

Oracle Fundamentals



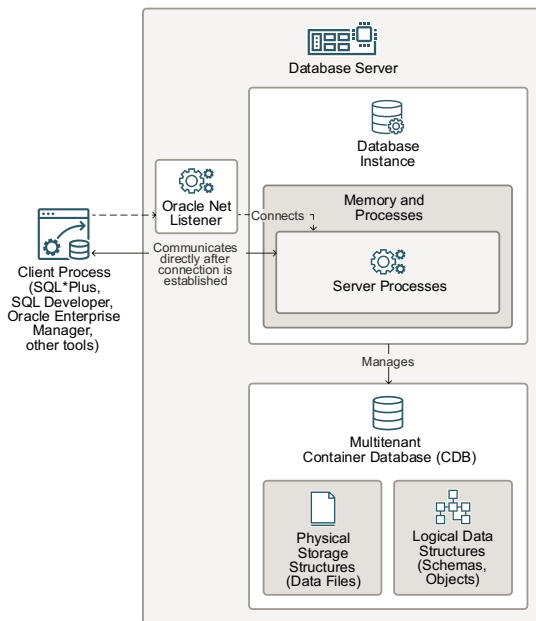
Arquitetura da instância

SUMÁRIO

1.	SERVIDOR DE BANCO DE DADOS	4
1.1.	INSTÂNCIA E DATABASE	4
1.2.	LISTENER.....	5
2.	DATABASE INSTANCE	6
2.1.	SINGLE INSTANCE DATABASE	6
2.2.	RAC DATABASE.....	7
2.3.	COMPONENTES DA INSTÂNCIA	7
2.4.	EXEMPLO PRÁTICO.....	8
2.5.	REFERÊNCIAS	8
3.	SYSTEM GLOBAL AREA (SGA)	9
3.1.	SHARED POOL	10
3.1.1.	LIBRARY CACHE.....	10
3.1.2.	DATA DICTIONARY CACHE.....	10
3.1.3.	EXEMPLO.....	11
3.2.	BUFFER CACHE.....	12
3.2.1.	EXEMPLO.....	12
3.3.	LOG BUFFER.....	14
3.3.1.	EXEMPLO.....	14
3.4.	LARGE POOL.....	15
3.4.1.	POR QUE É IMPORTANTE?	15
4.	PROGRAM GLOBAL AREA (PGA)	17
4.1.	SQL WORK AREA.....	18
4.2.	USER GLOBAL AREA (UGA).....	18
4.3.	PRIVATE SQL AREA.....	18
5.	PROCESS ARCHITECTURE.....	20
5.1.	CLIENT PROCESS E SERVER PROCESS	20
5.2.	CONNECTIONS AND SESSIONS	22
6.	BACKGROUND PROCESS	23
6.1.	DATABASE WRITER (DBWN)	24
6.1.1.	EXEMPLO.....	24

6.2.	LOG WRITER (LGWR)	25
6.2.1.	EXEMPLO	25
6.3.	CHECKPOINT (CKPT)	26
6.4.	ARCHIVER (ARCN)	27
6.5.	SYSTEM MONITOR (SMON)	29
6.6.	PROCESS MONITOR (PMON)	30

1. SERVIDOR DE BANCO DE DADOS



A estrutura de um servidor de banco de dados é composta por várias camadas e um DBA precisa ter uma compreensão muito forte dessa estrutura. Se você não entender essa estrutura, você terá dificuldades em administrar um banco de dados Oracle.

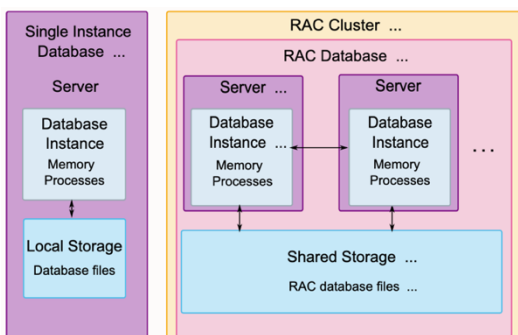
Quando se trata de um banco de dados Oracle, temos três componentes principais: o **Database**, **Instância** e **Listener**. Apesar do database e da instância estarem fortemente conectados, existe uma distinção entre o que é

um database e o que é uma instância. O database é a parte **física**, que é composto por **arquivos**, enquanto a instância é a parte **lógica**, composta por **memória** e **processos**. A instância existe para **gerenciar** o database.

Devido à esta forte correlação entre um database e uma instância, muitas vezes os dois são tratados como uma coisa só. Então é muito comum referenciarem a instância como se fosse o banco e vice-versa. Particularmente, não vejo problema nisso, porém é importante que você compreenda a que o database e a instância são fortemente ligados, porém são coisas diferentes.

Já o listener é um processo que roda no servidor e é responsável por gerenciar as conexões dos usuários com a instância.

1.1. INSTÂNCIA E DATABASE



O database só existe fisicamente em um único local e pode ser gerenciado por uma ou múltiplas instâncias. Já a instância ela está relacionada a apenas um único database. Então você pode ter um cenário onde você tenha uma única instância gerenciando o database ou você pode ter várias instâncias gerenciando o database.

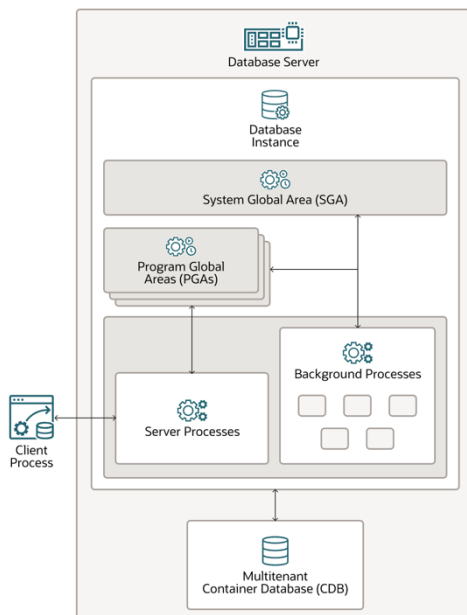
O fato de serem distintos (database e instância), significa que não necessariamente os dois estarão no mesmo local, principalmente no cenário de múltiplas instâncias gerenciando um único database.

1.2. LISTENER

Funcionando como um **intermediário** no servidor de banco, temos um processo chamado de **Listener** que tem como finalidade gerenciar as conexões de usuário com a **instância**. Então quando um usuário tenta se conectar no banco, o listener “recebe” esta conexão e cria uma sessão dentro da instância. Podemos notar que não há uma comunicação direta entre o usuário e o database, a instância fica responsável por gerenciar toda a interação entre o usuário e o banco de dados.

2. DATABASE INSTANCE

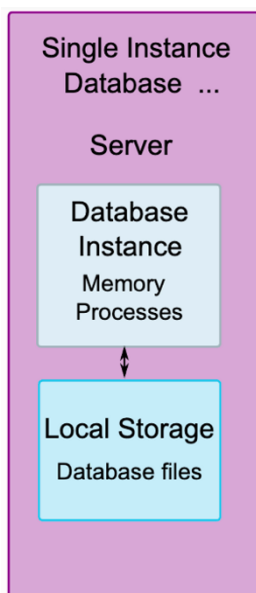
Database Instance



No contexto do Oracle Database, uma instância é um conjunto de recursos de software (memória e processos) que opera em um servidor e interage com o banco de dados. A instância é responsável por gerenciar e alocar recursos de sistema e banco de dados, facilitando o processamento e a execução de SQL e transações.

O banco de dados pode ser gerenciado uma instância (Single Instance) ou por várias instâncias ao mesmo tempo (Rac Database)

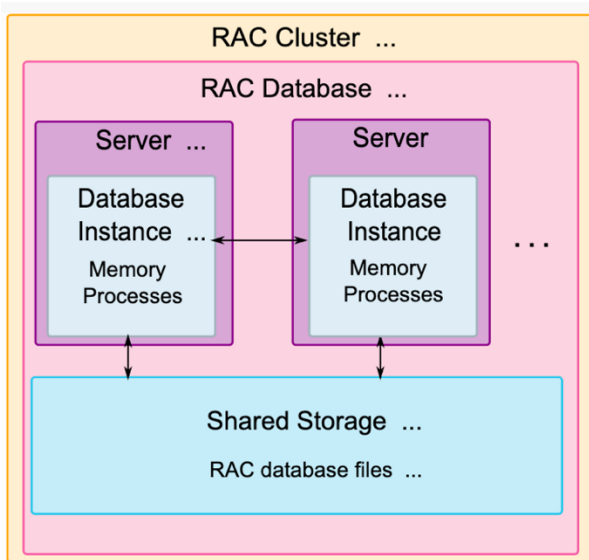
2.1. SINGLE INSTANCE DATABASE



Uma arquitetura de banco de dados **Single Instance** consiste em uma instância de banco de dados e um banco de dados.

Existe um relacionamento um-para-um entre o banco de dados e a instância do banco de dados. Vários bancos de dados de Single Instance podem ser criados no mesmo servidor.

2.2. RAC DATABASE

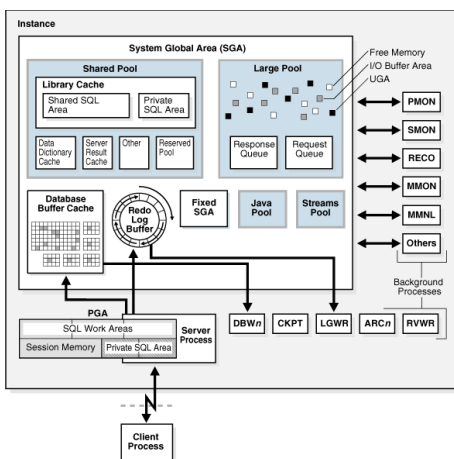


Uma arquitetura de banco de dados Oracle Real Application Clusters (Oracle RAC) consiste em diversas instâncias executadas em servidores separados.

Todos eles **compartilham** o mesmo banco de dados. O cluster de servidores aparece **como um único servidor**.

Essa configuração foi projetada para alta disponibilidade, escalabilidade e desempenho de ponta.

2.3. COMPONENTES DA INSTÂNCIA



System Global Area (SGA): É uma área de memória compartilhada que contém informações essenciais, como cache de dados e informações de controle, usadas por uma instância do Oracle Database.

Program Global Area (PGA): É uma área de memória não compartilhada que armazena dados e informações de controle para os processos do servidor Oracle.

Background Processes: São processos que executam tarefas de gerenciamento e manutenção para a instância e o banco de dados.

2.4. EXEMPLO PRÁTICO

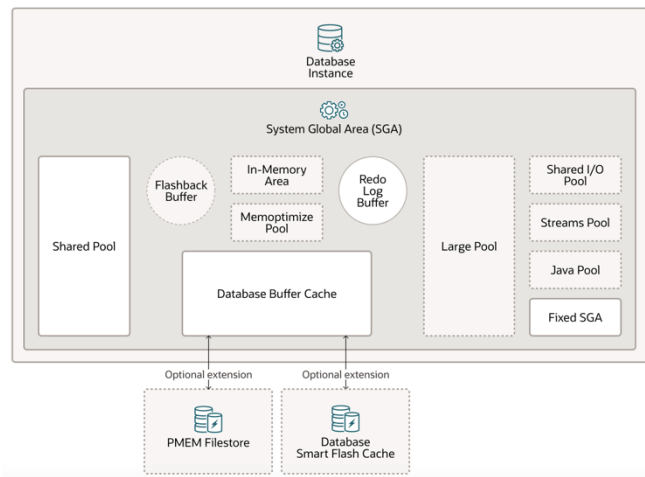
```
[oracle@ol8 ~]$ cat > /tmp/init_qqcoisa.ora <<EOF
*.db_name='qqcoisa'
EOF
[oracle@ol8 ~]$ export ORACLE_SID=qqcoisa
[oracle@ol8 ~]$ sqlplus / as sysdba
SQL> STARTUP NOMOUNT PFILE='/tmp/init_qqcoisa.ora';
SQL> COL SPID FORMAT a8
SQL> COL USERNAME form a12
SQL> set lines 120
SQL> SELECT USERNAME ,SPID, STID, PROGRAM FROM V$PROCESS ORDER BY SPID;
SQL> exit;
[oracle@ol8 ~]$ ps -ef | grep qqcoisa
```

2.5. REFERÊNCIAS

<https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/introduction-to-oracle-database.html#GUID-8F2EEEC8-0372-4419-88FF-7D77A9C0FCAD>

3. SYSTEM GLOBAL AREA (SGA)

System Global Area



A System Global Area (SGA) é a área de memória que contém dados e informações de controle para uma instância do Oracle Database.

Todos os processos de servidor e de background compartilham a SGA. Quando você inicia uma instância de banco de dados, a quantidade de memória alocada para a SGA é exibida.

Abaixo, as principais estruturas da SGA:

- DATABASE BUFFER CACHE
- SHARED POOL
- REDO LOG BUFFER

Quando você trabalha com um banco de dados Oracle, várias operações precisam ser realizadas, como buscar dados, alterar informações e gerenciar diferentes acessos ao sistema. A SGA é o espaço na memória do servidor onde o Oracle mantém todas as informações necessárias para realizar essas operações rapidamente. Sem ter que buscar constantemente as informações no disco rígido.

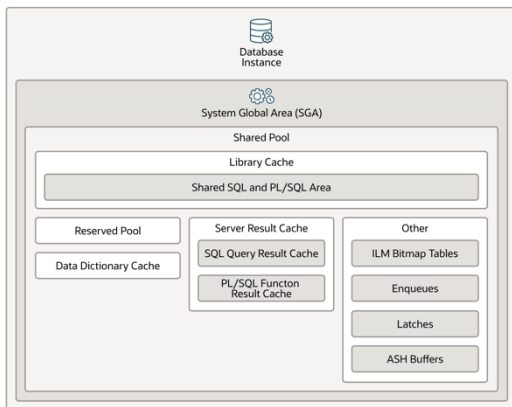
```
SQL> show parameter sga;
```

NAME	TYPE	VALUE
allow_group_access_to_sga	boolean	FALSE
lock_sga	boolean	FALSE
pre_page_sga	boolean	TRUE
sga_max_size	big integer	1536M
sga_min_size	big integer	0
sga_target	big integer	1536M

```
SQL> █
```

3.1. SHARED POOL

Shared Pool



A **Shared Pool** é uma parte da memória do sistema que o Oracle Database usa para armazenar informações que são compartilhadas entre usuários. Pense nela como uma área de armazenamento comum onde o banco de dados mantém informações que muitos usuários podem precisar acessar.

Exemplo

*Imagine que você está em uma biblioteca. Cada vez que você quer um livro, você pede ao bibliotecário, que então vai buscar o livro nas prateleiras. Isso pode levar algum tempo. Agora imagine se os **livros mais populares** já estivessem em uma mesa bem na **entrada da biblioteca**. Seria muito mais rápido pegar esses livros, certo? A **Shared Pool** funciona de maneira semelhante. Ela armazena informações **frequentemente acessadas** para que possam ser recuperadas rapidamente quando necessário.*

3.1.1. LIBRARY CACHE

Armazena **comandos SQL e blocos de código PL/SQL** que foram executados recentemente. Assim, se o mesmo comando SQL ou bloco PL/SQL for executado novamente, o Oracle pode simplesmente reutilizar as informações armazenadas na Shared Pool, em vez de ter que reler e reprocessar o comando. Esta área de cache é conhecida como Library Cache.

3.1.2. DATA DICTIONARY CACHE

Armazena informações sobre a estrutura do banco de dados, como nomes de tabelas, colunas e assim por diante. Isso ajuda o Oracle a entender como acessar e manipular os dados corretamente.

3.1.3. EXEMPLO

Imagine que você executa um comando SQL para buscar informações de um cliente em um banco de dados:

```
SELECT * FROM clientes WHERE id_cliente = 123;
```

Na primeira vez que esse comando é executado, o Oracle Database lê e processa o comando, e então armazena as informações sobre esse comando na Shared Pool.

Agora, se alguém mais executar o mesmo comando, o Oracle pode simplesmente verificar a Shared Pool e ver que já sabe como executar esse comando, tornando o processo muito mais rápido.

```
SQL> show parameter shared_pool_size
```

NAME	TYPE	VALUE
shared_pool_size	big integer	0

```
SQL> █
```

Então, em resumo, a Shared Pool é como uma área de armazenamento rápido no banco de dados Oracle, onde informações frequentemente usadas são mantidas para fácil acesso, ajudando a melhorar a eficiência e a velocidade das operações do banco de dados.

<https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/memory-architecture.html#GUID-1CB2BA23-4386-46DA-9146-5FE0E4599AC6>

3.2. BUFFER CACHE

O Database Buffer Cache é uma área na memória do sistema onde o Oracle Database armazena cópias temporárias de blocos de dados lidos de disco.

É uma parte crucial do sistema de gerenciamento de banco de dados, pois permite que o banco de dados acesse e manipule os dados muito mais rapidamente do que se tivesse que ler os dados do disco toda vez.

Imagine que você está lendo um livro. Cada vez que você vira a página, você tem que ir até a estante, pegar um novo livro e procurar a próxima página. Isso seria incrivelmente ineficiente!

O Database Buffer Cache funciona como se você mantivesse o livro aberto em seu colo, permitindo que você simplesmente vire as páginas para acessar as informações de que precisa.

```
SQL> show parameter db_cache_size;
```

NAME	TYPE	VALUE
db_cache_size	big integer	0

```
SQL> █
```

3.2.1. EXEMPLO

Imagine que você executa um comando SQL para buscar informações de um cliente em um banco de dados:

```
SELECT * FROM clientes WHERE id_cliente = 123;
```

Na primeira vez que esse comando é executado:

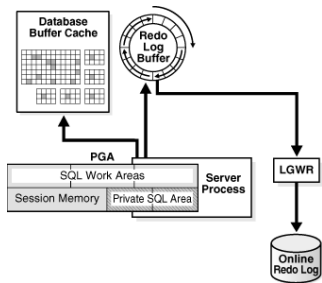
1. O Oracle verifica o Database Buffer Cache para ver se os dados do cliente com `id_cliente = 123` já estão lá.
2. Se não estiverem, o Oracle lê os dados do disco e os coloca no Database Buffer Cache.
3. O Oracle então retorna os dados para você.

Agora, se você ou outra pessoa executar o mesmo comando novamente pouco depois, o Oracle pode obter os dados diretamente do Database Buffer Cache, sem ter que lê-los do disco novamente. Isso torna a operação muito mais rápida.

Em resumo, o Database Buffer Cache é uma área de memória que armazena blocos de dados lidos de disco, permitindo que o Oracle Database acesse e manipule os dados muito mais rapidamente do que se tivesse que ler os dados do disco toda vez.

<https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/memory-architecture.html#GUID-4FF66585-E469-4631-9225-29D75594CD14>

3.3. LOG BUFFER



O Redo Log Buffer é uma área na memória do sistema que o Oracle Database usa para armazenar informações sobre transações que modificam os dados no banco de dados. Essas informações são chamadas de registros de redo, e elas são essenciais para a recuperação do banco de dados em caso de falha.

Quando uma transação é realizada e os dados são modificados:

1. As informações sobre as mudanças (registros de redo) são armazenadas no Redo Log Buffer.
2. Periodicamente, ou quando uma transação é confirmada, o Oracle escreve essas informações do Redo Log Buffer para os arquivos de redo log no disco.
3. Se o banco de dados falhar, o Oracle pode usar os arquivos de redo log para descobrir quais mudanças foram feitas e reverter ou completar essas mudanças durante o processo de recuperação.

NAME	TYPE	VALUE
log_buffer	big integer	4192K

```
SQL>
```

3.3.1. EXEMPLO

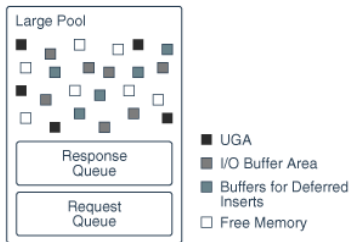
Imagine que você atualiza as informações de um cliente no banco de dados:

```
UPDATE clientes SET nome = 'Carlos' WHERE id_cliente = 123;  
COMMIT;
```

1. O Oracle armazena informações sobre essa atualização no Redo Log Buffer.
2. Quando você executa o COMMIT, o Oracle escreve as informações do Redo Log Buffer para os arquivos de redo log no disco.
3. Se o banco de dados falhar depois disso, o Oracle pode usar os arquivos de redo log para garantir que a atualização do nome do cliente seja preservada quando o banco de dados for recuperado.

<https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/memory-architecture.html#GUID-C2AD1BF6-A5AE-42E9-9677-0AA08126864B>

3.4. LARGE POOL



O **Large Pool** é uma área de memória separada dentro da System Global Area (SGA) no Oracle Database, que é usada para operações que necessitam de **grandes quantidades de memória**.

Ele é **opcional** e é utilizado principalmente para reduzir a contenção de memória que pode ocorrer durante **operações de backup, restauração** e processamento de mensagens em larga escala em sistemas de banco de dados distribuídos.

3.4.1. POR QUE É IMPORTANTE?

O Large Pool é importante porque ajuda a melhorar a eficiência e a performance do banco de dados ao fornecer uma área de memória dedicada para **operações específicas** que necessitam de grandes quantidades de memória.

Isso evita que essas operações consumam memória da Shared Pool ou do Database Buffer Cache, garantindo que essas áreas permaneçam otimizadas para suas funções específicas.

- **Alocação de Memória para Operações Específicas:** O Large Pool aloca memória para operações como backup e restauração do banco de dados, e para o processamento de mensagens em sistemas de banco de dados distribuídos.
- **Redução de Contenção de Memória:** Ao separar a memória usada para operações grandes e intensivas de memória, o Large Pool ajuda a reduzir a contenção de memória na Shared Pool e no Database Buffer Cache.

Imagine que você está realizando um backup do banco de dados. Essa operação requer uma quantidade significativa de memória para ser executada eficientemente.

```
SQL> show parameter large_pool;

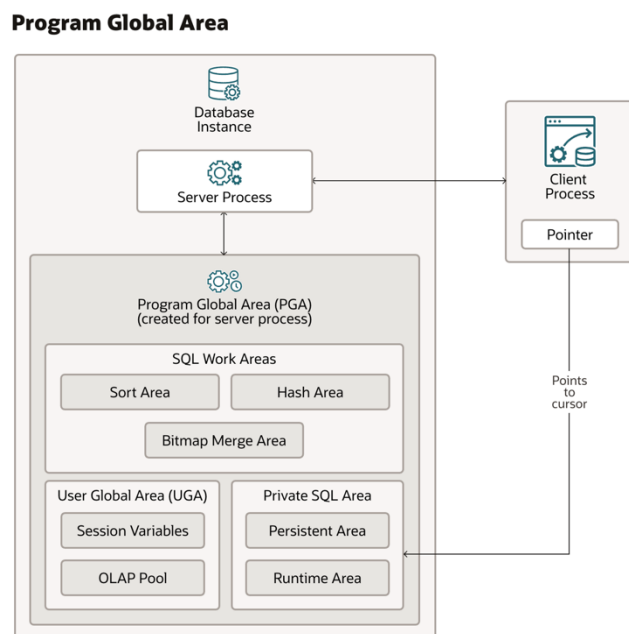
NAME                                TYPE        VALUE
-----
large_pool_size                      big integer 0
SQL>
```

Se o Large Pool estiver configurado, o Oracle Database pode usar a memória do Large Pool para essa operação, garantindo que o backup seja realizado de forma eficiente, sem afetar outras operações do banco de dados que estão usando o Shared Pool ou o Database Buffer Cache.

Em resumo, o Large Pool é uma área de memória dedicada dentro da SGA no Oracle Database, usada para operações que necessitam de grandes quantidades de memória, ajudando a melhorar a eficiência e a performance do banco de dados ao reduzir a contenção de memória.

<https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/memory-architecture.html#GUID-1ECB5213-AC4E-4BB4-9113-91C761676B34>

4. PROGRAM GLOBAL AREA (PGA)



A PGA (Program Global Area) é uma área de memória que é alocada para cada processo de servidor do Oracle Database. Cada processo de servidor, que pode ser um processo de usuário conectado ao banco de dados ou um processo em segundo plano, tem sua própria PGA. A PGA é usada para armazenar dados e informações de controle específicas desse processo. Ao contrário da SGA (System Global Area), que é compartilhada por todos os processos do banco de dados, a PGA é privada e **exclusiva** de cada processo.

A PGA é crucial para o funcionamento eficiente dos processos de servidor no Oracle Database. Ela armazena informações temporárias necessárias durante a execução de uma consulta ou transação, e ajuda a melhorar a performance e a eficiência ao manter essas informações na memória, em vez de no disco.

A PGA é crucial para o funcionamento eficiente dos processos de servidor no Oracle Database. Ela armazena informações temporárias necessárias durante a execução de uma consulta ou transação, e ajuda a melhorar a performance e a eficiência ao manter essas informações na memória, em vez de no disco.

```
SQL> show parameter pga;
```

NAME	TYPE	VALUE
pga_aggregate_limit	big integer	2G
pga_aggregate_target	big integer	512M

```
SQL> █
```

A PGA possui 3 áreas:

1. SQL WORK AREA
2. USER GLOBAL AREA (UGA)
3. PRIVATE SQL AREA

4.1. SQL WORK AREA

SQL Work Areas são áreas de memória na PGA usadas para operações de processamento de consultas SQL que precisam de espaço de memória adicional, como classificação e agrupamento.

Quando você executa uma consulta que requer, por exemplo, a classificação de dados, o Oracle usa uma SQL Work Area para armazenar e manipular os dados temporariamente durante a execução da consulta.

Se você executar uma consulta para listar os 10 principais clientes com base nas compras, o Oracle pode usar uma SQL Work Area para classificar todos os clientes por valor de compra antes de retornar os 10 principais.

4.2. USER GLOBAL AREA (UGA)

A User Global Area (UGA) é a área na PGA que armazena informações de sessão e estado para cada usuário conectado ao banco de dados.

Quando um usuário se conecta ao banco de dados, o Oracle aloca espaço na PGA para a **UGA** desse usuário, onde armazena informações como variáveis e cursores SQL.

Se você estiver conectado ao banco de dados e executar várias consultas, o Oracle usará a UGA para armazenar informações sobre suas consultas e outros dados de sessão.

4.3. PRIVATE SQL AREA

A Private SQL Area é uma área na PGA que armazena informações e recursos necessários para processar uma instrução SQL específica.

Quando você executa uma instrução SQL, o Oracle aloca espaço na PGA para uma **Private SQL Area**, onde armazena informações como o plano de execução da consulta e outros recursos necessários.

Se você executar uma consulta para buscar dados de uma tabela, o Oracle usará uma Private SQL Area para armazenar e gerenciar os recursos necessários para executar essa consulta.

SQL Work Areas, UGA e Private SQL Area são componentes da PGA que o Oracle usa para armazenar e gerenciar informações e recursos necessários para executar consultas SQL e gerenciar sessões de usuário.

Eles ajudam a melhorar a eficiência e a performance do banco de dados, garantindo que o Oracle tenha acesso rápido e fácil às informações e recursos de que precisa.

<https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/memory-architecture.html#GUID-859795E2-87CD-442B-B36F-584A77755F59>

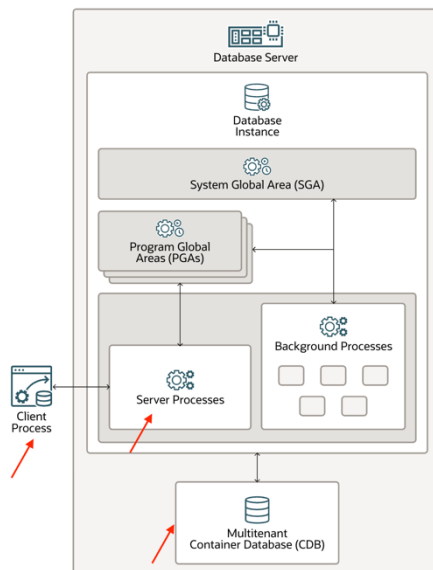
5. PROCESS ARCHITECTURE

Um processo é um mecanismo do **Sistema Operacional** que permite que uma série de instruções possam ser executadas. O Oracle Database cria diversos processos, que estão divididos nos seguintes grupos:

- Client Process
- Oracle Process
 - Server Process
 - Background Process

5.1. CLIENT PROCESS E SERVER PROCESS

Database Instance



Antes do usuário ou aplicação submeter uma instrução SQL no Oracle Database, primeiro ele precisa se conectar na instância. Ao criar essa conexão, o SO cria um client processes (também chamado de user processes) para executar esta instrução.

Como podemos perceber na imagem ao lado, quando uma conexão é estabelecida com o banco, o Client processes não é criado no servidor e sim no computador que ele executou o programa para se conectar no banco. Porém, se você realizar esta conexão a partir do próprio servidor, o Client processes também será criado no servidor.

Um outro ponto a se destacar é que o Client processes não interage diretamente com o banco de dados. Ele cria uma **sessão** no banco de dados e inicia um **Server Processes** para aquela sessão.

No exemplo abaixo, uma conexão é criada com o banco de dados a partir de um host com um usuário de SO chamado “Client User”

```
SQL> CONNECT SYS@inst1 AS SYSDBA
```

```
Enter password: *****
```

```
Connected to an idle instance.
```

No host aonde a conexão foi criada, podemos ver o Client Processs criado no SO

```
% ps -ef | grep -e sample -e sqlplus | grep -v grep
clientuser 29437 29436 0 15:40 pts/1 00:00:00 sqlplus as sysdba
```

Já no servidor de banco de dados, podemos verificar o Server Processs criado.

```
% ps -ef | grep -e sample -e sqlplus | grep -v grep
serveruser 29441 1 0 15:40 ? 00:00:00 oraclesample (LOCAL=NO)
```

Após a conexão estabelecida, tanto o Client Process quanto o Server Process foram criados, o primeiro no computador no qual foi executado o programa e o segundo, no servidor de banco de dados.

O Server Process será responsável por toda a interação com o banco de dados, desta forma, o Client Process sempre se comunicará com o Server Process para ter suas requisições atendidas.

Existem duas configurações para o Server Processes:

- **Dedicated Server:** O Client Process é associado a um Server Process exclusivo
- **Shared Server:** O Client Process é associado a um Dispatcher Process que atende a vários Clients Processes

A grande maioria dos ambientes trabalham com a configuração Dedicated Server, inclusive é a configuração default.

```
SQL> select username, status, server from v$session where username = 'SYS';
```

USERNAME	STATUS	SERVER
SYS	ACTIVE	DEDICATED
SYS	ACTIVE	DEDICATED

```
SQL> █
```

5.2. CONNECTIONS AND SESSIONS

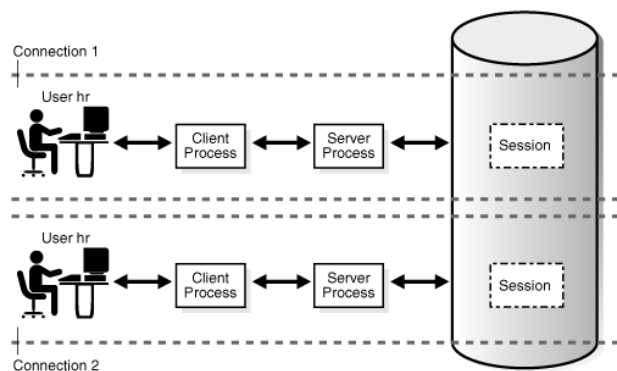
Uma conexão é o meio físico pelo qual as solicitações do cliente são transmitidas ao servidor do banco de dados e as respostas são enviadas de volta ao cliente.

A sessão é o contexto lógico no Oracle Database associado a um usuário específico e sua conexão atual. Uma sessão começa quando um usuário se autentica e termina quando o usuário se desconecta.

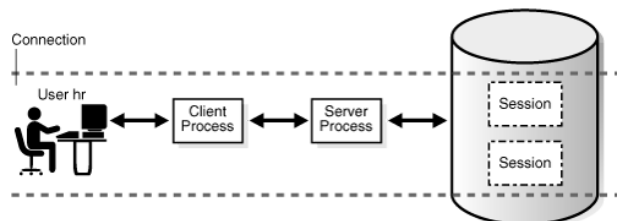
Em outras palavras, enquanto a conexão é o canal físico, a sessão representa o ambiente lógico e personalizado que acompanha o usuário enquanto ele está conectado ao banco de dados.

Normalmente, quando um usuário se conecta ao banco de dados, uma sessão é automaticamente criada para essa conexão.

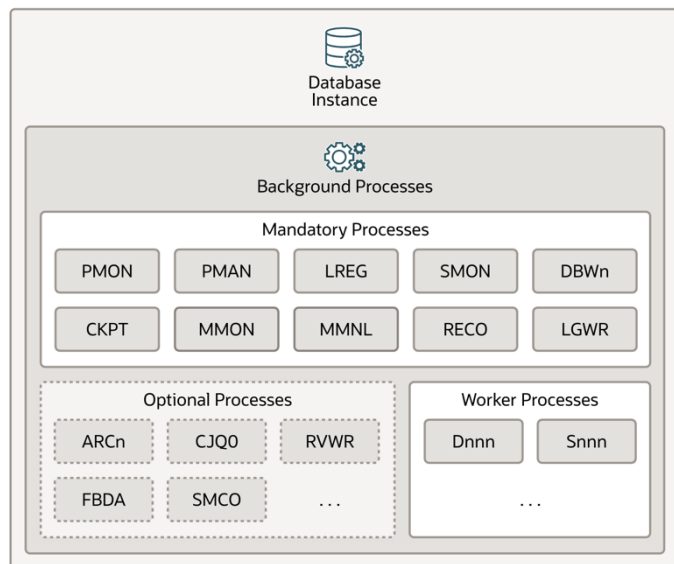
No exemplo ao lado, podemos notar que para cada conexão criada uma sessão no banco de dados.



Já neste exemplo, temos uma conexão com mais de uma sessão. Uma conexão pode criar uma ou várias sessões no banco de dados



6. BACKGROUND PROCESS



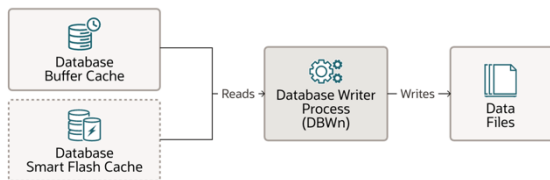
Os **Background Processes** são processos em segundo plano que executam tarefas essenciais para o funcionamento do banco de dados Oracle. Eles operam automaticamente e de forma independente dos processos de usuário, ajudando a gerenciar e otimizar as operações do banco de dados.

Os **Background Processes** são cruciais porque realizam várias tarefas administrativas e de manutenção que são essenciais para a operação eficiente, confiável e segura do banco de dados Oracle.

Os **Background Processes** são iniciados quando a instância é inicializada.

6.1. DATABASE WRITER (DBWN)

Database Writer Process (DBWn)



O Database Writer (frequentemente abreviado como DBWn, onde "n" é um número) é um processo de background no Oracle Database responsável por transferir os blocos de dados modificados da memória para o disco. Esses blocos de dados modificados na memória são chamados de "blocos sujos", e a área de memória onde eles são mantidos é o Database Buffer Cache, uma parte da System Global Area (SGA).

Uma vez que essa alteração é feita na memória e ainda não foi escrita de volta para os arquivos de dados no disco, esse bloco de memória é marcado como "sujo".

6.1.1. EXEMPLO

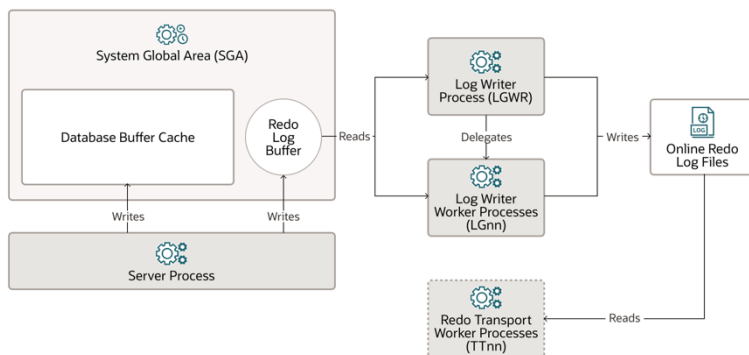
Imagine que você tem um aplicativo de gerenciamento de clientes e deseja atualizar o endereço de um cliente existente

```
UPDATE clientes SET endereco = 'Rua das Flores, 123' WHERE id = 101;
```

1. **Execução da Instrução:** Quando você executa a instrução UPDATE, o Oracle localiza o registro do cliente com id = 101 no Database Buffer Cache e atualiza o campo endereço com o novo valor.
2. **Registro no Redo Log Buffer:** Paralelamente, essa alteração é registrada no Redo Log Buffer
3. **Atuação do Log Writer (LGWR):** O LGWR, ao detectar entradas no Redo Log Buffer, escreve essas entradas nos arquivos de redo no disco. Isso garante que, em caso de falha, as alterações possam ser recuperadas.
4. **Confirmação ao Usuário:** Uma vez que o LGWR escreveu as entradas de redo no arquivo de redo, o Oracle confirma ao usuário que a operação foi bem-sucedida. Neste momento, você vê uma mensagem como "1 linha atualizada".
5. **Atuação do Database Writer (DBWn):** Enquanto isso, o DBWn, monitorando o Database Buffer Cache, identifica que o bloco contendo o registro do cliente foi modificado (um "bloco sujo"). Em algum momento após a confirmação ao usuário (e isso pode ser imediatamente ou após um certo tempo, dependendo de vários fatores), o DBWn decide escrever esse bloco modificado de volta para o arquivo de dados no disco. Isso garante que a atualização do endereço seja permanentemente salva.

6.2. LOG WRITER (LGWR)

Log Writer Process (LGWR)



Quando você faz alterações nos dados do banco de dados, o Oracle registra essas alterações em uma área especial da memória chamada Redo Log Buffer. O Log Writer é o processo responsável por pegar essas alterações do Redo

Log Buffer e gravá-las em um arquivo especial chamado Redo Log File.

6.2.1. EXEMPLO

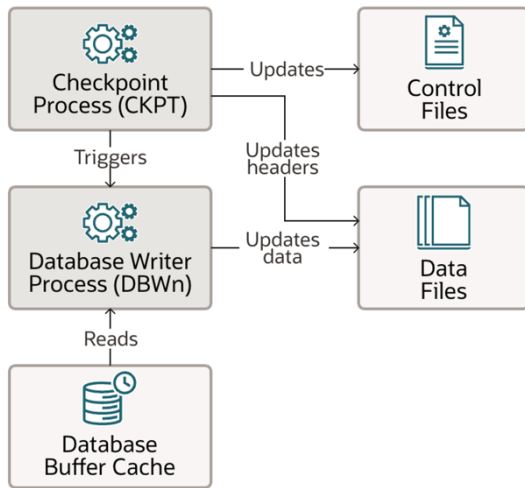
Imagine que você tem um aplicativo de gerenciamento de clientes e deseja adicionar um novo cliente ao banco de dados. Você executa a instrução SQL:

```
INSERT INTO clientes (id, nome, email) VALUES (101, 'Maria Silva', 'maria.silva@email.com');
```

- Execução da Instrução:** Quando você executa a instrução INSERT, o Oracle começa a processar a solicitação e prepara os dados para serem inseridos na tabela clientes.
- Registro no Redo Log Buffer:** Antes de realmente inserir os dados na tabela, o Oracle registra a instrução INSERT e os dados associados em uma área de memória chamada Redo Log Buffer. Esta é uma espécie de "rascunho" que o Oracle mantém para todas as alterações que estão sendo feitas.
- Atuação do Log Writer:** Enquanto isso, o Log Writer está constantemente monitorando o Redo Log Buffer. Ele vê que há uma nova instrução INSERT registrada e decide gravá-la em um arquivo no disco chamado Redo Log File. Esta ação é como se o Oracle estivesse dizendo: "Vou salvar essa alteração em um local seguro, caso algo dê errado e eu precise lembrar o que foi feito".
- Confirmação ao Usuário:** Uma vez que o Log Writer grava a instrução no Redo Log File, o Oracle finalmente insere os dados na tabela clientes e envia uma confirmação de que a operação foi bem-sucedida. Isso é quando você vê a mensagem "1 linha inserida" ou algo semelhante.

6.3. CHECKPOINT (CKPT)

Checkpoint Process (CKPT)



Em determinados momentos, o Oracle decide fazer um "salvamento automático" dos dados. Isso pode acontecer depois de um certo tempo ou quando muitas alterações foram feitas no Database Buffer Cache.

Gravando as Alterações: Durante um Checkpoint, o Oracle olha para todas as alterações que foram feitas no Database Buffer Cache e as grava nos arquivos de

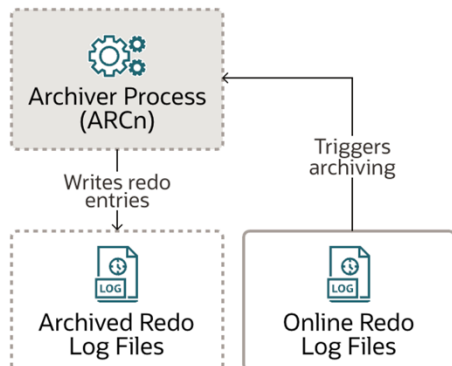
dados do banco de dados.

Atualizando o "Histórico": Além de gravar as alterações, o Oracle também atualiza um "histórico" (chamado de redo log) que mantém um registro de todas as alterações. Isso é útil se o banco de dados precisar ser recuperado.

Pense no Checkpoint como um "salvamento automático" para o banco de dados Oracle. Ele garante que todas as alterações feitas no Database Buffer Cache sejam gravadas de forma segura nos arquivos de dados do banco de dados e ajuda o banco de dados a se recuperar rapidamente em caso de problemas. É uma das muitas maneiras pelas quais o Oracle mantém os dados seguros e acessíveis.

6.4. ARCHIVER (ARCn)

Archiver Process (ARCn)



O ARCn é um dos processos de background do Oracle Database responsável por arquivar os registros de redo logs. Estes registros contêm um histórico de todas as alterações feitas no banco de dados.

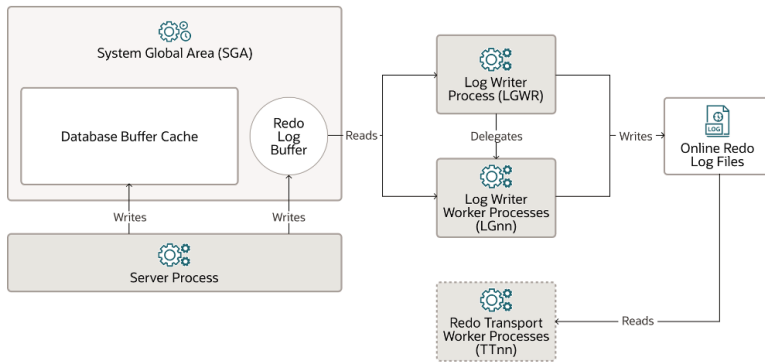
Funções principais do ARCn

- **Arquivamento de Redo Logs:** O Oracle grava todas as alterações feitas no banco de dados em arquivos chamados "redo log files". Quando um desses arquivos fica cheio, o ARCn entra em ação, copiando seu conteúdo para arquivos especiais chamados "archive logs". O LGWR não pode reutilizar e sobrescrever um redo log até que o redo log tenha sido arquivado.
- **Preparação para Recuperação:** Os "archive logs" são vitais para a recuperação do banco de dados. Se houver algum problema, esses arquivos permitem que o banco de dados seja restaurado ao seu estado anterior, usando o histórico de alterações armazenadas neles.

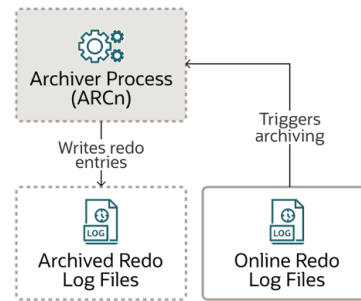
Quando o ARCn age

- **Quando um "redo log file" está quase cheio:** O ARCn começa a arquivar as informações para garantir que o espaço não acabe.
- **Durante uma troca de "redo log file":** O Oracle, às vezes, decide usar um novo "redo log file". Quando isso acontece, o ARCn arquiva o conteúdo do "redo log file" antigo.

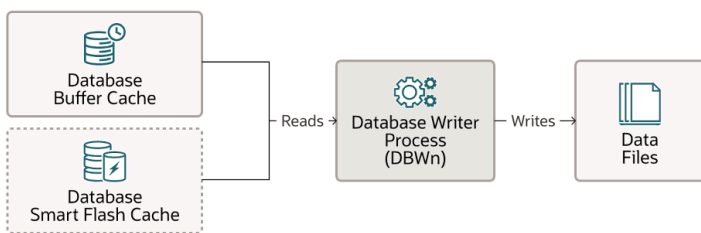
Log Writer Process (LGWR)



Archiver Process (ARCn)

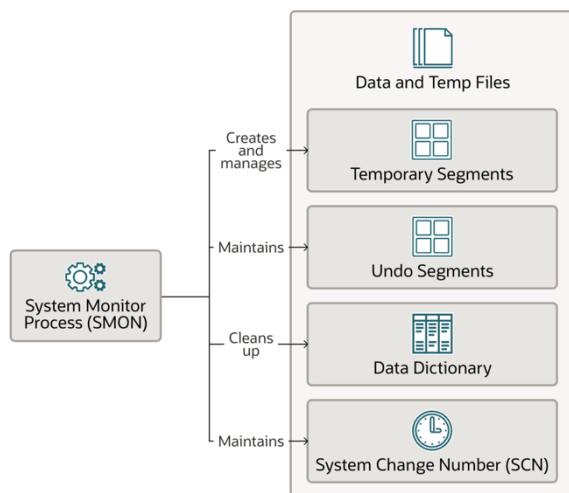


Database Writer Process (DBWn)



6.5. SYSTEM MONITOR (SMON)

System Monitor Process (SMON)



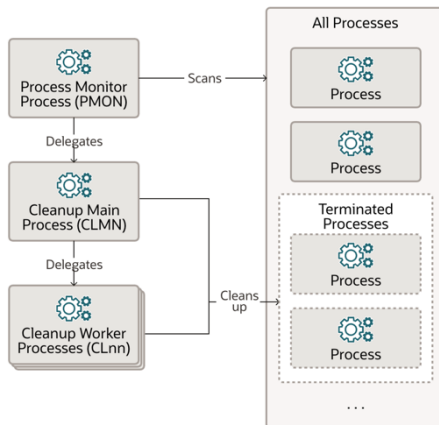
O SMON, ou System Monitor, é um dos muitos processos de background essenciais no Oracle Database. Ele desempenha um papel crucial na manutenção da saúde e estabilidade do banco de dados.

Funções principais do SMON

- **Recuperação de Instância:** Se o banco de dados sofrer uma falha inesperada (como uma queda de energia ou falha no sistema), o SMON é responsável por garantir que o banco de dados seja recuperado corretamente na próxima inicialização. Ele faz isso ao verificar os arquivos de redo log para quaisquer transações que não foram confirmadas (committed) antes da falha e garantindo que essas transações sejam completadas ou desfeitas.
- **Limpeza de Espaço Temporário:** O SMON também é responsável por liberar espaço que foi temporariamente alocado para operações de banco de dados, mas que não é mais necessário. Isso ajuda a garantir que o banco de dados não consuma mais espaço do que o necessário.
- **Monitoramento de Espaço:** O SMON monitora regularmente o espaço disponível no banco de dados e pode desencadear ações para liberar espaço ou alertar os administradores se o espaço estiver ficando baixo.
- **Limpeza do Dicionário de Dados:** Se houver algum problema ou inconsistência no dicionário de dados, o SMON entra em ação para corrigi-lo.
- **Manutenção da área de Undo:** O SMON ajusta e otimiza o espaço usado para desfazer ações no banco de dados, baseado em como esse espaço está sendo usado.

6.6. PROCESS MONITOR (PMON)

Process Monitor Process (PMON)



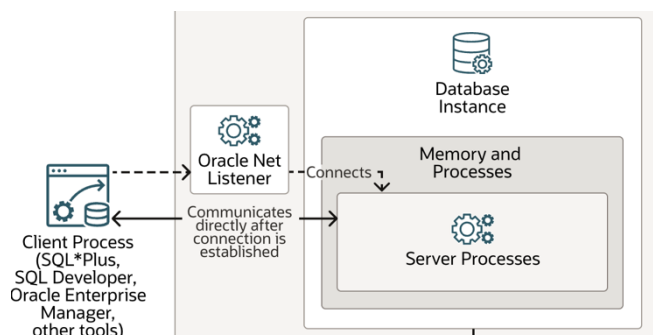
Monitora e limpa os recursos de processos falhos, garantindo que os recursos sejam liberados para reutilização após uma falha de processo.

Funções principais do PMON

- **Auxílio em Problemas:** Se um usuário estiver conectado ao banco de dados e, por algum motivo, sua conexão for interrompida (como uma falha de energia ou um problema de rede), o PMON percebe isso e toma medidas para "limpar" os recursos que esse usuário estava usando, garantindo que o banco de dados continue funcionando bem para os outros usuários.
- **Recuperação de Recursos:** O PMON garante que os recursos (como memória) que estavam sendo usados por usuários ou operações que não terminaram corretamente sejam liberados e possam ser usados novamente.

<https://docs.oracle.com/en/database/oracle/oracle-database/23/cncpt/process-architecture.html#GUID-FB843ADE-8DDD-4F83-8EB9-D4B5E4B6B022>

7. ORACLE NET ARCHITECTURE



O Oracle Net é um componente que permite a comunicação entre um cliente e um servidor de banco de dados Oracle, ou entre servidores de banco de dados Oracle em diferentes locais. Oracle Net facilita a troca de

dados entre processos de software que estão rodando em computadores em locais diferentes.

É usado sempre que uma aplicação precisa acessar um banco de dados Oracle de forma remota, sendo uma parte integral de todas as instalações do Oracle Database.

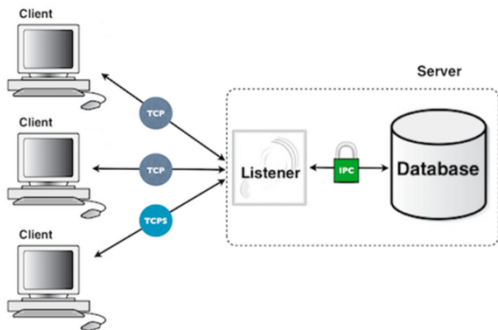
7.1. CONFIGURAÇÃO

A configuração do Oracle Net é feita através de arquivos como o **listener.ora** (para o servidor) e o **tnsnames.ora** (para o cliente), que definem como as conexões de rede são estabelecidas, incluindo os endereços de rede, protocolos e nomes de serviços. Ambos os arquivos podem ser criados/alterados pelo utilitário **netca** ou realizando uma edição manual do arquivo.

7.1.1. FERRAMENTAS

Inclui ferramentas como **tnsping** para diagnosticar a conectividade entre o cliente e o Listener, e **lsnrctl**, uma ferramenta de controle para gerenciar o Listener.

7.2. LISTENER



O Listener no Oracle é um processo de servidor fundamental que reside no host do servidor e ouve solicitações de sessão de clientes. Ele é essencial para a comunicação entre o cliente e o servidor em uma rede. A configuração e administração do Listener são gerenciadas

através do utilitário Oracle Net Services.

7.2.1. FUNÇÕES DO LISTENER

- **Escuta Solicitações de Conexão:** O Listener monitora as conexões de rede em uma ou mais portas (a porta padrão é a 1521) e estabelece uma comunicação inicial entre o cliente e o servidor.
- **Registro de Serviços:** O Listener gerencia os registros dinâmicos dos serviços de banco de dados. Os serviços do banco de dados registram-se automaticamente com o Listener (isso é conhecido como registro dinâmico). O Listener utiliza essas informações para direcionar as solicitações de conexão para a instância de banco de dados apropriada.
- **Roteamento de Conexão:** Após receber uma solicitação de conexão, o Listener determina a instância de banco de dados apropriada com base no serviço solicitado e facilita a comunicação inicial para estabelecer a sessão entre o cliente e o servidor.

7.2.2. CONFIGURAÇÃO BÁSICA

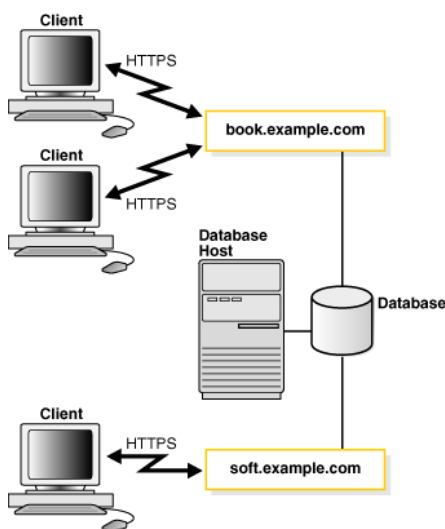
O arquivo de configuração principal do Listener é o **listener.ora**, localizado no diretório **\$ORACLE_HOME/network/admin**. Este arquivo contém configurações como o nome do Listener, o protocolo de rede, a porta de escuta e os serviços de banco de dados associados.

```
LISTENER =  
  (DESCRIPTION_LIST =  
    (DESCRIPTION =  
      (ADDRESS = (PROTOCOL = TCP) (HOST = vm01o18193.localdomain) (PORT = 1521))  
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1521))  
    )  
  )
```


7.2.3. NOMES DE SERVIÇO

Um nome de serviço é uma representação lógica de um serviço usado para conexões de cliente.

Ele permite que diferentes clientes identifiquem o mesmo banco de dados de maneiras diferentes, e os administradores podem alocar recursos de forma mais eficaz com base no serviço solicitado.

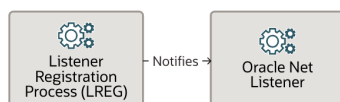


7.2.4. REGISTROS DE SERVIÇO

No Oracle Net, o registro de serviço é uma característica pela qual o processo LREG registra dinamicamente informações da instância com um listener.

Isso permite que o listener encaminhe solicitações de conexão de clientes para o manipulador de serviço apropriado. A vantagem é que o registro é dinâmico e não requer configuração manual, reduzindo a sobrecarga administrativa.

Listener Registration Process (LREG)



Além do registro dinâmico, também é possível registrar manualmente os serviços no listener. Normalmente esta configuração é realizada quando você está fazendo operações específicas como duplicando um banco ou configurando um Data Guard.

7.2.5. GERENCIAMENTO DE CONEXÕES

O listener é responsável por gerenciar as conexões através da rede no banco de dados. Caso listener não esteja disponível, mesmo que o banco de dados esteja no ar, não será possível realizar conexões externas ao servidor.

7.3. TNSNAMES

O arquivo é composto por entradas que mapeiam "alias" de conexão a especificações de rede para alcançar uma instância do banco de dados Oracle. Cada entrada no tnsnames.ora define um nome de serviço que os clientes podem usar para conectar-se a um banco de dados específico. Essa entrada inclui detalhes como o nome do host, o número da porta e o SID (System Identifier) ou o Service Name do banco de dados.

```
CDB1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = vm01o18193.localdomain) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = cdb1)
    )
  )
```

<https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/application-and-networking-architecture.html#GUID-75A4D49A-87AE-49C0-A49E-2A3486E80053>