

Administração de Banco de Dados Oracle

Por

Edson Almeida Junior



Consulting Tecnologia & Sistemas Ltda
www.consulting.com.br

Março – 2006

Ultima Atualização 11/04/2006

ÍNDICE

INTRODUÇÃO.....	4
BANCO DE DADOS RELACIONAL.....	4
CONCEITOS.....	4
SQL - LINGUAGEM DE PESQUISA ESTRUTURADA.....	5
OUTROS CONCEITOS DE BANCO DE DADOS.....	7
BANCO DE DADOS DISTRIBUÍDOS.....	7
DATA WAREHOUSE.....	7
REPLICAÇÃO DE DADOS.....	8
ATRIBUIÇÕES/RESPONSABILIDADES DE UM DBA (DATABASE ADMINISTRATOR).....	8
ARQUITETURA ORACLE.....	9
ESTRUTURAS FÍSICAS.....	10
INSTÂNCIA ORACLE.....	11
OBJETOS DO BANCO DE DADOS ORACLE.....	11
PROCESSANDO COMANDOS SQL.....	13
DATABASE BUFFER CACHE.....	13
SEGMENTOS DE RECONSTRUÇÃO: O "DESFAZER" DO ORACLE.....	14
OS PROCESSOS ORACLE.....	15
PRINCIPAIS PROCESSOS DE SEGUNDO PLANO (BACKGROUND PROCESS).....	15
CONFIGURAÇÕES DE SERVIDORES DE BANCO DE DADOS.....	16
SERVIDOR MULTILINEAR (MULTI-THREADED).....	17
SERVIDOR DEDICADO.....	18
LOG DE TRANSAÇÕES (LOG DE RECONSTRUÇÃO: REDO).....	18
REDO LOG FILES.....	18
LOG SWITCHES.....	19
CHECKPOINTS.....	19
O ARQUIVO DE PARÂMETROS.....	20
EXERCÍCIO I.....	22
INICIALIZANDO E TERMINANDO UM BANCO DE DADOS ORACLE.....	22
INICIALIZANDO (STARTUP).....	22
TERMINANDO UM BD ORACLE (SHUTDOWN).....	23
ALTER DATABASE: ALTERANDO O ESTADO DO BANCO DE DADOS.....	23
ALTER SYSTEM: RESTRINGINDO O ACESSO.....	24
GERENCIANDO O ESPAÇO DO BANCO DE DADOS.....	25
COMPONENTES DE UM BLOCO ORACLE.....	26
CONTROLANDO O ESPAÇO LIVRE.....	26
ESTUDANDO AS TABLESPACES.....	26
criação de TABLESPACES.....	27
ALTERANDO UMA TABLESPACE.....	27
DELETANDO UMA TABLESPACE.....	28
criando segmentos de RECONSTRUÇÃO (ROLLBACK SEGMENTS).....	28
criando um BANCO DE DADOS.....	29
ALTERANDO UM BANCO DE DADOS.....	30
INSTALAÇÃO DO ORACLE.....	30
O ARQUIVO DE PARÂMETROS: ESTUDANDO OS PARÂMETROS.....	30
criação e GERENCIAMENTO DE USUÁRIOS DO BANCO DE DADOS.....	31
ALTERANDO UM USUÁRIO DO BANCO DE DADOS ORACLE.....	32
DELETANDO UM USUÁRIO.....	33
PRIVILÉGIOS.....	33
COMANDO GRANT.....	34
COMANDO REVOKE.....	34

AGRUPANDO PRIVILÉGIOS: “ROLE”	35
VISÕES (VIEW’S) DO DICIONÁRIO DE DADOS	36
BACKUP E RECOVERY DO BANCO DE DADOS	36
BACKUP COMPLETO (“FULL”) OFFLINE	37
RESTAURANDO (RECOVERY) DO BD	37
MODOS DE ARQUIVAMENTO ORACLE	38
HABILITANDO/DESABILITANDO O MODO DE ARQUIVAMENTO	38
BACKUP ONLINE PARCIAL	39
EXPORT/IMPORT/LOAD DO ORACLE	40
USANDO A BARRA DE FERRAMENTAS ADMINISTRATIVAS DO ORACLE	41
ORACLE NA INTERNET	41
BIBLIOGRAFIA	42

INTRODUÇÃO

Este material é uma síntese da Introdução e Administração de Banco de Dados Oracle, e visa transmitir aos administradores do banco de dados Oracle conhecimentos básicos para o gerenciamento/administração do mesmo.

BANCO DE DADOS RELACIONAL

Um banco de dados relacional pode ser definido como um banco de dados que aparece ao usuário como uma coleção de tabelas. A característica mais fundamental num banco de dados relacional é a de que temos que identificar univocamente seus registros (que chamamos de linhas), ou seja, devemos definir um campo (ou mais de um) dentre os vários campos do registro, para ser o atributo identificador do mesmo (chave primária).

O Oracle é um SGBD Relacional - Sistema de Gerenciamento de Banco de Dados Relacional (RDBMS - Relational Database Management System) e foi um dos primeiros sistemas de banco de dados a utilizar a Linguagem de Pesquisa Estruturada (SQL - Structured Query Language) como interface do usuário sendo, talvez, por este motivo, agora um dos principais sistemas de gerenciamento de banco de dados.

CONCEITOS

No estudo de banco de dados convém sabermos algumas definições importantes:

Tabelas - São as unidades básicas de um SGBD Relacional. É formada por linhas e colunas, onde as linhas representam os registros e as colunas os campos da tabela. São muito conhecidas como Tabelas-Base.

Exemplo:

Cod_Funcionario	- CHAR(06)
Cod_Departamento	- CHAR(05)
Nome_Funcionario	- VARCHAR2(40)
Data_Nascimento	- DATE

Colunas - Um nome de coluna é o nome que o usuário informará para representar os valores de dados que serão entrados sob cada coluna.

Linhas - São os registros da tabela.

Valores de dados - São os valores encontrados na interseção de uma coluna e uma linha.

Chave primária - Uma ou mais de uma coluna que identifica uma linha, facilitando em muito o acesso aos dados.

Visão ou Tabela Virtual(View) - É uma tabela formada por uma pesquisa em uma ou mais tabela-base. Ela não existe, mas parece ao usuário como se existisse.

Toda tabela num banco de dados relacional tem que possuir uma Chave Primária. As tabelas poderão também ter chaves estrangeiras, que são campos que relacionam as tabelas do banco de dados, fazendo referência entre uma e outra tabela.

SQL - LINGUAGEM DE PESQUISA ESTRUTURADA

A **SQL - Linguagem de Pesquisa Estruturada** (Structured Query Language) devido as suas características tornou-se a linguagem padrão de banco de dados relacionais.

A Linguagem **SQL** pode ser usada através de dois modos:

SQL Interativa - Onde os comandos SQL são digitados interativamente, ou seja, logo após a digitação do comando vemos sua execução.

SQL Embutida - Neste modo os comandos SQL ficam embutidos no programa-fonte de uma linguagem de programação. Tal linguagem é normalmente chamada linguagem hospedeira. Linguagens de programação como COBOL, C, PASCAL, PL/1, etc, admitem a SQL Embutida.

A SQL divide-se em três grupos:

- Linguagem de Definição de Dados (DDL)
- Linguagem de Controle de Dados (DCL)
- Linguagem de Manipulação de Dados (DML)

Linguagem de Definição de Dados (DDL)

Tem como objetivo definir, alterar e eliminar as tabelas usadas para armazenar os dados. CREATE TABLE, CREATE VIEW, são comandos de definição de dados, dentre outros.

Exemplo: Criação de Tabelas

Em SQL as tabelas possuem a função de armazenar os dados do Banco de Dados. O comando para criação de tabelas no SQL é o CREATE TABLE.

Exemplo de CREATE TABLE:

```
CREATE TABLE T_Funcionario
(cod_funcionario          VARCHAR(2),
 nome_funcionario        VARCHAR(30),
 end_funcionario          VARCHAR(40),
 CONSTRAINT pk_funcionario PRIMARY KEY
 (cod_funcionario),
 TABLESPACE ST_funcionario;
```

Linguagem de Controle de Dados (DCL)

A DCL contém elementos que serão úteis num sistema multiusuário, onde a privacidade das informações é importante, bem como a segurança das tabelas e o estabelecimento de fronteiras entre as transações. Os comandos GRANT e REVOKE são alguns dos comandos utilizados para o controle de dados.

Linguagem de Manipulação de Dados (DML)

Esta linguagem contém os componentes da linguagem e conceitos para a manipulação das informações armazenadas nas tabelas. Os comandos SELECT, UPDATE, INSERT, bem como outros, são comandos de manipulação.

O Comando mais utilizado na linguagem SQL para manipulação dos dados é o SELECT. Com ele é que produzimos as “Queries”, ou seja, as pesquisas no Banco de Dados.

Exemplos: Selecionando Dados

a) `SELECT * FROM ALUNO;`

Retorna todas as linhas e todas as colunas da tabela ALUNO.

b) `SELECT MATRICULA, NOME_ALUNO
FROM ALUNO;`

Retorna todas as linhas das colunas matrícula e o nome do aluno da tabela ALUNO.

c) `SELECT MATRICULA, NOME_ALUNO
FROM ALUNO
WHERE NOME_ALUNO = 'JOAO DA SILVA';`

Retorna a matrícula e o nome do aluno que tenham nome igual a JOAO DA SILVA.

d) `SELECT MATRICULA FROM CURSA
WHERE ((NOTA1+NOTA2+NOTA3)/3) > 7
AND CD_DISCIPLINA = 'PORT';`

Retorna a matrícula de todos os alunos que obtiveram média acima de sete na disciplina PORT.

e) `SELECT AVG(SALARIO) FROM EMPREGADO;`

Retorna a média dos salários dos funcionários da tabela.

Obs.:

1) A Sintaxe completa do comando SELECT encontra-se no manual de referência de SQL da ORACLE.

2) Podemos usar o comando Select para acessar dados de duas ou mais tabelas, ao mesmo tempo:
`SELECT COD_EMPREGADO, NOME_EMPREGADO, SALARIO_EMPREGADO
FROM FUNCIONARIO, SALARIO
WHERE FUNCIONARIO.COD_EMPREGADO = SALARIO.COD_EMPREGADO;`

Esta “Query” acima retorna os dados da tabela FUNCIONARIO (COD_EMPREGADO e NOME_EMPREGADO) e da tabela SALARIO (SALARIO_EMPREGADO).

OUTROS CONCEITOS DE BANCO DE DADOS

BANCO DE DADOS DISTRIBUÍDOS

Permitem o manuseio da informação armazenada no Banco de Dados que se encontra distribuído em diversas máquinas geralmente em locais geográficos distintos. A distribuição dos dados pode ser feita no nível de uma entidade do esquema conceitual (1 tabela, por exemplo) ou através de fragmentos (partições da tabela). Os fragmentos podem ser obtidos através do uso de técnicas de fragmentação horizontal (seleção de linhas da tabela), fragmentação vertical (seleção de colunas da tabela) ou de forma híbrida (combinando a fragmentação horizontal e vertical ao mesmo tempo).

A alocação dos fragmentos pode ser realizada com ou sem replicação. No primeiro caso, cópias de um mesmo fragmento são distribuídas pelos diversos nós que compõem o Banco de Dados distribuído.

O principal objetivo dos bancos de dados distribuídos é disponibilizar o dado o mais próximo possível do local onde é mais acessado, permitindo um aumento no desempenho de consultas devido ao aproveitamento da localidade de acesso e também, por trabalhar com um volume menor de dados.

DATA WAREHOUSE

Em sua forma essencial, um data warehouse é um repositório para informações organizacionais. Pode incluir tabelas relacionais geradas pelos sistemas de controle e acompanhamento, dados consolidados altamente estruturados, documentos e objetos multimídia. Também podem ser encapsuladas regras de negócio. Sem dúvida, uma solução data warehouse deve ser aberta, escalável, com velocidade e facilidade de acesso, e deve ser gerenciável.

Dada a tendência da indústria para a distribuição de recursos e o desejo de maior controle e acessibilidade, uma solução deve ser projetada para operação geograficamente distribuída ao longo das unidades de negócio da corporação. Facilidade de instalação, ajustes, gerenciamento, controle e interoperabilidade são pontos críticos, particularmente em um ambiente distribuído.

Data warehouse é uma arquitetura, não um produto, e não deve ser associada com um repositório de dados especializado, tal como um sistema de gerenciamento de dados multidimensional, e pode, de fato, ser perfeitamente hospedado em um sistema de gerenciamento de dados relacional, que provê uma base mais sólida sobre a qual se construirá as aplicações que alimentarão o repositório de dados.

Mais importante que a ferramenta para hospedar o repositório é o conjunto de aplicações e ferramentas que compõem as informações organizacionais. Data warehouse é uma forma de unificar e facilitar o acesso a todas as informações de que os membros de uma corporação necessitam para a execução, controle e planejamento de atividades de negócio.

A construção do repositório é consequência da Administração de Dados, que é responsável pela elaboração de um modelo global de dados para a corporação, que normalmente surge como resultado da integração dos sistemas de transações on line. Para o usuário, aspectos importantes são as ferramentas de query e análise de dados, bem como o grau de integração com as

ferramentas de automação de escritório, que correspondem ao dia-a-dia dos usuários corporativos.

Muitos vendedores consideram gerenciadores de banco de dados especializados para data warehouse, mas esses desconsideram as melhorias da tecnologia relacional nas áreas de processamento e otimização em paralelo de queries, bem como a integração com ferramentas de acesso, modelagem, apresentação e análise, tais como as planilhas eletrônicas e bancos de dados de mesa.

O Oracle Warehouse suporta além de dados estruturados da forma relacional, outros tipos de dados, como por exemplo, textos desestruturados, dados espaciais, vídeo, ...

A proposta do Oracle Warehouse é a de possibilitar o gerenciamento dos dados da empresa, independente da origem dos dados, ou seja, se são dados históricos, de produção, etc.

REPLICAÇÃO DE DADOS

O Oracle permite a replicação dos dados, isto é, cópias de tabelas dos bancos de dados podem ser instaladas nas unidades regionais da empresa, visando assim a rapidez no acesso aos mesmos (evita o tráfego na rede).

ATRIBUIÇÕES/RESPONSABILIDADES DE UM DBA (DATABASE ADMINISTRATOR)

O DBA é o administrador do banco de dados Oracle e tal função, geralmente é exercida por um Analista de Sistemas com alguma formação/experiência em Banco de Dados.

Diversas atividades de administração do SGBD Oracle, devem ser exercidas pelo(s) DBA(s).

Dentre elas destacamos:

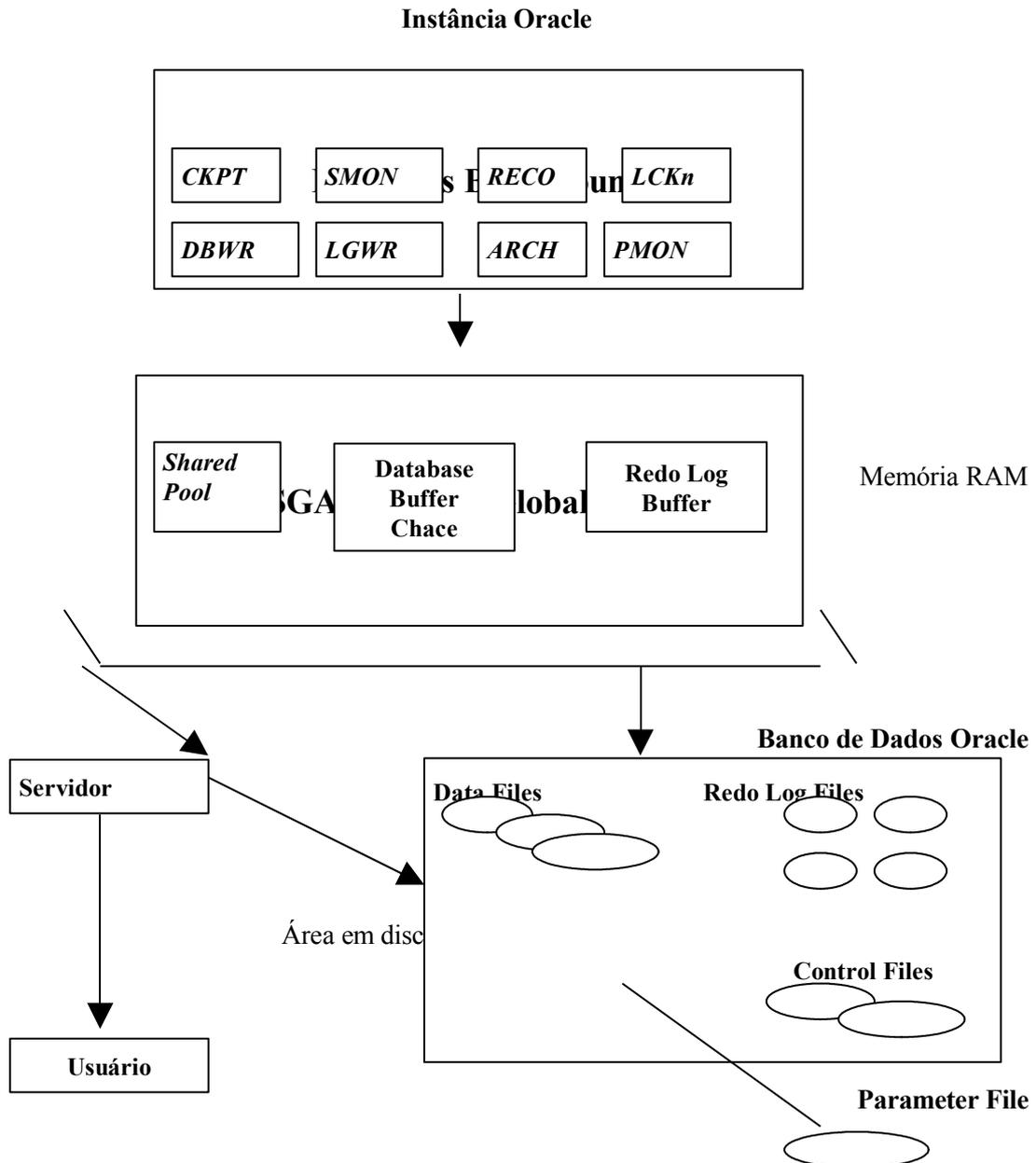
Instalar e fazer o Upgrade do Oracle Server e das ferramentas diversas da Oracle;

- Criar as estruturas e objetos primários;
- Planejar/calcular, bem como, disponibilizar o espaço necessário para o Banco de Dados Oracle;
- Modificar a estrutura do banco , quando necessário;
- Criar, deletar, dar/remover privilégios, enfim, gerenciar os usuários do banco;
- Gerar cópias de segurança (Backup's) e recuperar o banco Oracle;
- Manter a segurança do sistema, controlando o acesso ao banco de dados utilizando os recursos de segurança do sistema operacional, do sistema de rede e do Oracle;
- Monitorar e otimizar a performance do Banco de dados.

ARQUITETURA ORACLE

O DBA deve conhecer a arquitetura Oracle, isto é, como o Oracle “trabalha”, tendo assim subsídios para a administração do(s) banco(s).

A Arquitetura Oracle



Arquitetura Oracle = Instância + BD Oracle

Instância = Memória + Processos Background

ESTRUTURAS FÍSICAS

O Banco de Dados Oracle é composto por 3 tipos de estruturas físicas (arquivos):

Dados (data files);
Controle (control files);
Log (redo log files) e
Parâmetros (parameter file).

Data files – Contém todos os dados do banco. Estruturas lógicas, tabelas, índices, visões e outros objetos são armazenados nestes arquivos. Um ou mais data files podem compor uma tablespace (objeto lógico que armazena os arquivos de dados).

Control files – Arquivos que armazenam informações sobre as estruturas físicas do banco de dados (nome, localização, ...). Todos os data files e redo log files são identificados no control file, bem como o nome do banco de dados.

Recomenda-se que se tenha no mínimo dois Control Files, armazenados em discos diferentes (se possível).

Redo log files – Arquivos que guardam todas as alterações efetuadas nos dados das tabelas do banco e são utilizados para recuperação destas.

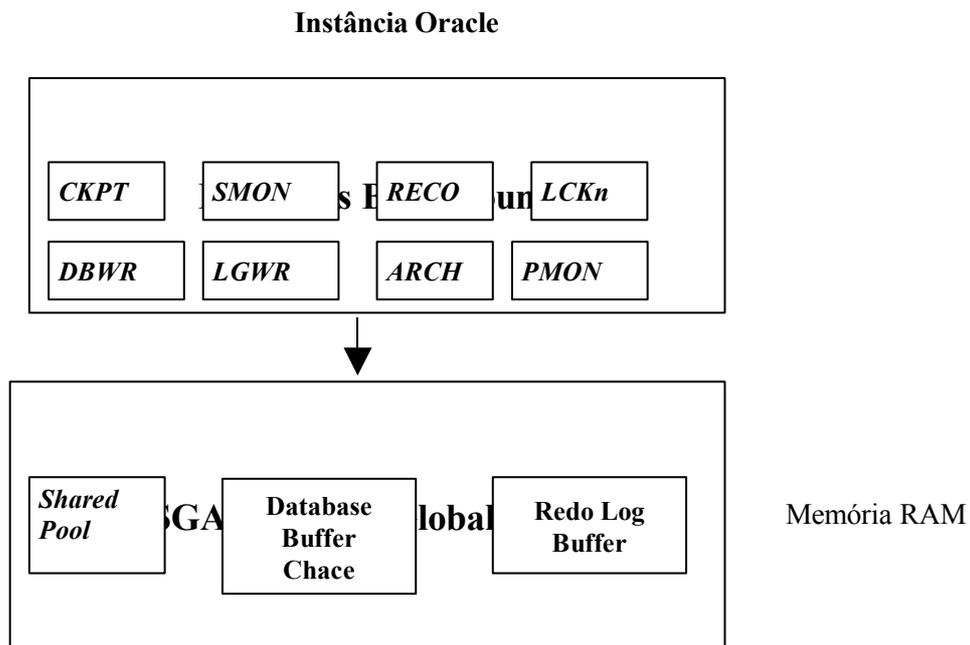
Para garantir uma boa recuperação cada redo log file deve Ter no mínimo uma réplica. Um conjunto de redo log files iguais recebe o nome de grupo, e um grupo é formado por membros (redo log file original e suas réplicas).

Ao terminar o espaço disponível nos membros de um grupo redo log, a gravação continua nos membros do grupo seguinte (log switch). O grupo anterior fica então disponível para Ter seu conteúdo copiado para outro local (archive log files), pois, ao terminar o espaço no último grupo, o processo irá reutilizar o primeiro, sobregravando-o.

Além dos 3 tipos de arquivos citados acima temos também o Parameter file – Para inicializar uma instância, o Oracle deve ler um arquivo de parâmetros de nome padrão init<nome_da_instância>.ora. Este arquivo é do tipo texto e contém diversos parâmetros de configuração associados a uma instância, aos quais devem ser atribuídos valores específicos.

Os valores dos parâmetros serão utilizados para inicializar muitas das configurações de memória e processos da instância Oracle. Cada instância tem seu arquivo de parâmetros que define as particularidades de cada ambiente.

INSTÂNCIA ORACLE



A cada inicialização do banco Oracle, parte da memória RAM do computador é alocada para a SGA (System Global Area) e processos Oracle de segundo plano (Background Process) são inicializados, visando controlar o banco. Então uma Instância Oracle (Instance) é o conjunto da SGA (buffers da memória RAM) + processos Oracle de segundo plano (Background), que juntos oferecem aos vários usuários o acesso ao banco de dados.

Existe um banco de dados para cada instance, e é possível criar em um mesmo equipamento mais de uma instance, sendo, por exemplo, uma para conter um banco de dados de produção e outra para conter o banco de dados de teste. Neste caso, haverá processos sendo executados em background específicos para cada instance e uma área de memória independente utilizada em cada instance.

Obs.:

- 1) SGA – Memória compartilhada por todos os usuários Oracle.
- 2) Instância (instance) = SGA + Processos “Background”
- 3) É no arquivo de parâmetros (parameter file) que determinamos as características da instância (instance)

OBJETOS DO BANCO DE DADOS ORACLE

O Oracle possui diversos componentes denominados Objetos do Banco de Dados . Dentre eles podemos destacar:

Tabelas - São as unidades básicas de um SGBD Relacional. É formada por linhas e colunas, onde as linhas representam os registros e as colunas os campos da tabela. São muito conhecidas como Tabelas-Base (conforme já vimos anteriormente).

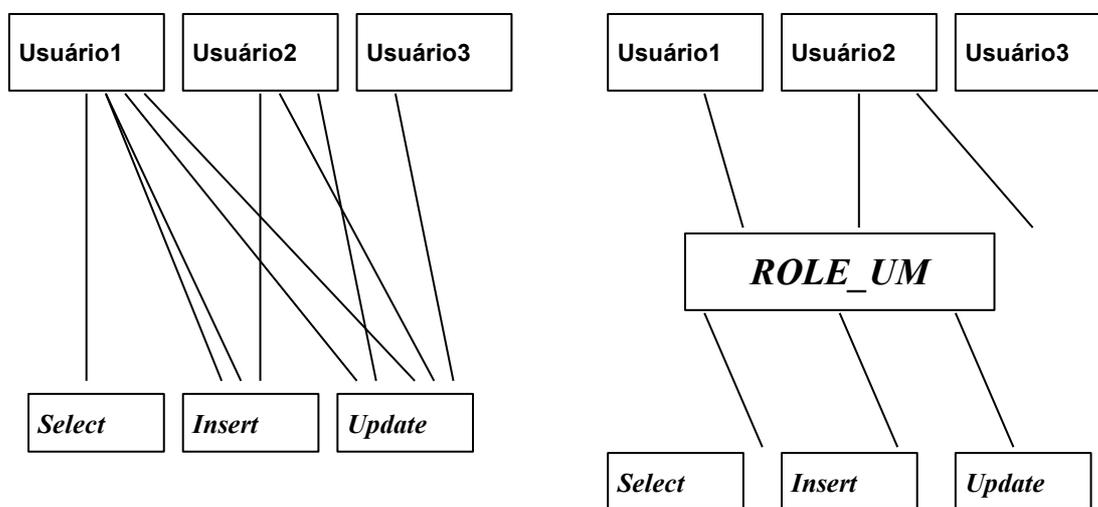
Visão ou Tabela Virtual(View) - É uma tabela formada por uma pesquisa em uma ou mais tabela-base. Ela não existe, mas parece ao usuário como se existisse.

Muitas vezes o acesso aos dados exigem “queries” complexas. Um usuário pode construir uma visão que contenha uma certa complexidade e disponibilizá-la para usuários finais e programadores, que farão “queries” mais simples sobre esta Visão. Você estará então, retirando a complexidade da navegação de uma aplicação ou de um usuário no acesso aos dados.

Usuário - São as pessoas que irão acessar o BD Oracle. Possui nome e senha para o acesso ("login").

Cabe ao DBA criar, modificar e excluir os usuários do BD.

Role - Agrupamento de privilégios, ou seja, em uma role podemos agrupar diversos privilégios e conceder aos usuários, não mais os privilégios e sim as roles.



Rotinas armazenadas (Stored Procedures, Functions e Triggers) - São grupos de comandos SQL que poderão ser ativados pelos usuários. Uma "Function" retorna um valor sempre, enquanto que as "Procedures" não.

As Triggers (gatilhos) são também rotinas armazenadas que são executadas (disparadas) quando ocorre algum evento modificando alguma tabela (Ex. Ao clicar o botão ...)

Índices - Quando criamos índices para uma tabela, especificando uma coluna, tal tabela é classificada de tal forma que, sempre que for mencionada uma query, o sistema usará o índice para Ter acesso direto aos dados desejados, ao invés de vasculhar a coluna toda. O objetivo da criação de índices é agilizar a recuperação de dados, ou seja, acelerar a procura dos dados na tabela.

Tablespace - Objeto lógico que guarda os arquivos de dados do BD Oracle. Convém, antes de definir o tamanho da Tablespace, fazer uma estimativa do quanto de espaço em disco será necessário para alocar os objetos (tabelas, índices, ...). Entretanto, podemos aumentar o tamanho da Tablespace, adicionando datafiles (arquivos de dados) à mesma.

PROCESSANDO COMANDOS SQL

Observando a figura anterior, podemos ver a presença de processos de usuários (users) e servidores (servers).

Todos os comandos SQL são executados pelos processos servidores, dividindo-os em 3 (três) fases:

Parse

Execute

Fetch.

Parse

Checa a sintaxe do comando;

Checa também os privilégios do usuário, definindo logo após, o caminho da pesquisa;

Define o plano de execução do comando.

Execute

Analisa o requerimento que está no buffer (na “Shared SQL Area”), e

Faz a leitura física ou lógica (no arquivo datafile ou no Database Buffer Cache).

Obs.:

Na “Shared SQL Area” temos: o texto do comando SQL (ou PL/SQL), o plano de execução.

Fetch

Retorna o resultado do comando SQL (geralmente chamada de “tabela-resultado”).

Obs.:

Na Shared SQL Area temos dois caches a saber:

a)Cache de biblioteca – armazena instruções SQL (e/ou procedimentos) bem como planos de execução das instruções (ex.: vários usuários (clientes) podem estar executando uma mesma aplicação compilada).

b)Cache de dicionário – Armazena informações contidas no dicionário de dados do BD, reduzindo as E/S de disco (o Oracle usa as informações do dicionário do BD em praticamente todas as operações).

DATABASE BUFFER CACHE

No Database Buffer Cache dados do BD Oracle são armazenados. O processo DBWR (database writer) grava os blocos de dados que as transações modificam na memória (database buffer cache) de volta para os arquivos de dados do BD.

Os buffer do Database Buffer Cache são compartilhados por todos os processos de usuário do Oracle conectados à instância (instance). O tamanho do bloco do database buffer cache é determinado no parâmetro DB_BLOCK_SIZE. A quantidade de blocos é definida no parâmetro DB_BLOCK_BUFFERS (parâmetros contidos no arquivo de parâmetros do Oracle).

Com a atuação do Database Buffer Cache as E/S de disco são reduzidas, armazenando dados que as transações solicitaram recentemente.

Exemplo:

Quando uma transação inclui uma instrução pra atualizar uma linha o Oracle manda para o database buffer cache o bloco que contém a linha, atualizando a linha logo em seguida. Se outra transação quiser atualizar a mesma linha, ou outra que está no mesmo bloco, não será necessário outro acesso a disco, pois a linha já está no database buffer cache (memória).

SEGMENTOS DE RECONSTRUÇÃO: O “DESFAZER” DO ORACLE

Um Segmento de Rollback do Oracle é uma área utilizada para guardar temporariamente dados anteriores a uma transação (que atualizou ou eliminou os dados). Se for necessário, através do comando Rollback (comando SQL) os dados são reconstruídos (que estavam no segmento de rollback) voltando os dados ao estado anterior.

O Segmento de Rollback System

Todo banco de dados Oracle tem um segmento de rollback denominado System. Durante a criação do BD, o segmento inicial é criado na tablespace System.

Obs.:

Ao se criar uma nova Tablespace devemos criar um Segundo segmento de Rollback denominado System.

Para se criar um segmento de rollback usa-se o comando CREATE ROLLBACK SEGMENT

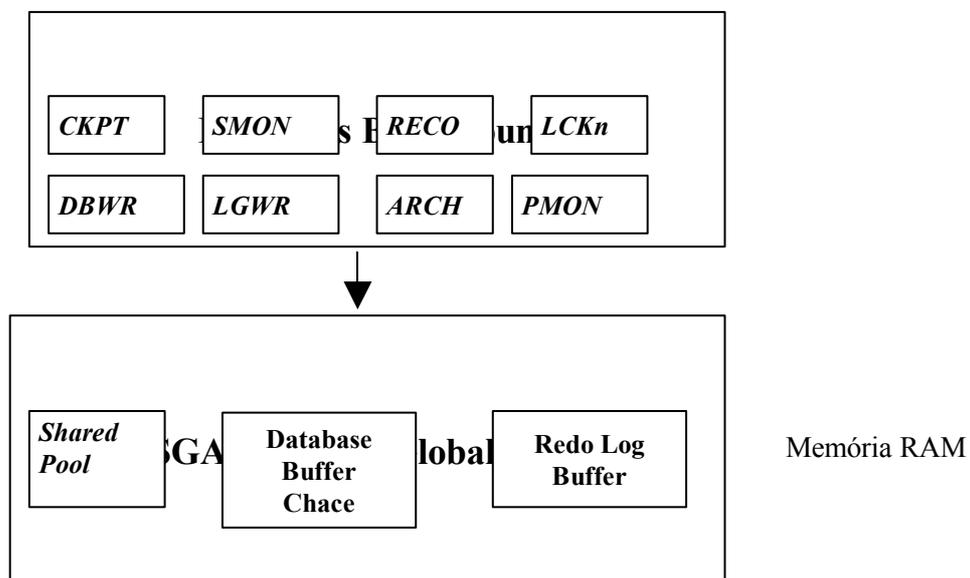
Sintaxe:

```
CREATE PUBLIC ROLLBACK SEGMENT rbs01
TABLESPACE st_01
STORAGE (INITIAL 10K NEXT 10K
          OPTIMAL 20K MAXEXTENTS 121);
```

Para colocar o segmento ativo (ONLINE):

```
ALTER ROLLBACK SEGMENT rbs01 ONLINE;
```

OS PROCESSOS ORACLE



Os processos Oracle executam tarefas específicas para os usuários do banco de dados. Tais processos trabalham no sentido de manipular as solicitações enviadas pelos processos de clientes e operam as diversas funções de um servidor de banco de dados Oracle. O Oracle distingue os processos de servidor de primeiro plano dos de segundo plano. Os processos de primeiro plano manipulam diretamente as solicitações dos processos de clientes, enquanto os processos de servidor de segundo plano manipulam outros serviços do Oracle.

Funcionamento de um sistema Cliente/Servidor Oracle:

- 1) Processo cliente envia solicitação para o servidor Oracle. Um processo despacha (Dispatcher) o mesmo para uma fila de solicitações.
- 2) Um processo de servidor de primeiro plano pega a solicitação mais antiga, processando-a. Ao terminar o processamento, coloca o resultado numa fila de respostas.
- 3) O Dispatcher gerencia a fila de respostas, enviando o resultado de cada solicitação para o processo cliente que fez a solicitação.

A quantidade de processos de servidor de primeiro plano é ajustada automaticamente pelo Oracle. De acordo com a necessidade o Oracle inicia/encerra processos de servidor de primeiro plano, visando atender as solicitações dos clientes.

PRINCIPAIS PROCESSOS DE SEGUNDO PLANO (BACKGROUND PROCESS)

DBWR (Database Writer) – Grava os dados do Database Buffer Cache para os arquivos físicos do Banco de Dados (datafiles). O DBWR não faz esta gravação imediatamente após a efetivação das transações. Tal evento acontece quando:

um processo servidor precisa ler um novo bloco para a memória e não existe espaço disponível, ou ele está ocioso por alguns segundos, ou Quando o Oracle efetua um Checkpoint do Banco de Dados ou da Tablespace.

LGWR (Log Writer) – Grava informações de alterações efetuadas sobre o BD do redo log buffer para os redo log files.

ARCH (Archiver) – É opcional. Copia automaticamente grupos de log de transações (redo log files) para um outro local físico (gerando os archives redo log files). Pode ser para fita ou para disco.

LCKn (Lock) - Faz bloqueios (Lock's) entre instâncias em servidores paralelos.

PMON (Process Monitor) – Realiza todo o processo de recuperação para um processo de usuário que tenha falhado. Por exemplo, se cair a conexão de um usuário com o Oracle, o Process Monitor irá desfazer a transação que caiu, liberando os lock's da transação, liberando assim o acesso de outras transações aos mesmos dados.

SMON (System Monitor) – Executa diversas operações de gerenciamento do espaço à medida que as transações liberam mais espaço para dados do BD.

RECO (Recoverer) – Recupera falhas que envolvam transações distribuídas.

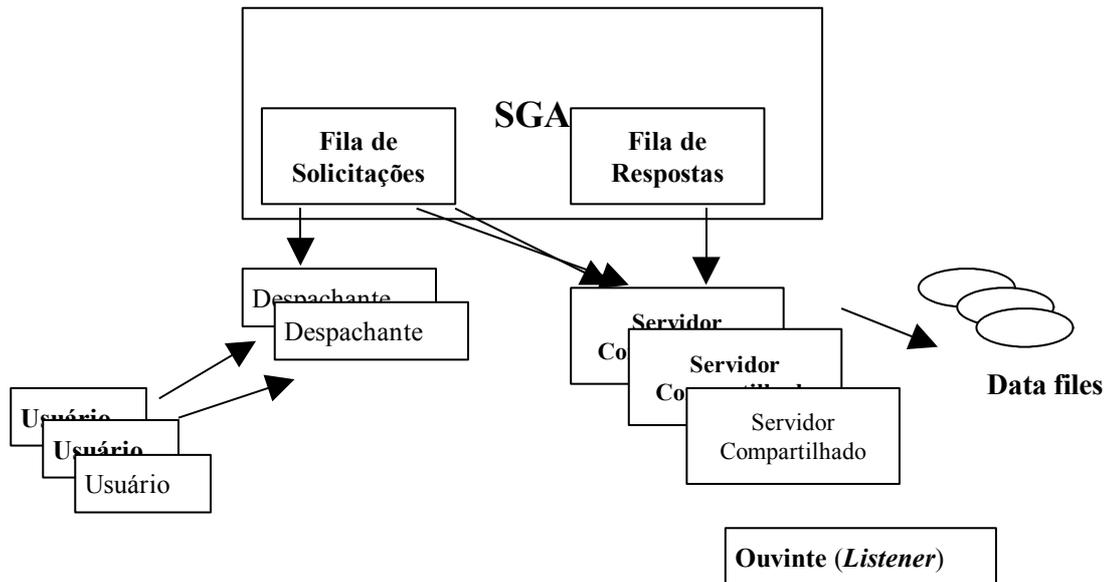
CKPT (Checkpoint) – Faz periodicamente uma verificação (checkpoint), para se certificar de que todas as informações modificadas na memória estão armazenadas fisicamente no disco. Tem a função de marcar o quanto o log de transações precisará aplicar, caso o sistema venha a exigir uma recuperação por queda. Somente as mudanças provocadas pelas transações a partir do último Checkpoint serão aplicadas pois o Oracle sabe que os dados foram gravados no disco (anteriores ao último Checkpoint). Atualiza os “headers” de todos os arquivos físicos do BD.

Dnnn (Dispatcher) – Direciona para a fila de solicitações, as solicitações dos processos cliente, e também, retorna para os clientes as respostas que estão na fila de respostas. Um número limitado de processos servidores são compartilhados pelos usuários (configuração multi-threaded).]

CONFIGURAÇÕES DE SERVIDORES DE BANCO DE DADOS

Em relação à configuração de servidores o Oracle pode trabalhar de 2 (dois) modos:
Servidor multilinear (Multi-threaded Server);
Servidor dedicado (Dedicated Server).

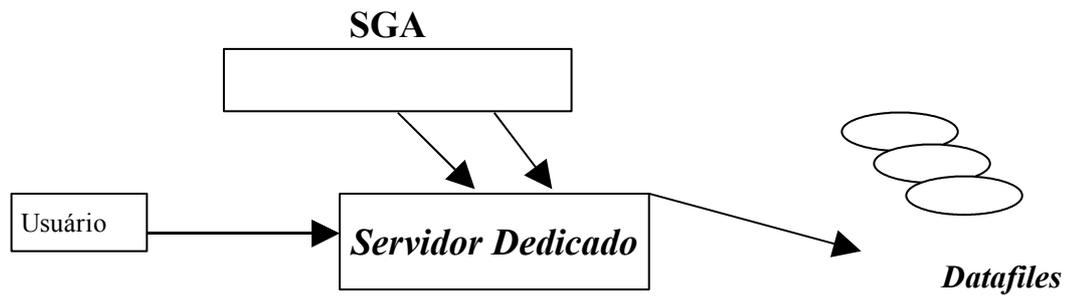
SERVIDOR MULTILINEAR (MULTI-THREADED)



Cada estação cliente usa um processo Cliente. O Oracle usa a arquitetura de servidor multilinear – que envolve os processos despachantes (Dispatchers), ouvinte (listener) e servidor (shared server) – para atender às solicitações dos diversos clientes. Duas filas são formadas: fila de solicitações (Request Queue) e fila de respostas (Response Queue).

Quando uma aplicação cliente estabelece uma conexão com o Oracle, o listener fornece o endereço de rede de um processo Dispatcher, com o qual o cliente se conecta. O Dispatcher então pega a solicitação do cliente e coloca-a na fila de solicitações. As solicitações são processadas e os resultados são inseridos na fila de respostas. A partir daí o Dispatcher retorna os resultados para as aplicações clientes adequadas.

SERVIDOR DEDICADO



Quando uma aplicação cliente se conecta ao Oracle, o processo ouvinte inicializa um processo servidor dedicado e conecta o cliente diretamente ao servidor dedicado, ignorando a arquitetura de servidor multilinear. Como um servidor dedicado consome uma quantidade significativa de memória do servidor de banco de dados para apenas um usuário, os servidores dedicados não deve ser usados para as conexões de aplicações clientes, a menos que absolutamente necessário.

LOG DE TRANSAÇÕES (LOG DE RECONSTRUÇÃO: REDO)

O Log de transações do Oracle registra imediatamente as mudanças feitas no Banco de Dados pelas transações em andamento, para que, se for necessário (em virtude, por exemplo, de uma falha), todo o trabalho confirmado seja protegido e recuperado.

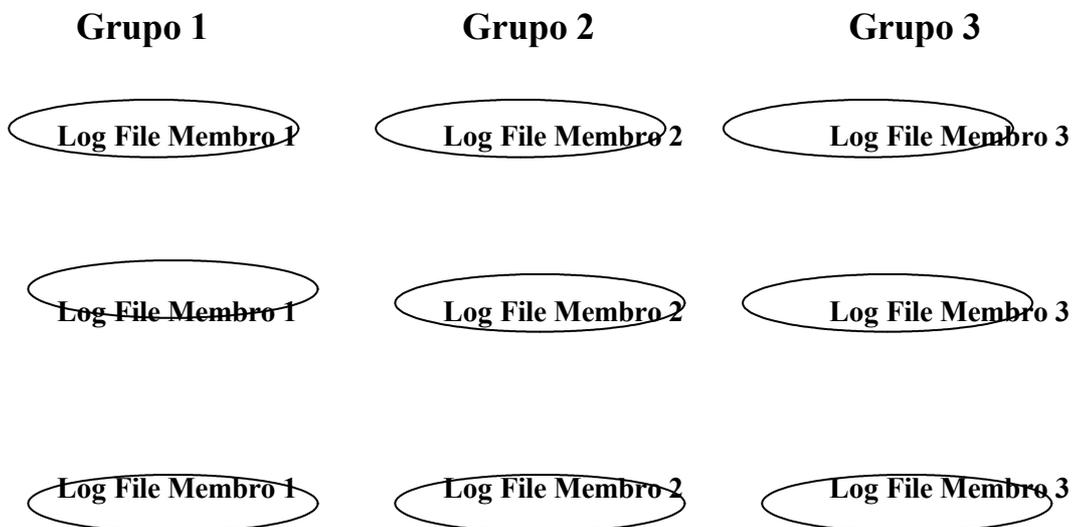
Na SGA existe o Redo Log Buffer que armazena as informações que serão gravadas nos Redo log Files.

REDO LOG FILES

Redo Log Files



Redo Log Files Espelhados



Os Redo Log Files armazenam todas as alterações no Banco de Dados, e são usadas em um processo de recuperação. Se os Redo Log Files são espelhados em vários discos, as mesmas informações são gravadas nos arquivos espelhados.

Os Redo Log Files são gravados de maneira circular, conforme figura acima.

Com o espelhamento dos Redo Log Files pelos discos, todos os grupos de membros são gravados simultaneamente. A vantagem do espelhamento é que se, por exemplo, algum problema em um disco, tem-se a cópia em outro.

LOG SWITCHES

Um Log Switch (troca de log) ocorre quando o Oracle troca de um redo log para outro. Enquanto isto o servidor fica gravando novas transações em outro grupo de log.

CHECKPOINTS

Processo que verifica se as últimas transações que estão na memória já foram gravadas fisicamente em disco.

Ocorre quando:

Acontece um Log Switch;

- Durante um intervalo de tempo (definido no parâmetro Log_Checkpoint_Timeout do arquivo de parâmetro);

Acontece um Shutdown;

O DBA força um Checkpoint; e

Quando a Tablespace é passada para Offline.

O processo Checkpoint é habilitado através do parâmetro Checkpoint_Process (init<instância>.ora).

O ARQUIVO DE PARÂMETROS

No arquivo de parâmetros é que definimos os parâmetros de inicialização da instância Oracle.

Exemplo (sistema Unix):

```
# Parameter file para o database pb1 (Exemplo)
#
#
# Nome do Banco de Dados

db_name = pb1

# Compatibilidade com as características da versão 7.3.2

compatible = 7.3.2

# Localização dos controlfiles

control_files = (/prd3/oracle/pb1/dbs/control01.ctl,
                /prd4/oracle/pb1/dbs/control02.ctl,
                /prd5/oracle/pb1/dbs/control03.ctl)

# Especificação dos Segmentos de Reconstrução (Rollback Segments)

rollback_segments = (r01,r02,r03,r04)

# Definição do tamanho da SGA do Banco
#
# SGA = [( db_block_size * db_block_buffers) + shared_pool_size + log_buffers ]

db_bloc_size = 2048          # 2k

shared_pool_size = 9000000   # 3500000 → BD Pequeno
                             # 6000000 → BD Médio
                             # 9000000 → BD Grande
                             # 18000000 → BD com Designer/2000

db_block_buffers = 3200      # 200 → BD Pequeno
                             # 550 → BD Médio
                             # 3200 → BD Grande

log_buffer = 163840          # 8192 → BD Pequeno
                             # 32768 → BD Médio
                             # 163840 → BD Grande

# SGA = [ ( 2048 * 3200 ) + 9000000 = 163840 ] → 15 Mbytes (Aproximadamente)
```

```

# Parametros de ajuste dos redologs files

log_files = 3          # Num. De redo logs que podem ser abertos
                      # limitado pelo parametro maxlogfiles do create database
                      # que indica o num. Máximo de grupos de redolog files

log_checkpoint_interval = 10000# opcao por desempenho. Nunca um checkpoint será
                              # disparado por Ter atingido este limite (10 mil blocos
                              # de 512 gravados, desde o último checkpoint).

log_checkpoint_timeout = 0    # valor default (em segundos).

log_archive_start = true     # utilizacao de archiving automatico

# Localização dos arquivos de archived redologs, trace e log.
# Trace (dump) dos processos background, de usuários

log_archive_dest = /prd5/oracle/pb1/arch/A
log_archive_format = "RCH%s.log"

background_dump_dest = /prd2/oracle/pb1/bdump
core_dump_dest = /prd2/oracle/pb1/cdump
user_dump_dest = /prd2/oracle/pb1/udump

max_dump_file_size = 10240

# Num. de processos = proc. Servers + proc Backgs

processes = 50           # 50 → BD Pequeno
                        # 100 → BD Medio
                        # 200 → BD Grande

# Numero de blocos lidos para memoria numa operacao de I/O

db_file_multiblock_read_count = 32    # 8 → BD Pequeno
                                       # 16 → BD Medio
                                       # 32 → BD Grande

# Parametros de configuracao do Asynchronous I/O

use_async_io = true
lgwr_use_async_io = true
use_post_wait_driver = true

# Parametrso de configura;ão do servidor multi-threaded
# Multi_threaded server requer a instala;ão do SQL*Net V2

mts_dispatchers          = "tcp,2"
mts_max_dispatchers     = 10
mts_servers             = 2
mts_max_servers         = 10
mts_service             = pb1

```

mts_listener_address = "(ADDRESS=(PROTOCOL=tcp) (HOST=durjbf01) (PORT=1521))"

EXERCÍCIO I

1) Responda abaixo:

- a) Quais os arquivos físicos que compõem um banco de dados Oracle?
- b) Como é composta uma instância Oracle?
- c) Cite, no mínimo, 04 (quatro) atribuições/responsabilidades de um DBA
- d) Localize e exiba na tela o arquivo de parâmetros do Oracle (instância default → ORCL)
- e) Quais os processos de segundo plano (background processes) Oracle? Para que serve cada um deles?
- f) A SQL é a linguagem padrão de BD. Ela pode ser usada de dois modos. Quais são esses modos? Fale sobre eles.
- g) Monte uma "Query" (comando SELECT) qualquer.
- h) Qual o principal objetivo dos Bancos de Dados Distribuídos?
- i) Fale sobre o conceito Data Warehouse
- j) Disserte sobre 05 (cinco) objetos Oracle

INICIALIZANDO E TERMINANDO UM BD ORACLE

Usamos o comando STARTUP para iniciar um BD e o SHUTDOWN para terminá-lo. Antes de inicializar ou terminar um banco Oracle é necessário conectar-se como usuário INTERNAL.

Obs.: Conectar-se como INTERNAL é equivalente a conectar-se como usuário SYS (O "dono" do dicionário de dados do Oracle)

INICIALIZANDO (STARTUP)

Um processo de inicialização é composto de 3 passos: START, MOUNT e OPEN.

START – O Oracle lê o arquivo de parâmetros; aloca espaço na memória para a SGA; "starta" os processos de segundo plano (background process) e abre os arquivos de trace (background trace files).

MOUNT – Abre o(s) Control(s) File(s) e lê os nomes do Banco de dados e dos REDO LOG FILES.

OPEN – Abre o Banco de Dados disponibilizando-o para as transações dos usuários.

Sintaxe:

```
STARTUP OPEN/MOUNT/NOMOUNT database
EXCLUSIVE/PARALLEL
PFILE = parfile
RECOVER/FORCE/RESTRICT
```

Onde:

OPEN → Disponibiliza o Banco para o acesso aos usuários.

MOUNT → Monta o banco para atividades de DBA, mas não deixa livre o acesso ao BD.

NOMOUNT → Cria a SGA e "starta" os processos de segundo plano, mas não deixa livre o acesso ao BD.

EXCLUSIVE → Permite que somente a instância corrente seja acessada.
 PARALLEL → Disponibiliza múltiplas instâncias.
 PFILE=parfile → Pega do arquivo de parâmetro especificado (parfile) e “starta” tal instância.
 FORCE → Aborta a instância que está “rodando” para “startar” a nova.
 RESTRICT → Permite que só os usuários com privilégio RESTRICT SESSION acesse o BD.
 RECOVER → Inicia processo de Recover quando o banco é Startado.

Ex:

A) SVRMGR>STARTUP teste;

Inicializa o banco de dados teste.

B) STARTUP EXCLUSIVE MOUNT teste PFILE inittest.ora;

Inicializa o BD teste, apenas montando-o, utilizando o arquivo de parâmetros inittest.ora

Obs.:

Com o comando ALTER DATABASE podemos alterar o modo de banco, ou seja, podemos por exemplo apenas montá-lo, deixando em modo Exclusivo (para serem feitas atividades de DBA. O banco não fica disponível aos usuários).

Obs: Nos Bancos Apartir da versão 9 o comando para inicializar ou terminar o banco é

SQLPLUS /NOLOG

Connect / as sysdba

TERMINANDO UM BD ORACLE (SHUTDOWN)

O comando SHUTDOWN fecha o BD e indisponibiliza-o aos usuários.

Sintaxe:

SHUTDOWN ABORT/IMMEDIATE/NORMAL;

Onde:

ABORT → Não espera a saída (desconexão) dos usuários, terminando rapidamente.

IMMEDIATE → Desconect os usuários e depois termina.

NORMAL → espera todos os usuários conectados saíre, para depois terminar o BD.

ALTER DATABASE: ALTERANDO O ESTADO DO BANCO DE DADOS

Com o ALTER DATABASE podemos fazer trocas no estado do Banco, para disponibilizá-lo a todos os usuários.

Sintaxe:

ALTER DATABASE database

MOUNT EXCLUSIVE/PARALLEL
OPEN

Onde:

Database	Nome do BD a ser alterado.
MOUNT	Monta o BD para atividades do DBA, mas não disponibiliza-o aos usuários.
OPEN	Permite o acesso dos usuários ao BD.
EXCLUSIVE	Permite que somente a instância corrente acesse o BD.
PARALLEL	Permite que várias instâncias acessem o BD.

Ex.:

```
SVRMGR>CONNECT INTERNAL  
SVRMGR> ALTER DATABASE OPEN;
```

Libera aos usuários o acesso ao BD.

ALTER SYSTEM: RESTRINGINDO O ACESSO

Tem a capacidade de disponibilizar o BD para somente os usuários que tenham o privilégio de RESTRICTED SESSION (só eles se conectam ao BD).

Sintaxe:

```
ALTER SYSTEM  
ENABLE/DISABLE RESTRICT SESSION;
```

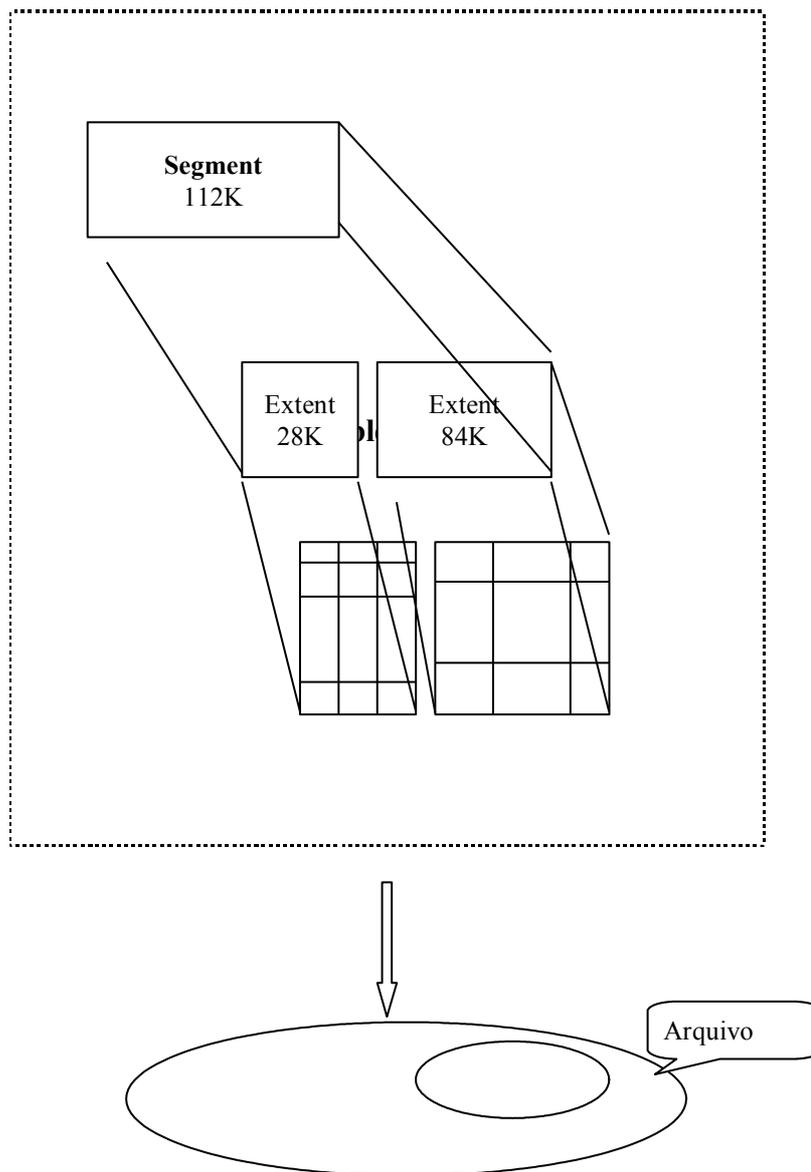
Onde:

ENABLE	Permite que só os usuários que tem privilégio RESTRICT SESSION acessem o BD.
DISABLE	Permite acesso ao BD a todos os usuários.

Ex:

```
ALTER SYSTEM ENABLE RESTRICTED SESSION;
```

GERENCIANDO O ESPAÇO DO BANCO DE DADOS



Database

As seguintes estruturas são utilizadas para o gerenciamento do espaço do BD.

Estrutura	Definição
BLOCO	São blocos que ficam alocados nos datafiles
EXTENT	Compõem o bloco do datafile.
SEGMENT	Armazena um ou mais EXTENT's contendo todos os dados na TABLESPACE.
TABLESPACE	Estrutura lógica do Oracle que pode ter um ou mais datafiles (arquivos físicos).
DATAFILES	Arquivos físicos que armazenam os dados do BD.
DATABASE	Coleção lógica dos dados compartilhados (que estão armazenados nos TABLESPACES)

COMPONENTES DE UM BLOCO ORACLE

Um bloco é composto das seguintes partes: Cabeçalho (Header), Diretório da Tabela (Table Directory), Diretório da Linha (Row Directory), Espaço Livre (Free Space), Linha de Dados (Row Data).

Parte	Descrição
Cabeçalho(Header)	Contém informações gerais do bloco: endereço do bloco, tipo de segmento.
Diretório da Tabela (Table Directory)	Armazena informações sobre as tabelas.
Diretório da Linha (Row Directory)	Contém informações sobre a linha que está no bloco.
Espaço Livre (Free Space)	É o conjunto de bytes do bloco disponível para operações de manipulação (insert, update, ...)
Linha de Dados (Row Data)	Guarda dados da tabela (ou de índices).

Obs.: O espaço livre do bloco também pode conter transações.

CONTROLANDO O ESPAÇO LIVRE

O controle do espaço livre para comandos de manipulação no bloco é especificado nos parâmetros PCTFREE, PCTUSED, INITRANS, MAXTRANS.

PCTFREE -	Armazena o percentual de espaço livre no bloco.
PCTUSED -	Permite ao bloco reconsiderar o espaço livre para a inserção de novas linhas.
INITRANS -	Define o número inicial de transações para o bloco (default 1, mínimo 1, máximo 255)
MAXTRANS	Número máximo de transações concorrentes para o bloco (default 255, mínimo 1, máximo 255).

ESTUDANDO AS TABLESPACES

As TABLESPACES são objetos lógicos que armazenam outros objetos do BD Oracle.

Cada TABLESPACE possui um ou mais DATAFILES.

Podemos colocar uma TABLESPACE, no estado de OFFLINE (desativar a mesma) com exceção da TABLESPACE SYSTEM (possui o dicionário de dados do Oracle).

A TABLESPACE SYSTEM contém informações do dicionário de dados do Oracle, bem como contém também o Segmento de Reconstrução (ROLLBACK SEGMENT) denominado SYSTEM.

Obs: a TABLESPACE SYSTEM pode conter dados de usuários.

O uso de outras TABLESPACE diferente da SYSTEM permite uma maior flexibilidade na administração, por parte do DBA

Tais TABLESPACES armazenam Segmentos de ROLLBACK, segmentos temporários, dados de aplicações, índices, espaço para os usuários.

CRIAÇÃO DE TABLESPACES

Sintaxe:

```
CREATE TABLESPACE nome_da_tablespace
DATAFILE "especif_do_datafile"
DEFAULT STORAGE cláusulas
ONLINE/OFFLINE
```

Onde:

Nome_da_tablespace	É o nome da TABLESPACE a ser criada.
"especif_do_datafile"	Especificação do arquivo. Contém:
nome do datafile	É o nome do datafile ou redo log file.
SIZE inteiro	É o tamanho do mesmo (em K ou M)
REUSE	Permite ao Oracle reusar o arquivo.
DEFAULT STORAGE cláusulas	Especifica os parâmetros default de armazenamento para todos os objetos criados na TABLESPACE.
ONLINE/OFFLINE	Disponibiliza/indisponibiliza a TABLESPACE.

Cláusulas (DEFAULT STORAGE):

INITIAL inteiro K/M	Tamanho em bytes do primeiro EXTENT.
NEXT inteiro K/M	Tamanho em bytes do próximo EXTENT a ser alocado para a TABLESPACE.
MINEXTENTS	Total de EXTENTS alocados quando o objeto for criado.
MAXEXTENTS	Total de EXTENTS, incluindo o primeiro, que o Oracle pode alocar para o objeto.
PCTINCREASE	Percentual de incremento que o Oracle aloca novos EXTENTS para uma Tabela ou Índice.

ALTERANDO UMA TABLESPACE

Comando: ALTER TABLESPACE

Sintaxe:

```
ALTER TABLESPACE nome_da_tablespace
ADD DATAFILE especific_do_datafile
RENAME DATAFILE 'nome_datafile' TO 'nome_datafile'
DEFAULT STORAGE Cláusulas
ONLINE/OFFLINE
NORMAL/TEMPORARY/IMMEDIATE
BEGIN/END BACKUP
```

Onde:

nome_da_tablespace	É o nome da TABLESPACE a ser alterada.
--------------------	--

ADD DATAFILE	Adiciona DATAFILE
RENAME DATAFILE	Renomeia um ou mais DATAFILES da TABLESPACE.
DEFAULT STORAGE	Altera as Cláusulas
ONLINE/OFFLINE	Disponibiliza/indisponibiliza a TABLESPACE.
NORMAL	Faz um CHECKPOINT para todos os DATAFILES da TABLESPACE.
TEMPORARY	Faz um CHECKPOINT para todos os DATAFILES da TABLESPACE, mas não assegura que serão gravados dados em todos eles.
IMMEDIATE	Não assegura que os datafiles serão disponibilizados e não faz o CHECKPOINT.

DELETANDO UMA TABLESPACE

Comando: DROP TABLESPACE

Sintaxe:

```
DROP TABLESPACE nome_da_tablespace
INCLUDING CONTENTS [CASCADE CONSTRAINTS]
```

Onde:

Nome_da_tablespace	É o nome da TABLESPACE a ser apagada.
INCLUDING CONTENTS	Apaga todo o conteúdo da TABLESPACE.
CASCADE CONSTRAINTS	Apaga todas as restrições de integridade referencial das tabelas fora da TABLESPACE

CRIANDO SEGMENTOS DE RECONSTRUÇÃO (ROLLBACK SEGMENTS)

Comando: CREATE ROLLBACK SEGMENT

Sintaxe:

```
CREATE ROLLBACK SEGMENT nome_segmento
TABLESPACE nome_da_tablespace
STORAGE cláusulas
```

Onde:

nome_segmento	É o nome do segmento a ser criado.
Nome_da_tablespace	TABLESPACE onde o segmento será criado.
STORAGE cláusulas	Define como o espaço será alocado para o segmento.

Ex:

```
CREATE ROLLBACK SEGMENT rbs1
TABLESPACE SYSTEM
STORAGE (INITIAL 100K, NEXT 100K,
MINEXTENTS 2, MAXEXTENTS 121,
```

OPTIMAL 1000K);

CRIANDO UM BANCO DE DADOS

Comando: CREATE DATABASE

Sintaxe:

```
CREATE DATABASE nome_do_banco
CONTROL FILE REUSE
LOGFILE especific_arquivo GROUP inteiro
MAXLOGFILES inteiro
MAXLOGMEMBERS inteiro
MAXLOGHISTORY inteiro
DATAFILE especific_arquivo
MAXDATAFILES inteiro
MAXINSTANCES inteiro
    ARCHIVELOG/NOARCHIVELOG
EXCLUSIVE
CHARACTER SET charset
```

Onde:

Nome_do_banco Nome do banco de dados a ser criado

Especific_arquivo

 Especificação do arquivo. Contém:

nome do datafile	É o nome do datafile ou redo log file.
SIZE inteiro	É o tamanho do mesmo (em K ou M)
REUSE	Permite ao Oracle reusar o arquivo.

CONTROL FILE REUSE	Especifica o control file especificado no arquivo de parâmetros (podendo ser reusado).
LOGFILE GROUP	Especifica o nome dos logfiles.
MAXLOGFILES	máximo de log files que podem ser criados para o banco de dados.
MAXLOGMEMBERS	máximo de membros de log files para um grupo de log files.
MAXLOGHISTORY	máximo de archived redo logs para ser recuperado automaticamente (Oracle Parallel Server).
DATAFILE especific_arquivo	Nome(s) do(s) datafile(s) a ser(em) usado(s).
MAXDATAFILES	Máximo de datafiles que podem ser criados para o banco.
MAXINSTANCES	Máximo de instâncias que podem ser montadas e abertas simultaneamente pelo banco de dados.
ARCHIVELOG	Habilita o modo de arquivamento.
NOARCHIVELOG	Não habilita o modo de arquivamento.
EXCLUSIVE	monta o banco no modo exclusivo após a sua criação.
CHARACTER SET	Seta o tipo de caracteres que será usado pelo banco para armazenar os dados (Ex. ASCII).

ALTERANDO UM BANCO DE DADOS

Comando: ALTER DATABASE

Sintaxe:

```
ALTER DATABASE nome_do_banco
ADD LOGFILE MEMBER 'nome_arquivo'
REUSE TO GROUP inteiro ("nome_arquivo, nome_arquivo)
```

Onde:

nome_do_banco	Nome do banco de dados a ser criado
nome_arquivo	nome de arquivo a ser adicionado (colocar o caminho igual dos que já existem).
GROUP inteiro	número do grupo que receberá o log file membro a ser adicionado.

Obs.:

A Visão V\$LOGFILE mostra os grupos e membros de log files.

INSTALAÇÃO DO ORACLE

Na instalação do Oracle geralmente utilizamos o instalador da Oracle, que é o Oracle Installer. Tal produto tanto instala como desinstala os produtos Oracle. É de fácil utilização pois basta escolhermos os produtos Oracle contratados e selecionar a opção Install. (para maiores detalhes ver Manual de Instalação do produto).

Obs Importante:

Alguns procedimentos (scripts) devem ser executados após a instalação do Oracle Server (como usuário SYS)

- a) CATALOG.SQL Cria visões do dicionário de dados.
- b) CATEXP.SQL Cria visões do dicionário de dados para o uso do Export/Import.
- c) CATPROC.SQL Cria tabelas e visões adicionais do dicionário de dados para a opção procedural do Oracle.

Tais procedimentos podem demandar de muito tempo, pois formam muitas estruturas de dados.

O ARQUIVO DE PARÂMETROS: ESTUDANDO OS PARÂMETROS

O arquivo de parâmetros Oracle possui diversos parâmetros que devem ser conhecidos pelo DBA:

Db_name	nome do banco de dados (obrigatório)
Control_files	nome/localização dos control files
rollback_segments	nome dos segmentos de rollback alocados para a instância.
db_block_size	tamanho do bloco Oracle (em bytes)

shared_pool_size	parâmetro que define a shared pool (para comando SQL compartilhados)
db_block_buffers	número de blocos armazenados na SGA
log_buffer	número de bytes alocados para o redo log buffer na SGA
log_files	número de redo log's que podem ser abertos.
Log_checkpoint_interval	define a frequência dos checkpoints
Max_dump_file_size	tamanho máximo de blocos para os arquivos de trace.
Log_archive_start	Habilita ou desabilita o modo de arquivamento (archivelog mode)
Log_archive_dest	localização dos archives redo log files
Log_archive_format	formato dos archives redo log files
User_dump_dest	caminho dos arquivos de trace dos usuários
Background_dump_dest	caminho onde serão gerados arquivos de trace dos processos background do Oracle
Processes	número máximo de processos (processos servidores + processos background)
Sql_trace	habilita/desabilita o trace do SQL para todas as sessões de usuário
Audit_trail	habilita/desabilita o Audit

Obs: Para ver os parâmetros usar o comando SHOW PARAMETER

Exemplo:

```
SVRMGR>SHOW PARAMETER DB;
```

```
SVRMGR>SHOW PARAMETER DEST;
```

CRIAÇÃO E GERENCIAMENTO DE USUÁRIOS DO BANCO DE DADOS

Cada usuário do BD Oracle possui um esquema associado a ele, ou seja, cada objeto criado para (ou pelo) usuário é do esquema do mesmo. Ao se criar um usuário Oracle, um esquema com o mesmo nome é criado para ele.

Comando: CREATE USER

Sintaxe:

```
CREATE USER nome_usuario
IDENTIFIED BY senha/EXTERNALLY
DEFAULT TABLESPACE nome_tablespace
TEMPORARY TABLESPACE nome_tablespace_temporaria
PROFILE arquivo_profile
QUOTA inteiro/UNLIMITED ON nome_tablespace
```

Onde:

Nome_usuario	nome do usuário a ser criado.
BY senha	senha do usuário
EXTERNALLY	usa como senha a senha do sistema operacional.
DEFAULT TABLESPACE	identifica a TABLESPACE default para os objetos do usuário.
TEMPORARY TABLESPACE	TABLESPACE temporária que armazena os segmentos temporários.
QUOTA inteiro	define o espaço da TABLESPACE para o usuário que está sendo criado (número inteiro em K ou M bytes).

UNLIMITED o espaço na TABLESPACE fica sem limite para o usuário.
 PROFILE arquivo_profile define o arquivo de profile do usuário.

Ex:

```
SVRMGR>CREATE USER ALUNO1
        IDENTIFIED BY aluno1
        DEFAULT TABLESPACE TS_Treina
        TEMPORARY TABLESPACE Temp
        QUOTA 5M ON TS_Treina
        PROFILE default;
```

Obs:

O Oracle possui dois usuários especiais: o SYS e o SYSTEM. O SYS é o usuário para colocar/tirar o BD do “ar” O usuário SYSTEM é o usuário administrador do BD.

ALTERANDO UM USUÁRIO DO BANCO DE DADOS ORACLE

Comando: ALTER USER

Sintaxe:

```
ALTER USER nome_usuario
IDENTIFIED BY senha/EXTERNALLY
DEFAULT TABLESPACE nome_tablespace
TEMPORARY TABLESPACE nome_tablespace_temporaria
QUOTA inteiro/UNLIMITED ON nome_tablespace
PROFILE arquivo_profile
DEFAULT ROLE nome_role/ALL [EXCEPT nome_role]/NONE
```

Onde:

Nome_usuario	nome do usuário a ser alterado.
BY senha	senha do usuário
EXTERNALLY	usa como senha a senha do sistema operacional.
DEFAULT TABLESPACE	identifica a TABLESPACE default para os objetos do usuário.
TEMPORARY TABLESPACE	TABLESPACE temporária que armazena os segmentos temporários.
QUOTA inteiro	define o espaço da TABLESPACE para o usuário que está sendo criado (número inteiro em K ou M bytes).
UNLIMITED	o espaço na TABLESPACE fica sem limite para o usuário.
PROFILE arquivo_profile	define o arquivo de profile do usuário.
DEFAULT ROLE nome_role	define a role (conj. de privilégios) default do usuário.

Ex:

```
SVRMGR> ALTER USER aluno1
        IDENTIFIED BY aluno1
        QUOTA 6M ON TS_Treina;
```

DELETANDO UM USUÁRIO

Comando: DROP USER

Sintaxe:

```
DROP USER nome_usuario CASCADE
```

Onde:

Nome_usuario nome do usuário a ser deletado.
CASCADE deleta todos os objetos do esquema do usuário que está sendo deletado, antes de deletar o usuário.

Ex:

```
SVRMGR>DROP USER aluno1 CASCADE;
```

Obs:

se o usuário está conectado ele não pode ser deletado antes do fim da conexão.

As Visões (VIEW's) USER_USERS, ALL_USERS, DBA_USERS, USER_TS_QUOTAS E DBA_TS_QUOTAS são visões do dicionário de dados com informações sobre os usuários.

PRIVILÉGIOS

Nem todos os usuários do Oracle podem acessar todas as tabelas do Banco de Dados. Para conceder ou revogar privilégios de acesso/manipulação aos usuários o DBA utiliza os comando GRANT e REVOKE.

Podemos conceder/revogar diversos tipos de privilégios, dependendo do usuário. Se um usuário final irá somente consultar Tabelas, Visões O DBA concederá somente privilégios de consulta (SELECT). Entretanto para um desenvolvedor o DBA deverá dar, além de privilégios de consulta, outros privilégios (por exemplo, privilégios de Criação/Alteração de tabelas, visões, etc.)

Um privilégio de sistema é um privilégio poderoso que, se concedido ao usuário, dará ao mesmo o direito de executar uma operação em nível de banco de dados. Cabe portanto ao DBA a tarefa de definir quais usuários do Banco terão tais privilégios (poderão ser outros administradores e/ou desenvolvedores de aplicações).

Exemplo:

Um usuário com privilégio SELECT ANY TABLE pode consultar qualquer tabela do banco.

Um usuário com o privilégio ALTER DATABASE pode alterar a estrutura física do banco, incluindo novos arquivos.

Um privilégio de objetos do BD administra uma operação do banco de dados sobre um objeto específico do BD (uma tabela, uma visão, ...).

Exemplo:

Um usuário com privilégio de SELECT na tabela SALÁRIOS, pode consultar a mesma.

Um outro usuário com privilégio de UPDATE, na tabela SALARIOS, poderá alterar a mesma (ação que não poderá ser realizada pelo usuário com privilégio de SELECT apenas).

COMANDO GRANT

Concede privilégios aos usuários Oracle cadastrados.

Sintaxe:

```
GRANT <privilégio> to { usuário | role | public }  
[ WITH ADMIN OPTION ];
```

Exemplos:

1) GRANT CREATE SESSION, CREATE TABLE TO JOAO;

2) GRANT SELECT ANY TABLE TO manael;

3) GRANT SELECT, INSERT, UPDATE, DELETE
ON T_Salario TO joaquim;

Obs: A opção WITH ADMIN OPTION dá ao usuário o “poder” de conceder privilégios (que ele possui) a outros usuários.

COMANDO REVOKE

O comando REVOKE retira (revoga) o privilégio concedido através do GRANT.

Sintaxe:

```
REVOKE {privilégio | role } FROM {usuário | role | PUBLIC };
```

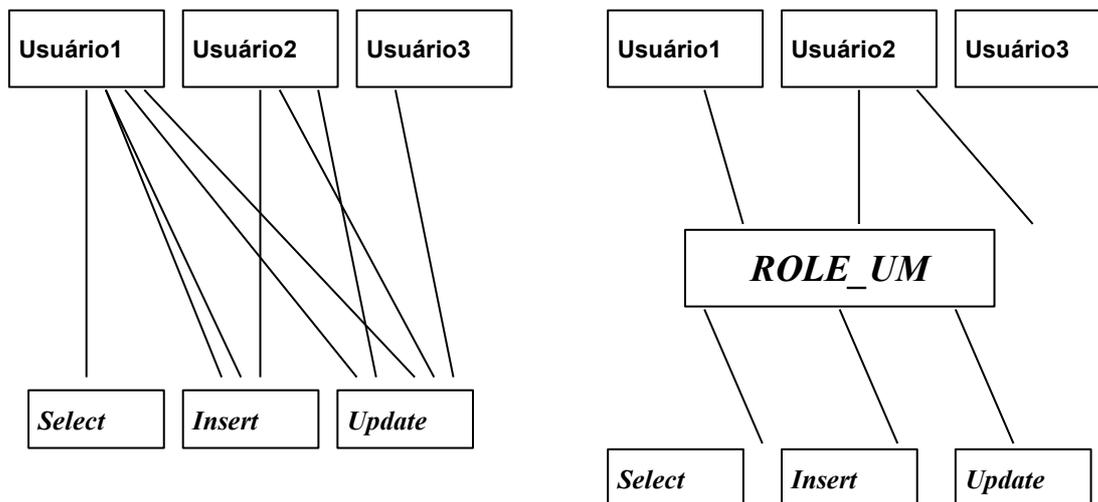
Exemplos:

1) REVOKE SELECT ANY TABLE FROM manael;

2) REVOKE UPDATE FROM joaquim;

AGRUPANDO PRIVILÉGIOS: “ROLE”

Tendo em vista que o Oracle possui uma infinidade de privilégios e que a quantidade e diversidade de perfis de usuários em um banco de dados também é grande, a administração dos privilégios de todos eles (usuários) seria uma tarefa no mínimo bastante trabalhosa. Visando dirimir este problema existe o objeto Oracle denominado ROLE. Em uma ROLE podemos agrupar diversos privilégios e conceder aos usuários, não mais os privilégios e sim as ROLES.



Exemplo:

- 1)CREATE ROLE admin_salarios;
- 2)GRANT SELECT, INSERT, UPDATE,DELETE
ON T_SALARIOS
TO admin_salarios;
- 3)GRANT SELECT,INSERT, UPDATE, DELETE
ON T_COMISSOES
TO admin_salarios;
- 4)GRANT admin_salarios to USUARIO_A, USUARIO_B, USUARIO_C, ...

No exemplo acima, os usuários A, B e C receberam os privilégios da ROLE admin_salarios (que Seleciona, Insere, Altera e Elimina dados das Tabelas T_SALARIOS e T_COMISSOES).

Obs.:

- O Comando REVOKE (revoga privilégios) também é utilizado com ROLES. As ROLES DBA, CONNECT e RESOURCE são ROLES padrão do Oracle. O DBA pode, por exemplo, revogar a ROLE RESOURCE de determinados usuários (concedendo aos mesmos somente alguns privilégios, de tal ROLE).

Exemplo:

- 1)REVOKE UPDATE ON T_COMISSOES FROM admin_salarios;

2)REVOKE USUARIO_B FROM admin_salarios;

VISÕES (VIEW'S) DO DICIONÁRIO DE DADOS

O dicionário de dados Oracle possui diversas visões que estão divididas em categorias:

Categoria	Descrição
DBA_XXX	Visões do DBA, com informações de qualquer definição de objeto do BD.
USER_XXX	Visões que podem ser acessadas por qualquer usuário, dando informações de seus objetos.
ALL_XXX	Visões que podem ser acessadas por qualquer usuário, com informações de todos os objetos que ele pode acessar.

Obs:

Todas as tabelas e visões do DD podem ser acessadas pelo usuário SYS.
Para ver as visões use os comandos DESCRIBE e SELECT.

Lista de algumas Visões:

V\$SESSION	Mostra as sessões logadas.
USER_USERS	Usuário conectado.
ALL_USERS	Todos os usuários conectados e outras colunas.
DBA_USERS	Todas as informações dos usuários.
USER_TS_QUOTAS	Limite que o usuário tem (Quota) na TABLESPACE.
DBA_TS_QUOTAS	Todas as quotas de todos os usuários.
DBA_DATAFILES	DATAFILES e TABLESPACES do BD.
DBA_TABLES	Tabelas do DBA
ALL_TABLES	Todas as tabelas.
DBA_ROLE_PRIVS	Tabela com os privilégios.
DBA_AUDIT_TRAIL	Conjunto de trilhas de auditoria.
DBA_EXP_FILES	Descrição dos arquivos exportados.
DBA_ROLES	As roles do BD.
DBA_SEGMENTS	Informações sobre os segmentos do BD.
DBA_TABLESPACE	Tablespaces do BD.
DBA_ROLLBACK_SEGS	Segmentos de Rollback.
V\$DATAFILE	Informações sobre os DATAFILES do BD.
V\$PARAMETER	Parâmetros de inicialização do BD.
V\$LOG_HISTORY	Informações sobre o Log de transações do BD.

BACKUP E RECOVERY DO BANCO DE DADOS

Métodos de Backup Oracle: Backup completo (full) ou parcial. Pode ser feito ONLINE ou OFFLINE.

BACKUP COMPLETO (“FULL”) OFFLINE

Copiar através de comando do sistema operacional os arquivos do BD Oracle: datafiles, redo log files, control files e parameter file.

Passos:

- a) Selecionar os arquivos relevantes para o Backup;
- b) Dar um “Shutdown” na instância Oracle,
- c) “Backuppear” (copiar) todos os datafiles, redo log files, control file(s) e parameter file(s) usando comandos do Sistema Operacional;
- d) Reinicializar a instância Oracle (Startup).

Ex:

```
SVRMGR>CONNECT INTERNAL
SVRMGR>SHUTDOWN NORMAL
SVRMGR>HOST CP <ARQUIVOS_ORACLE> <DIRETÓRIO_DE_BACKUP>
SVRMGR>STARTUP
```

Obs:

- Para saber quais os nomes dos arquivos a serem copiados consultar as visões: DBA_DATAFILES, V\$LOGFILE (coluna MEMBER) Usar o comando SHOW PARAMETER CONTROL_FILES para ver o nome dos Control Files.

RESTAURANDO (RECOVERY) DO BD

Passos:

- a) Tirar o BD do “ar” (executar um SHUTDOWN ABORT, para sair mais rápido).
- b) Restaurar o mais recente dos Backup’s completo, incluindo todos os arquivos: datafiles, redo log files e control files.
- c) Inicializar a instância novamente.

Obs:

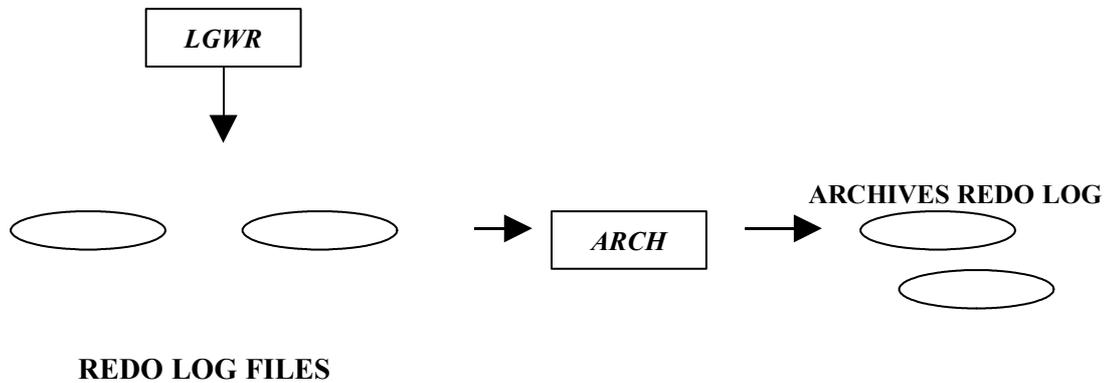
Se for necessário editar o arquivo de parâmetros para indicar o novo caminho do(s) control file(s), se necessário (em caso de perda de disco).

Por default, o BD fica no modo NOARCHIVELOG

MODOS DE ARQUIVAMENTO ORACLE

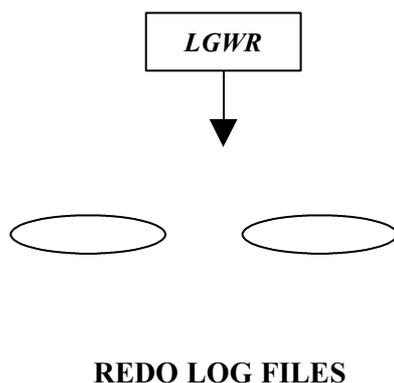
O Oracle pode trabalhar em dois modos de arquivamento: ARCHIVELOG e NOARCHIVELOG.

ARCHIVELOG



No modo ARCHIVELOG o Oracle além de usar o sistema circular de preenchimento no Redo Log files, ele copia, antes de fazer a troca, para os Archives Redo Log files (passo administrado pelo processo de segundo plano ARCH (ver definição deste processo background no início da apostila).

NOARCHIVELOG



Já no modo NOARCHIVELOG o Oracle somente faz as trocas, não armazenando em outros arquivos.

HABILITANDO/DESABILITANDO O MODO DE ARQUIVAMENTO

1) Para ver o modo atual do BD usar o comando ARCHIVE LOG LIST

Exemplo:

```
SVRMGR>ARCHIVE LOG LIST;
```

2) Os parâmetros LOG_ARCHIVE_START, LOG_ARCHIVE_DEST e LOG_ARCHIVE_FORMAT, são os parâmetros do “parameter file” que devem ser alterados para passar o modo para ARCHIVELOG (quando de um novo STARTUP).

Ex:

```
LOG_ARCHIVE_START = TRUE
```

```
LOG_ARCHIVE_DEST = diretório (ou nome do dispositivo).
```

```
LOG_ARCHIVE_FORMAT=nome_de_arquivo
```

Incluir no nome do arquivo “%s” para ser anexado no nome o número da seqüência de log (log sequence number).

3) Para trocar o modo de arquivamento automaticamente usa-se o comando ALTER SYSTEM:

Ex:

```
SVRMGR>ALTER SYSTEM ARCHIVE LOG START TO '/caminho/nome_arquivo'
```

Onde:

/caminho/nome_arquivo É o nome do diretório/arquivo onde será arquivado os archives redo log files.

Para desabilitar o modo de Arquivamento (NOARCHIVELOG):

A) Alterar o parâmetro LOG_ARCHIVE_START do arquivo de parâmetros:

```
LOG_ARCHIVE_START = FALSE
```

Daí, na próxima inicialização do BD o modo de arquivamento será NOARCHIVELOG.

B) Para parar automaticamente:

```
SVRMGR>CONNECT SYSTEM/MANAGER;
```

```
SVRMGR>ALTER SYSTEM ARCHIVE LOG STOP;
```

BACKUP ONLINE PARCIAL

Usar o comando ALTER TABLESPACE para preparar a TABLESPACE para o início da cópia:

```
ALTER TABLESPACE nome_tablespace BEGIN/END BACKUP
```

Onde:

BEGIN BACKUP Início do Backup com a Tablespace Online.

END BACKUP Fim do Backup com a Tablespace Online.

Ex:

```
SVRMGR>ALTER TABLESPACE TS_Treina BEGIN BACKUP;
```

```
SVRMGR>HOST cp $ORACLE_HOME/dbs/treina.dbf /tmp/dir_backup/treina.dbf
```

```
SVRMGR>ALTER TABLESPACE TS_Treina END BACKUP;
```

Obs:

- No Oracle Server para Windows NT usamos a interface gráfica do mesmo, o que facilita muito as atividades do DBA em relação ao BACKUP/RECOVER.

O comando “ cp “ utilizado acima é um comando de cópia do sistema UNIX.

Logo após copia-se os Control files com o comando ALTER DATABASE BACKUP CONTROLFILE.

Exemplo:

```
SVRMGR>ALTER DATABASE BACKUP CONTROLFILE
2> TO '/tmp/dir_backup/ctrltreina.ctl';
```

Copiar também o(s) arquivo(s) de parâmetro(s) , usando comandos do sistema operacional.

Além dos arquivos acima copiados, convém também ao DBA, copiar outros arquivos importantes de um servidor de BD:

Arquivos específicos de aplicações, como os executáveis, código-fonte, arquivos de configuração, dentre outros.

Os arquivos de configuração para o SQL*Net (TNSNAMES.ORA, LISTENER.ORA, SQLNET.ORA, etc.)

EXPORT/IMPORT/LOAD DO ORACLE

Os utilitários EXPORT e IMPORT do Oracle servem para mover dados de/para BD Oracle. O Export grava dados de um BD Oracle para arquivos do sistema operacional. Já o Import lê os dados de arquivos criados pelo Export de volta para o BD Oracle. O Export serve também para a criação de cópias suplementares de BD visando aumentar a flexibilidade da recuperação.

Temos diversos modos de “exportar”:

Modo	Descrição
Tabela	Exporta tabelas especificadas.
Usuários	Exporta todos os objetos de um usuário.
Banco de Dados completo	Exporta todos os objetos do BD.

Quando da “importação” alguns privilégios o usuário “importador” deve possuir: CREATE TABLE e TABLESPACE QUOTA ou UNLIMITED TABLESPACE, INSERT TABLE, ALTER TABLE.

Obs:

No Oracle somente usuários com privilégio EXP_FULL_DATABASE podem exporta o BD completo.