arquivos, sempre são gerados em apenas um só arquivo de backup, entretanto podemos ver que no utilitário gsbak, existe uma opção para isso.
3. 3. O arquivo de backup normalmente possui apenas as definições do banco e seus dados, portanto tudo será remontado, novamente o que traz vantagens de velocidade, integridade e economia de espaço ao restaurar um banco.

Medidas de Segurança

É interessante manter o backup em outra mídia que não seja no próprio servidor.

Opções de Backup

1. 1. Transportable Format

Para transportar o banco de dados de um sistema operacional para outro

2. 2. Backup Metadata Only

Realizar um backup apenas das definições de seus objetos, interessante para realizar copias de banco de dados sem nenhum dado realmente, para criação de um novo banco **zerado**.

3. 3. Disable Garbage Collection

Esta opção é para bancos de dados que estão com versões antigas de registros danificadas, e por algum motivo esse dano, não permite que seja feito a operação de backup.

Por default, esta opção não está selecionada.

4. 4. Ignore transactions in limbo

Esta opção faz com que o banco de dados, não realize backup dos registros de transações inacabadas, estes registros podem existir nos bancos quando por alguma falha durante as transações ocorreu e não foi possível para o servidor realizar um Rollbak.

É interessante antes de realizar um backup com esta opção executar o procedimento de recuperação de transações **Maintenance | Transaction Recovery** .

5. 5. Ignore CheckSums

CheckSums, são pontos de verificação que o Ib, realiza a cada gravação de uma pagina de dados. O Ib, não normalmente não permite que seja feito backup de bancos com checkSums não consistentes, isso previne que sejam feitos backups de dados danificados. É interessante examinar os dados após o restore de um backup com essa opção.

6. 6. Verbose Output

Normalmente o Backup exibe uma tela onde consta o início e o fim do backup, e possíveis mensagens de erro que possam existir, com essa opção selecionada ele irá listar todas as operações realizadas durante um backup.

O sistema de backup do interbase permite que usuários estejam conectados ao banco ao se realizar esta operação, no entanto é claro que os dados alterados após o backup ser feito não serão copiados.

Ao Especificar um arquivo de backup que já existe o Ib, irá sobrescrever-lo, atenção ao especificar o nome o path do arquivo.

Os diretórios especificados, tanto como o disco especificado, é da visão da máquina servidora e não de uma estação.

Operações de Restore

Visão Geral

Restaura um backup previamente realizado, Source é o nome do arquivo de backup e Destination, é o nome primário do banco de dados a ser criado.

Notas:

1. Nomes de arquivos a serem restaurados devem conter a letra do drive e os diretórios todos a serem criados.
2. Tipicamente Bancos de dados recém restaurados, são menores que os originais, pois não possuem uma serie de dados que foram “Limpados”, durante a operação de backup;
3. Não é possível restaurar um backup em um drive que não seja local, na visão do servidor.
4. O Ib permite que sejam restaurados bancos de dados que por algum motivo não foi possível remontar os índices da tabela ainda que as opção deactive index, não esteja selecionada, isso pode ocorrer caso não houve espaço em disco suficiente para tal operação ; neste caso o banco estará disponível e os índices podem ser recriados após.

Restaurando Um Banco de Dados Com Múltiplos Arquivos

Você pode otimizar seu banco de dados restaurando em vários arquivos dispostos em vários discos, para isso basta preencher as informações

File path : Drive e Nome do Arquivo

Size : Tamanho que o arquivo pode atingir em N° de paginas de dados, (Tamanho da Pagina * N° de paginas) = Tamanho em Bytes do Arquivo. **O Ib, não ira respeitar o N° de paginas se todos os volumes já foram preenchidos e ainda não restaurou todo o banco.**

Opções de Restore

1. **Page Size** : O Intabase permite que sejam mudadas o tamanho das paginas de dados, isso serve para aumentar a velocidade de acesso aos dados, no caso de grandes índices, ou grandes campos do tipo BLOB, ou ainda banco de dados realmente grandes. O Ib Suporta paginas de tamanho: **1024, 2048, 4096, 8192**; Para bancos de dados com muitas operações de modificação de tabelas uma pagina de dados é mais apropriado.
2. **Replace Existing DataBase**: Sobrescreverá o banco de dados atual;
3. **Commit After Each Table**: O Ib, normalmente restaura inicialmente toda o metadata, para depois restaurar os dados, com esta opção ele irá restaurar cada tabela logo após sua criação no novo banco.
4. **Deactivate Indexes**: Normalmente o Ib, restaura todos os índices automaticamente, mas se por algum motivo estes índices não poderem ser criados, esta opção não recriará nenhum índice , setando-os como inativos, após a restauração podemos ativar-los com o comando ALTER INDEX.
5. **Do Not Restore Validity Conditions** : Ao redefinir as validações de campos de um banco de dados , as vezes ao restaurar-lo alguns registros podem não mais respeitar tais alterações, então selecione esta opção assim ao inserir dados nas tabelas, estes não serão validados. Atenção ao manter uma copia das definições destas validações para restaurar-las logo após a correção dos dados.
6. **Verbose Output** : Lista todas as operações de restore.

Ferramenta de Linha de Comando Gbak:

Está ferramenta é oferecida pelo interbase, para sistemas operacionais, que não possuem interface gráfica, é o aplicativo (DOS), para Backup e restauração de arquivos.

Para Backup

Gbak [-B] [Options] Arquivo de Backup

Para Restore

Gbak [-C | -R] [Options] Arquivo Banco de Dados

Para Restore em multiplos arquivos

Gbak [-C | -R] [Options] Arquivo Banco de Dados tamanho [n1 Banco1 n2 Banco2 ...]

**Pode ser uma boa idéia, Utilizar o Agendador de tarefas do windows 9x ,
ou NT para Agendar Backups diários.**

Demais Operações do Server Manager

Visão Geral

Como vimos no início deste capítulo o Interbase Server Manager é capaz de realizar várias operações em um banco de dados, entre elas as principais são a manutenção de usuários e operações de Backup e Restore. Para não deixar de citar as outras operações vamos fazer uma breve explicação sobre cada uma, sem aprofundar, pois levando em consideração que o objetivo desta apostila não é de se aprofundar muito nestas matérias.

1. **Visualização de Estatísticas:** Menu **TASKS | DATA BASE STATISTICS**, esta operação mostra as estatísticas de um banco de dados, com informações sobre a última transação, páginas alocadas, para análise.
2. **Análise de Banco de Dados:** Menu **VIEW | DATA BASE ANALYSIS** mostra uma análise mais detalhada, Objeto a objeto do banco de dados.
3. **Análise de Locks :** Menu **TASKS | LOCK MANAGER STATISTICS**, demonstra dados sobre o mecanismo de lock, no banco de dados, e mostra informações sobre locks, e páginas de dados alocadas, para dimensionamento das páginas de dados.
4. **Checar a Integridade do Banco de Dados:** Menu **MANTEINCE | DATA BASE VALIDATION** Verifica se os bancos de dados contêm algum erro não detectado ainda pelos usuários.
5. **“Limpeza” do Banco de Dados:** Menu **MANTEINCE | DATA BASE SWEEP**, executa uma limpeza imediata dos registros que foram executados o comando RollBack, ou versões de registros antigas; Essa operação é executada automaticamente, após um número especificado de “commits”, ver Propriedades de Banco de dados. Menu **MANTEINCE | TRANSACTION RECOVERY** normalmente o Interbase possui um mecanismo de duas fases de commits de transações, se algumas destas fases falhar por alguma razão a transação não será efetuada, e isso pode deixar alguns vestígios nas tabelas de sistema, essa opção recupera essas tabelas de transações pendentes.

IB_WISQL

Visão Geral

Devido a pobre interface de manutenção e de pesquisas interativas ao banco de dados, muitas empresas pelo mundo a fora, desenvolveram vários aplicativos, de manutenção e pesquisas para o banco de dados Interbase, dentre elas posso destacar o MARATHON da Ib Objects, e o IB_WISQL, da mesma empresa.

Escolheremos o IB_WISQL, devido ao seu grande numero de usuários, (talvez por ser um freeware, mas particularmente uso o Marathon que possui uma interface mais amigável para desenvolvedores).

Antes de tudo..

O Interbase possui compatibilidade com a linguagem de pesquisa a banco de dados ANSI-SQL92, isso quer dizer que existem vários padrões para pesquisa em banco de dados relacionais, onde cada software assume um padrão e em certos casos realiza algumas alterações para facilitar ou para oferecer maiores vantagens sobre a manipulação de objetos e dados.

No nosso caso o Ib, possui duas linguagens distintas são: DSQL, e SQL, onde DSQL é usado para alterações no meta data, e SQL para pesquisas e alterações nos dados propriamente ditas.

Ex. DSQL

```
CREATE TABLE , ALTER TABLE
```

```
CREATE PROCEDURE , ALTER PROCEDURE
```

Ex SQL

```
SELECT, UPDATE, INSERT
```

Objetos de Banco de Dados

Visão Geral

Um banco de dados possui vários tipos de objetos, para armazenar, garantir a integridade dos dados e ainda manipular estes dados, vamos para uma visão didática destes objetos, trazendo uma curta definição destes. Não é o objetivo desta apostila fornecer informações detalhadas sobre a sintaxe de criação / manutenção destes objetos apenas uma breve explicação.

CHECKS CONSTRAINTS : São restrições pré definidas a colunas ou até a tabelas onde apenas serão aceitas informações dentro destas colunas se respeitadas forem os checks constraints.

DOMAINS: São tipos de dados criados pelo desenvolvedor para padronização de certos campos de tabelas dentro do banco de dados, esses tipos de dados, sempre são derivados de algum tipo de dado existente, onde o desenvolvedor pode inserir restrições aos dados que serão armazenados por estas colunas.

COLUNAS: São repositórios de dados, que podem ser de um tipo nativo, ou de um tipo domain pré-definido.

TABELAS: São repositórios de dados, um conjunto de colunas e linhas.

VIEWS: São tabelas criadas em tempo de execução, pré-definidas por um comando SELECT, onde podem ser agrupados e / ou filtrados os dados de uma ou mais tabelas.

TRIGGERS: São gatilhos que são disparados sempre que ocorrer alguma alteração em uma tabela ou view, o Ib possui três tipos de triggers para cada tipo de alteração em uma tabela. Antes de Acontecer e depois de Acontecer (Update, Insert , Delete),

STORED PROCEDURES: São procedimentos de visualização ou manutenção de dados, estes procedimentos são pré-programados, e executados aleatoriamente seguindo a vontade dos usuários.

Estes são os principais objetos que um banco de dados que arquitetura cliente servidor possui, é com a utilização destes que as regras de negocio do sistema serão definidas dentro do banco de dados, garantindo assim a variedade de front-ends, e ainda a integridade dos dados.

Voltando ao Ib_WISQL

Tela principal do aplicativo, onde temos principalmente as seguintes funções

Create a Database : Cria um banco de dados interbase , vazio no caminho especificado , muita atenção para não criar um banco vazio encima de algum que já exista.

Connect a Database : Conecta – se a um banco de dados existente ,

Browse : Demonstra e permite alterações em todos os objetos existente no banco de dados.

DUMP: Exporta Dados para arquivos texto, delimitados em configuração

PUMP: Importa dados de um arquivo texto, delimitado em configuração.

SCRIPT: Executa um Arquivo.SQL, que é uma seqüência de comandos DSQL, ou SQL, para manutenção de banco de dados.

Browse

Esta tela permite a visualização e manutenção dos objetos do banco de dados.

Permite a visualização e manutenção de diversos dados e objetos ligados as tabelas e seus componentes.

Permite a visualização e manutenção de diversas Stored Procedures.

Comandos DSQL / SQL

Visão Geral

São comandos que modificam a estrutura ou os dados dos objetos de um banco de dados. Esses comandos são de vital importância para manutenção e expansão do banco. Cada comando seja DSQL ou SQL, o Db, antes de executar irá compilar, com isso serão garantidos a integridade, velocidade e o início e término do comando.

Lista de Funções

Avg()	Média Aritimética
Cast()	Conversão de tipos de dados
Count ()	Retorna o numero de linhas que satisfaz a condição da expressão SQL
Max ()	Retorna o Valor Maximo de um conjunto de valores
Min()	Retorna o Valor mínimo de um conjunto de valores
Sum ()	Retorna a soma de um conjunto de valores
Upper ()	Retorna caracteres com caixa alta.

Lista e Sintaxe dos comandos mais utilizados

Create Shadow

Cria um ou mais copias sincronizadas de um banco de dados, é usado para garantir o acesso continuo aos dados. Cada copia do banco de dados pode ser um único arquivo ou um conjunto de arquivos (conjunto de Shadow).

Em caso de falha no disco, o “DBA”, pode ativar uma das “Shadows”, para substituir o banco de dados, provisória ou permanentemente.

EXEMPLO :

CREATE SHADOW 1 AUTO ‘AGRICOLA.SHD’

**CREATE SHADOW 1 AUTO CONDITIONAL ‘C:\SISTEMA
AGRICOLA\UM\AGRSHADOW.SHD’**

Ativado as Copias:

O Interbase permite que as copias sejam ativadas manualmente, ou automaticamente dependendo de como elas foram criadas.

Auto	Automaticamente retoma a posição do banco de dados em casos de falha, e desconecta qualquer shadow existente; Permite a continuidade das operações mesmo sem a acessibilidade da copia.
Manual	Retoma manualmente a posição do banco de dados (gfix -a agrícola.shd), e não permite que o banco continue operando em caso de não acessibilidade da copia
Conditional	Recria uma nova copia em caso de promoção para a condição de banco de dados.

Grant

Configura permissões para usuários, ou grupos de usuários aos objetos do banco de dados.

A expressão Grant seta privilégios a os roles e objetos do banco de dados aos usuários

All	Todos os tipos de acessos	Delete	Acesso de Deleção
Select	Acesso apenas de pesquisa	Insert	Acesso de Inserção de linhas
Update	Acesso de Alteração	Execute	Acesso para execução

EXEMPLO :

GRANT INSERT, SELECT, UPDATE ON TBLAPONTAMENTOS TO JAIME

Insert

Insere uma ou mais linhas em uma tabela

Podemos utilizar o comando insert , em combinação com o comando Select, para adicionar mais de uma linha a uma tabela. Não é recomendado a inserção de linhas selecionadas da tabela de destino , isso pode causar um loop infinito !.

EXEMPLO :

```
Insert into TblMotoristas ( CodMotorista, Nome, NroFaltas, CodTpVeiculos, TerceiroPropio ) Values ( 1000, 'Antonio Carlos', 0, -1, 'P' )
```

```
Insert Into TblMotoristas ( CodMotorista, Nome, NroFaltas, CodTpVeiculos, TerceiroPropio ) Values (
```

```
Select CodFuncionario, Nome, 0, -1, 'P'
```

```
From TblFuncionarios
```

```
Where CodFuncionario => 1100 )
```

Atenção

Todos os comandos insert, executados devem ser precedidos do comando commit, para concretização das operações ou do comando rollback, para descartar-las.

Update

Altera uma ou mais linhas de uma tabela, iniciando implicitamente ou explicitamente uma transação.

O Comando Update altera colunas e linhas específicas, de uma tabela, respeitando uma cláusula 'where', ou não caso for para alterar todas as linhas.

Uma alteração só será visualizada por outro usuário, após um comando de commit, até que este seja executado, os registros alterados, estarão com acesso indisponível para alteração por outros usuários.

EXEMPLO :

```
Update TblMotoristas  
Set Faltas = 0
```

```
Update TblMotoristas  
Set Faltas = 0,  
Nome = 'Sr' + 'Nome'  
Where Sexo = 'M'
```

Atenção

Todos os comandos Update, devem ser precedidos de commit, ou rollback para concretizar-los ou descartá-los.

Select

É o principal comando para consultar banco de dados, este comando pode ser aplicado em vários tipos de objetos tais como : Tabelas, views, Stored Procedures.

Um Comando Select é Capaz de :

1. Retornar todas as linhas e colunas de uma tabela, view ou stored procedure;
2. Retornar uma única linha ou parte de uma linha (selecionando colunas especificamente) de uma tabela
3. Retornar muitas linhas de muitas tabelas, relacionadas pela clausula join entre campos de relacionamento;
4. Unificar duas ou mais tabelas parcialmente ou integralmente
5. Sumarizar informações de uma ou mais tabelas.

EXEMPLOS :

```
Select *  
From TblApontamentos
```

Retorna todas as linhas e colunas da tabela de apontamento.

```
Select NroApontameto, CodTurma, DataInicio  
From TblApontamentos
```

Retornará as colunas NroApontamento, CodTurma, DataInicio da tabela de apontamentos

CLAUSULA Where

```
Select NroApontameto, CodTurma, DataInicio  
From TblApontamentos  
Where DataInicio Between '06/01/2000' and '06/30/2000'
```

A Clausula Where funciona para restringir a pesquisa em linhas, selecionadas e que satisfação suas condições.

Retornará as colunas NroApontamento, CodTurma, DataInicio da tabela de apontamentos no intervalo do mês de junho

CLAUSULA JOIN

Um banco de dados relacional, para produzir informações, possui objetos correlacionados entre si. Estes relacionamentos são definidos como chaves estrangeiras entre tabelas.

É seguindo estas chaves estrangeiras que a clausula join opera, para informar ao mecanismo de procura do banco de dados por onde irá correlacionar suas linhas para desta forma produzir a informação desejada.

Tabela de apontamentos 1

A Clausula Join irá Apontar os Nro dos Apontamentos entre as tabelas de apontamento e a tabela de movimento de funcionários.

```
Select Apto.NroApontamento, Apto.DataInicio, Apto.CodTurma,  
Turm.CodTurma, Turm.Nome  
From TblApontamentos Apto Join  
TblTurmas Turm On ( Apto.CodTurma = Turm.CodTurma )  
Where NroApontamento Between 100 and 103
```

```
Select CCus.CodCCusto, CCus.Nome,  
Oper.CodCCusto, Oper.CodOperacao, Oper.Nome  
From TblCentroCusto CCus Join  
TblOperacoes Oper On ( CCus.CodCCusto = Oper.CodCCusto And  
Oper.CodOperacao Between 100 and 200 )
```

A Clausula Join também pode operar com uma restrição para pesquisa!

É importante seguir todos os relacionamentos, em uma clausula join existentes entre as tabelas que se deseja relacionar, pois quanto mais restrições e relacionamentos forem especificados pelo comando select, mais rápida será a pesquisa.

Chaves estrangeiras e chaves primarias de tabelas

São na verdade índices, que devem ser usados para possibilitar ao mecanismo de procura do Ib, uma pesquisa mais veloz!

CLAUSULA Group By

Esta Clausula é usada para organizar agrupamentos de informações, normalmente para produzir informações totalizadas, ou únicas.

É importante notar o mecanismo de agrupamento, pois nem sempre é possível realizar os agrupamentos aleatoriamente, principalmente em campos que não são correlacionados entre si.

Uma expressão SELEC, com a clausula GROUP BY não pode ter campos que não façam parte da sumarização ou das totalizações.

EXEMPLO :

```
Select Via.CodMotorista, Mot.Nome, Sum( Via.PesoLiquido ) as Toneladas,  
Count( * ) as NroLinhas  
From TblViagens Via Join  
TblMotoristas Mot On ( Via.CodMotorista = Mot.CodMotorista )  
Where Via.DataReferencia Between '05/01/2000' and '05/30/2000' And  
Via.PesoTara > 0  
Group By Via.CodMotorista, Mot.Nome
```

Esta expressão não iria compilar se a clausula group by não estivesse completa, com o código e o nome do motorista, pois estes dados são repetidos em varias linhas dentro da tabela de viagens. Sem a especificação completa das informações sumarizadas e sem as funções de sumarização não é possível executar a consulta.

A clausula Group By

É na verdade um agrupador de informações e normalmente é usada em conjunto das funções de sumarização (Sum, Avg, Min, Max) , para retorno de apenas uma linha por satisfação do agrupamento !

CLAUSULA Order By

Esta Clausula é usada para classificar em ordem crescente ou decrescente a pesquisa

```
Select Via.CodMotorista, Mot.Nome, Sum( Via.PesoLiquido ) as Toneladas, Count( * ) as NroLinhas  
From TblViagens Via Join  
    TblMotoristas Mot On ( Via.CodMotorista = Mot.CodMotorista )  
Where Via.DataReferencia Between '05/01/2000' and '05/30/2000' And  
    Via.PesoTara > 0  
Group By Via.CodMotorista, Mot.Nome  
Order By Mot.Nome Asc
```

O Prefixo ASC Significa que a pesquisa será Ascendente, e precedido de DESC, a pesquisa será Descendente.

Tambem podemos ordenar campos sumarizados

```
Select Via.CodMotorista, Mot.Nome, Sum( Via.PesoLiquido ) as Toneladas, Count( * ) as NroLinhas  
From TblViagens Via Join  
    TblMotoristas Mot On ( Via.CodMotorista = Mot.CodMotorista )  
Where Via.DataReferencia Between '05/01/2000' and '05/30/2000' And  
    Via.PesoTara > 0  
Group By Via.CodMotorista, Mot.Nome  
Order By 3
```