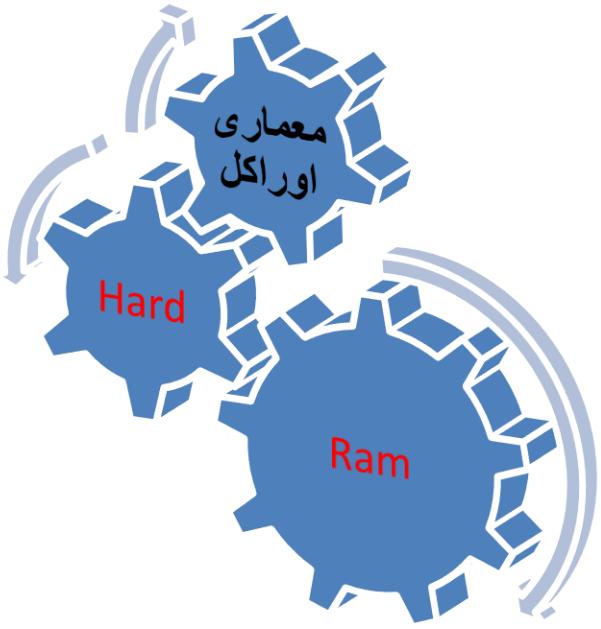
ORACLE_ARCHITECTURE



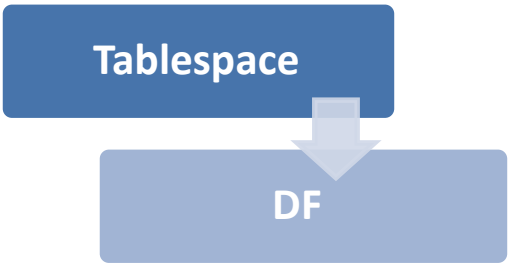
معماری اوراکل در Hard شامل بخش های ذیل میباشد:

۱- DF (Data File)

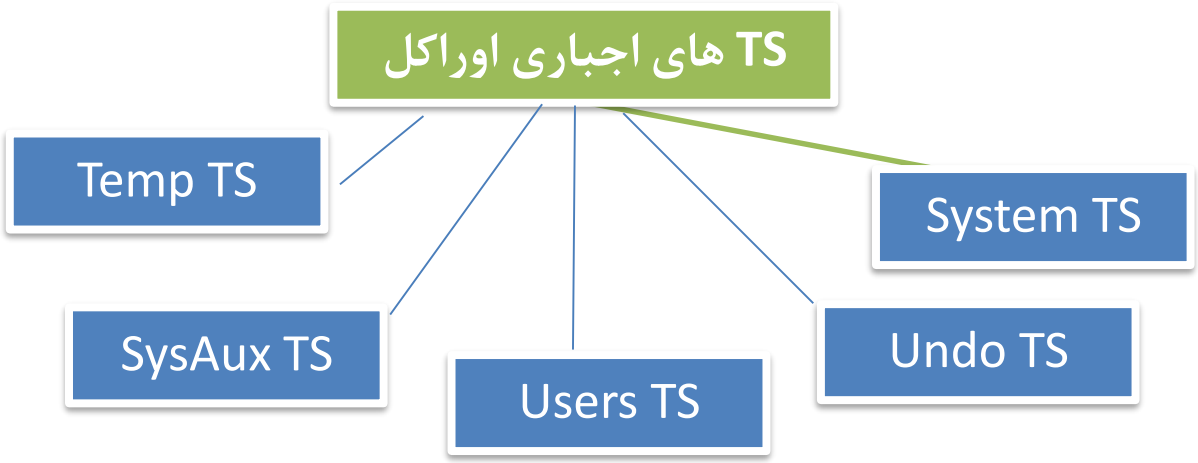
اشیائی که نیاز به فضای هارد دارند در داخل DFها ذخیره می شوند از حمله جداول،ایندکس ها و غیره در اوراکل از ۱ تا ۱۰۰ DF می توان ایجاد کرد.

توزیع اطلاعات بر روی دیسک از طریق تعریف چندین DF برروی دیسک های مختلف انجام می گیرد و اینکار می تواند کارایی بانک اطلاعاتی را افزایش دهد.

DF ها در ساختاری منطقی با نام Tablespace قرار داده می شوند هر Tablespace می تواند یک یا چند DF را در خود قرار دهد.



یک Tablespace می تواند اطلاعات یک یا چند Object را درون خود نگهداری کند. به طور کلی هنگامی که جدول یا ایندکس ایجاد می کنیم می توانیم Tablespace آن را مشخص کنیم بستگی به اینکه TSها چند DF و هرکدام از DFها بر روی کدام پارتیشن دیسک تعریف شده اند اطلاعات را بر روی آن قرار داده می شود.



۲- Online Redo Log File

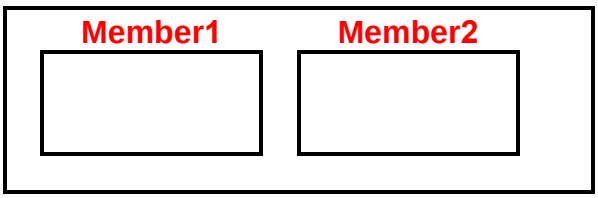
هر تغییری در ساختار یا اطلاعات موجود بانک اطلاعاتی در این فایل ها نوشته می شود مزیتی که این فایل ها برای بانک اطلاعاتی اجرا می کند شامل موارد ذیل است:

۱) افزایش سرعت بانک اطلاعاتی از طریق کاهش دفعات نوشتن تغییرات بر روی دیسک.

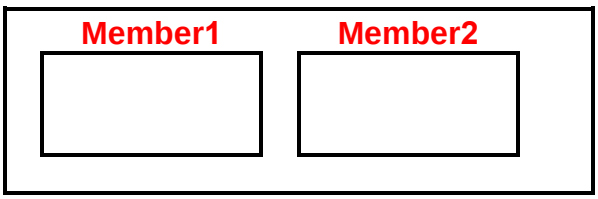
۲) در این فایل ها فقط داده های تغییر یافته نگهداری می‌شود ، اگر به هر دلیلی پایگاه داده فرصت نوشتن اطلاعات بر روی دیسک را پیدا نکرد اطلاعات تغییر یافته در یکی از فایل های Redo Log File وجود دارد و پایگاه داده می تواند از طریق آن Recovery را انجام دهد.

Redo Log File ها در دسته بندی های با نام Redo Log File Group قرار داده می شوند که هر Group می تواند شامل یک یا چندین Redo Log File باشد. وجود حداقل ۲ گروه برای بانک اطلاعاتی الزامی است ، اوراکل به هنگام نصب ۲ گروه که در داخل هر گروه یک Redo Log File وجود دارد را در نظر میگیرد.

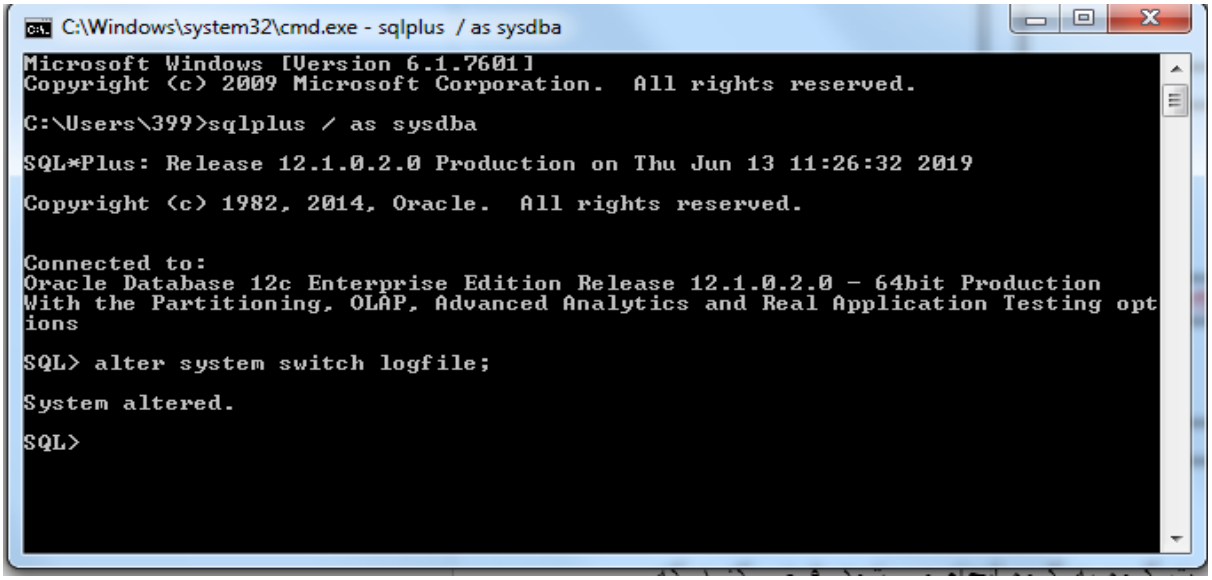
Group #1



Group #2



نحوه نوشتن اطلاعات بر روی Redo Log File ها بصورت گروه به گروه انجام می شود، فرض کنید که Group #1 در حال حاضر Current است و اطلاعات بر روی آن نوشته می شود به محض اینکه فضای Redo Log File های موجود در Group #1 تکمیل شد عملیات Log Switch اتفاق می افتد ، با انجام این عملیات Check Point اتفاق می افتد که Check Point با صدا زدن DBWR(DataBase WRiter) نوشتن اطلاعات موجود در Redo Log File ی که هم اکنون Active است را انجام می دهد، Log Swith را می توان بصورت دستی هم اجرا کرد.



SQL>alter system switch logfile;

با انجام این دستور Check Point صدا زده می وشود ولی برعکس آن وجود ندارد یعنی با صدا کردن دستی Check Point عملیات Log Switch اتفاق نمی افتد.

Multi Plexing

همانطور که گفتیم Redo Log File ها در دسته بندی با نام Group قرار داده می شوند برای محافظت از Redo Log File ها می توان در هر گروه بیش از یک member قرار داد البته برای کارایی (performance) بهتر پیشنهاد می شود که آنها را در دیسک های جداگانه قرار دهید تا در صورت آسیب دیدن یکی از دیسک ها بتوان از کپی Redo Log File که در دیسک دیگر قرار دارد استفاده کرد، رفتار اوراکل با member های یک گروه همانند هم است، یعنی هر وقت LGWR (Log WRiter) نوشتن در یکی از Redo Log File ها را شروع می‌کند اطاعات را در تمامی member های موجود در گروه جاری انجام می‌دهد.

۳- Control Files

فایل Binary با اهمیت که اطلاعات رفتاری اوراکل در آن نگهداری می شود و هر وقت تغییری در ساختار فیزیکی اوراکل اتفاق بیافتد تغییرات در این فایل ذخیره می شود اطلاعاتی که در این فایل وجود دارند شامل:

- ✓ نام بانک اطلاعاتی
- ✓ زمان ایجاد بانک اطلاعاتی
- ✓ نام و محل های DF (Data File) ها و CF (Control File) ها
- ✓ Log Sequence Number جاری
- ✓ شماره SCN (System Change Number)
- ✓ اطلاعات TS (TableSpace) ها
- ✓ اطلاعات Check Point
- ✓ اطلاعاتی در مورد RMan (Recovert Manager)
- ✓ و غیره

وجود CF (Control File) برای بانک اطلاعاتی ضروری است و بدون آن امکان ندارد تا پایگاه داده بتواند به مرحله mount برود بنابراین پیشنهاد می شود که از آن کپی تهیه کرده و در دیسک های مختلف قرار دهید.

٤- Archive Redo Log Files (Offline)

اطلاعات بر روی Online Redo Log File ها به صورت چرخشی بازنویسی می شود و به این علت امکان دسترسی به تمامی Redo Log File های قبلی وجود ندارد ، برای رفع این مشکل DBA ها پس از فعال کردن قابلیت آرشیو به پایگاه داده فرمان دهند تا پس از اتمام کار یک کپی از Online Redo Log File تهیه کرده و در مسیر مشخصی نگهداری کند (آرشیو کند) ، مزیت استفاده از این قابلیت این است که در صورت خرابی پایگاه داده میتوان به آخرین داده های ثبت شده در پایگاه داده دسترسی پیدا کرده و پایگاه داده را بصورت کامل بازیابی (Recovery) کنیم .

اگر Archive Redo Log File فعال نباشد و از Online Redo Log File های قبلی کپی نداشته باشیم باید به آخرین بک آپ گرفته شده بازگردیم.

فرض کنید که از پایگاه داده هر شب ساعت ۲ بامداد بصورت کامل نسخه پشتیبان (Backup) تهیه می‌شود و کاربران از ساعت ۷ صبح شروع به کار می‌کنند و Online Redo Log File ها دائما بازنویسی می‌شوند در ساعت ۱۲ ظهر خرابی در یکی از دیسک ها بوجود می‌آید (به عنوان مثال Bad Sector) بعد از فع مشکل دیسک باید پایگاه داده را start کنیم و به دلیل اینکه از Online Redo Log File ها کپی تهیه نشده است باید به آخرین نسخه پشتیبان موجود (Backup) یعنی ساعت ۲ بامداد رجوع کنیم و از ساعت ۲ بامداد تا کنون هرچه اطلاعات وارد شده است باید دوباره وارد شود برای رفع این مشکل این امکان وجود دارد که هر باری که کار برروی Redo Log File جاری تمام شد یک کپی در آن تهیه کرده و برروی دیسک نگهداری کنیم تا در صورت خرابی نسخه پشتیان (Backup) ساعت ۲ بامداد را بازگردانده و با اعمال Archive Log ها به آخرین وضعیت قبل از خرابی برسیم.

DB می تواند در یکی از دو حالت (Mode) زیر قرار بگیرید:

۱) Archive Log

۲) No Archive Log

پیشنهاد می شود که در پایگاه داده سیستم های (OLAP) مانند DW (Data Warehouse) از No Archive Log و در سیستم های OLTP (OnLine Transaction Proceeding) از Archive Log استفاده کنید.

۵- SPFile یا PFile

این فایل را فایل راه انداز یا پیکربندی اوراکل می نامند در این فایل اطلاعاتی مانند:

- ✓ نام بانک اطلاعاتی
- ✓ مقدار حافظه تخصیص یافته به Instance
- ✓ نام و محل ذخیره سازی Control File ها
- ✓ تنظیمات و زبان هایی که پایگاه داده از آنها پشتیبانی می شوند
- ✓ و غیره

نگهداری می‌شود.

هنگامی که Instance راه اندازی می شود از این فایل جهت پیکربندی SGA (System|Shared Global Area) و راه اندازی Background Process ها استفاده می شود.

دو نوع فایل پارامتر وجود دارد:

- A-استاتیک (PFile) معمولا با فرمت Init<sid>.ora
- B-داینامیک (SPFile) معمولا با فرمت SPFile<sid>.ora

A یک فایل متنی است که می توان با یک ویرایشگر آن را باز کرده و تغییر داد ولی B-یک فایل باینری است که برای تغییر آن باید از دستور Alter system استفاده کرد.

با دستورات ذیل می‌توان از فایل pfile یک spfile ساخت و یا از فایل spfile یک pfile ساخت

```
SQL> create spfile from pfile;
SQL> create pfile from spfile;
```

اوراکل در زمان start پایگاه داده به دنبال SPFile می گردد و اگر پیدا نکرد به دنبال pfile می گردد.

می توانید از SPFile ، PFile درست کنید و تغییرات را در ان اعمال کنید سپس پایگاه داده را با PFile start کرده و بعد از اطمینان از صحت و عملکرد تغییرات مجدد از Pfile یک SPFile بسازید ، در دفعه بعدی که پایگاه داده را start می‌کنید اوراکل بصورت اتوماتیک از SPFile فوق استفاده خواهد کرد.

٦- Alert Log File

نام این فایل Alert_<SID>.log است. با استفاده از این فایل می توان دریافت که چه اتفاقاتی در پایگاه داده رخ داده است ، اطلاعاتی که در فایل فوق نگهداری میشود شامل موارد ذیل می‌باشد:

- ✓ اطلاعات مربوط به روشن(startup) و خاموش(shutdown) شدن پایگاه داده
- ✓ تغییرات (changes) ، رویدادها (events) و خطاهای (errors) رخ داده در پایگاه داده
- ✓ خطاهای مربوط به DF (Data File) ها
- ✓ Dead Locks ها
- ✓ تغییرات که در ساختار فیزیکی پایگاه داده انجام شده مانند ایجاد یا تغییر نام DF (Data File) ها و یا Redo Log فایلها
- ✓ تمامی دستورات Alter system انجام شده
- ✓ تمامی دستورات Log Switch و Check Point های انجام شده
- ✓ و غیره

برای تعیین مسیر فایل Alert Log سه روش وجود دارد:

۱) مشخص کردن ADR (Automatic Diagnostic Repository): توسط پارامتر Diagnostic_Desc که در اوراکل ۱۱g وجود دارد که از فایل Alert Log فایلهایی با فرمت های Text و Xml ایجاد می کند.

۲) Set کردن پارامتر Background_dump_dest

۳) مسیر پیش فرض oracle_home/rdbms

۷- Trace Files

هرگاه خطایی اتفاق می افتد جزئیات خطا در یک فایل Trace ذخیره می گردد مسیر آن همانند Alert Log مشخص می‌شود ، روش نامگذاری این فایلها به این صورت می باشد :

<SID>_<Process_Name>_<Process_ID>.trc

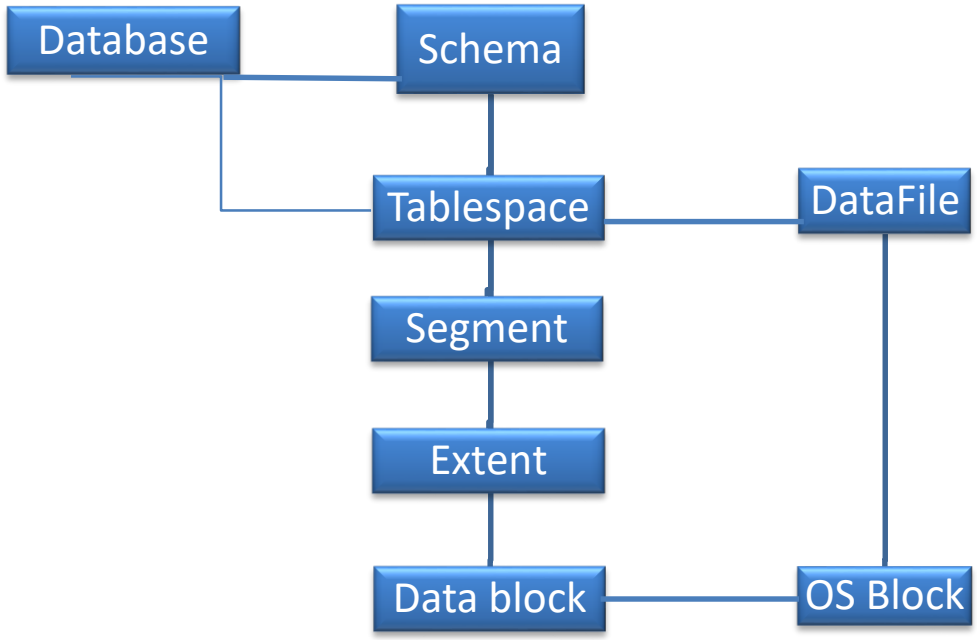
با استفاده از دستورات زیر می توان عملیات Trace را فعال یا غیر فعال کرد :

```
SQL> Alter system set trace_enable=false;
SQL> Alter system set trace_enable=true;
```

۸- Backup Files

از این فایلها برای بازیابی اطلاعات بانک استفاده می کنیم .

ساختار منطقی



۱) Data Block

کوچکترین واحد ساختار داده در اوراکل بوده و شامل مجموعه ای از بایت ها روی دیسک است.

مجموعه ای از بلاک های سیستم عامل توسط اوراکل فرمت شده و تبدیل به Oracle Black می شوند.

بلاک ها در واقع اساس و پایه سلسله مراتب بانک اطلاعاتی اوراکل هستند. در اوراکل اندازه DB (Data Block) پیش فرض هنگام نصب تعیین می گردد و بعد از نصب امکان تغییر اندازه آن وجود ندارد.

DB (Data Block) ها می تواند اندازه های ۲ و ۸ و ۱۶ و ۳۲ و ۶۴ داشته باشند. در اوراکل ۱۰ به بعد می توانید DF (Data File) هایی با اندازه بلاک های متفاوت داشته باشیم پارامتری ذیل مشخص کنند اندازه DB (Data Block) پیش فرض است.

DB_Block_Size

هر DB (Data Block) شامل سه بخش زیر است :

A. Header => Lock Information
B. Free Space => Update or Insert new row
C. Row data => table data or index or …

۲) Extent

شامل دو یا چندین DB (Data Block) پشت سرهم می باشد. در واقع مجموعه ای از چندین DB (Data Block) پشت سرهم است ، هنگامی که یک Object مانند جدول یا ایندکس ایجاد می کنید Initial Extent تخصیص داده می شود و Next extent وقتی تخصیص می یابد که می خواهید فضای Object را گسترش دهید.

۳) Segment

به هریک از اشیای اوراکل که به هنگام ایجاد نیاز به فضای دیسک دارند یک Segment گفته می‌شود که شامل مجموعه ای از Extent ها می باشد ، Segment ها شامل :

۱- Table یا Data Segment
۲- Partitioned Tables
۳- Cluster
٤- Index
٥- Partitioned Index
٦- Temporary Segments
  ۱- Order by
  ۲- Group by
  ۳- Materialized View
۷- Undo Segment
۸- LOB Segment
  ۱- LOB Index
  ۲- LOB Partition

هر Segment در واقع مجموعه ای از Extent ها می باشد که همگی متعلق به یک Object هستند.

٤) TableSpace:

از نظر منطقی هر TS (TableSpace) شامل یک یا چندین Segment می باشد ولی از لحاظ فیزیکی هر TS شامل یک یا چندین DF (Data File) است.

مزای استفاده از TS ها :

۱- مشخص کردن تنظیمات اولیه و رشد به مجموعه از DF (Data File) ها
۲- امکان پخش کردن Data بر روی دیسک های مختلف
۳- امکان تهیه Backup و Recovery راحت‌تر

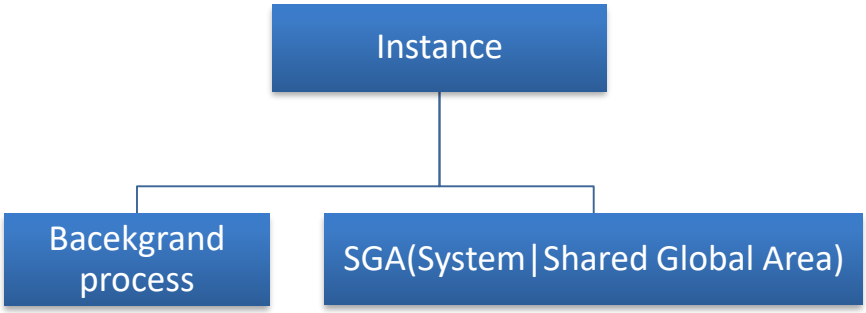
بعد از نصب اوراکل پنج ، TS (TableSpace) تعریف می شود :

۱- System (اجباری)
۲- Undo
۳- sysAux (کمکی)
٤- users
٥- Temp

دسته بندی TS (TableSpace) ها از نظر نوع عملکرد :

۱- Permanent
۲- Undo
۳- Temp

معماری اوراکل در حافظه (ساختار Instance بانک اطلاعاتی اوراکل) شامل موارد ذیل است :



اوراکل از مکانیزم های cache برای بهبود کارایی حافظه و پرهیز از مراجعه مکرر به دیسک استفاده می کند در اوراکل از ۲ نوع ساختار حافظه مورد استفاده قرار می‌گیرد.

۱- SGA (حافظه مشترک)
۲- PGA (حافظه مربوط به هر کاربر)

SGA شامل اجزاء ذیل می‌باشد:

۱- DBBC (DataBase Buffer Cache) (اجباری)
۲- Redo Log Buffer (اجباری)
۳- Shared Pool (اجباری)
   a. Library Cache
      i. SQL Area
      ii. PL/SQL Area
   b. Data Dictionary
   c. Result Cache
      i. Query
      ii. PL/SQL Functions
٤- Java Pool (اختیاری)

٥- Large Pool (اختیاری)
٦- Stream Pool (اختیاری)

DBBC

هرگاه دستورSQL (Select یا DML) نوشته می شود اطلاعات مورد نیاز آن دستور از DF (Data File) ها خوانده شده و در داخل DBBC قرار داده می شود و هر در مواقع مشخص توسط DBWRn بر روی دیسک نوشته می شود.

از این کامپونت برای بهبود سرعت اجرای SQL ها استفاده می شود ، وقتی یک دستور SQL می نویسید ابتدا DB (DataBase) به دنبال اطلاعات مورد نیاز در DBBC می گردد

اگر اطلاعات در DBBC وجود داشت آن را بازگردانده و کار مورد نیاز را انجام می دهد.

اگر اطلاعات در DBBC وجود نداشت به دیسک مراجعه کرده و یک کپی از آن را در DBBC قرار داده و کار مربوط به دستور SQL صادر شده را انجام می‌دهد

وضعیت بافرهای DBBC همواره یکی از موارد ذیل خواهد بود:

- Free or Unused
- Dirty: بلاک‌هایی که داده‌های آنها تغییر کرده ولی هنوز به دیسک منتقل نشده است و امکان استفاده از آنها نیز وجود ندارد
- Clean
- Pinned: بلاک هایی که در حال کار هستند

هر وقت DBBC به محدودیت فضا برخورد کند DBWRn شروع به کار کرده و بلاک های Dirty را براساس الگوریتم LRU (Last Recent Used) برروی دیسک می نویسید.
تعیین مقدار مناسب برای DBBC نقش مهمی در کارایی بانک اطلاعاتی دارد و هر چه مقدار I/O (رجوع به دیسک)کمتر باشد سرعت پایگاه داده بیشتر خواهد بود.

تنظیم DBBC از طریق پارامتر زیر انجام می‌شود:

DB_Block_Buffers DB_Cache_Size

استفاده از چندین buffer
اوراکل این امکان را دارد که شما سه نوع buffer ایجاد کنید

۱- Keep Buffer Pool

برای جداول کوچکی که رجوع آنها زیاد است استفاده می‌شود ، اندازی آن توسط پارامتر DB_Keep_Cache_Size تعیین می‌شود

۲- Recycle Buffer Pool

جداول بزرگی که ندرت استفاده می شود و سریع باید از DBBC حذف شوند ، اندازه آن توسط پارامتر DB_Recycle_cache_Size تعیین می‌شود

۳- Default Buffer Pool

برای اشیایی که نمی دانیم به کدام یک از ۲ حالت قبلی نیاز دارند که از الگوریتم LRU برای حذف آنها از DBBC استفاده می کند به هنگام ایجاد جدول یا Index می تواند از

Buffer_Pool_Keep یا Buffer_Pool_Recycle

استفاده نمایید.

در اوراکل این امکان این وجود دارد که بغیر از مشخص کردن بلاک پیش فرض ، TS (TableSpace) هایی با بلاک سایزهای متنوع داشته باشیم که برای هر کدام باید DBBC متناسب با تعریف کنیم:

DB_Keep_Cache_Size=10MB
DB_Recycle_Cache_Size=50MB
DB_Cache_Size=500MB
DB_2K_Cache_Size=200MB
DB_16K_Cache_Size=250MB

Redo Log Buffer

هرگاه داده ای تغییر کند تغییرات در Redo Log Buffer قرار داده شده و در زمان های مشخص LGWR (LoG WRiter) شروع به کار کرده و محتویات Redo Log Buffer را بر روی Online Redo Log File می نویسد اوراکل از Redo Log Buffer برای کاهش تعداد دفعات نوشتن بر روی Online Redo Log فایلها و از Online Redo Log فایلها برای Instance Recovery استفاده می‌کند

Shared Pool

مهم ترین بخش حافظه SGA است ، وظیفه آن نگهداری دستورات Parse شده ، DD (Data Dictionary) ها و همچنین نگهداری خروجی پرس و جوها است ، با استفاده از این کامپوننت می توانید از مکانیزم Cache استفاده کنید ، Shared Pool شامل ۳ قسمت است :

۱- Library Oracle
۱. SQL Query Area
۲. PL/SQL Area

۳. Soft|Hard Parse (در این قسمت دستورات Parse شده مربوط به SQL و PL/SQL نگهداری می شود)

۲- Data Dictionary Cache(اطلاعات cache شده مربوط به اشیاء ، مجوزها ، و غیره در این قسمت نگهداری می‌شود)

۳- Result Cache: در اوراکل ۱۱ اضافه شده و به اوراکل اجازه می دهد که نتایج پرس و جوهای خاص را ذخیره کرد ، این بخش دارای دو قسمت است :

۱- SQL Query Result Cache
۲- PL/SQL Function Result Cache

بصورت پیش فرض این قابلیت در اوراکل غیر فعال است و با استفاده از پارامترهای ذیل می توانید آن را فعال کنید

Result_Cache_Mode
Result_Cache_Size

نکته : زمانی که در Shared Pool فضای کافی برای قرار دادن کدهای Parse وجود نداشته باشد از الگوریتم LRU (Last Recent Used) برای خالی کردن این فضا استفاده می کند.
توسط پارامتر Shared_Pool_Size می‌توانید اندازه فضای Shared Pool را مشخص کنید

Large Pool

یک کامپونت اختیاری می باشد و اگر تعریف نشده باشد اوراکل از Shared Pool به جای آن استفاده می کند. کار برد آن در موارد ذیل است

۱- اجرای همزمان (Parallel) پرس و جوها
۲- انجام Backup و Recovery

از طریق پارامتر Large_Pool_Size می‌توانید مقدار آن را مشخص کنید.

Java Pool

اندازه آن توسط پارامتر Java_Pool_Size مشخص می شود و زمانی کاربرد دارد که اوراکل از کدهای Java استفاده کرده باشد.

Stream Pool

فرض کنید که چندین پایگاه داده دارید و می خواهید اطلاعات آنها را باهم Sync نگه دارید به این صورت که اگر در یکی از پایگاه داده‌ها تغییری رخ دهد تغییرات در پایگاه داده‌های دیگر اعمال شود ، اوراکل برای این کاراز تکنولوژی Stream استفاده می کند. اندازه آن توسط پارامتر Stream_Pool_Size مشخص می شود

PGA شامل بخش های ذیل می‌باشد:

به ازای هر session اوراکل یک حافظه PGA تخصیص داده می‌شود و در صورتی که تنظیمات سرور اوراکل با گزینه dedicated مشخص شده باشد شامل بخش های ذیل می باشد

۱. UGA (User Global Area)
   که شامل دسترسی کاربران و اطلاعات آماری مربوط به session است
۲. Private SQL Area
   شامل متغیرها و اطلاعات مورد نیاز جهت اجرای SQL است
۳. Sort Area
   شامل فضایی جهت مرتبط سازی داده ها است
۴. Stack Space
   سایر اطلاعات از جمله متغیرها و آرایه ها می باشد

PGA نقش زیادی برای بهبود کارایی سیستم دارد به عنوان مثال عملیات مرتب سازی باید در حافظه انجام شود و اگر نتواند آن را در حافظه انجام دهد اطلاعات را در Temp Tablespace قرارداده و مرتب سازی را در آنجا انجام می‌دهد لذا تنظیم درست PGA بسیار مهم است، پارامترهایی که برای تنظیم PGA در اختیار هستند شامل موارد ذیل می‌باشند.

Hash_Area_Size
Sort_Area_Size

Background Process های اجباری :

۱- DBWRn (Data Base WRiter):

وظیفه اصلی آن نوشتن Dirty Buffer های موجود در DBBC (Data Base Buffer Chache) بر روی دیسک است ، بطور کلی هرگاه DBBC (Data Base Buffer Chache) فضای خالی برای نگهداری اطلاعات نداشته باشد DBWRn (Data Base WRiter) فعال می شود و بر اساس الگوریتم LRU ، Dirty Buffer ها ها را بر روی دیسک می نویسد DBWRn (Data Base WRiter) در یکی از سه حالت ذیل فعال می شود:

۱) Checkpoint
۲) در DBBC بافر clean یافت نشود
۳) هر ۳ ثانیه

می توان تا تعداد ۲۰ پردازش برای DBWRn تعریف کرد. تعداد DBWRn توسط پارامتر ذیل تعریف می گردد.

DB_Write_Process

۲- LGWR (LoG Writer):

در یکی از حالتهای ذیل فعال می‌شود

۱) هر سه ثانیه
۲) وقتی ۱/۳ فضای Redo Log Buffer پر شد
۳) به ازای هر باری که دستور commit صادر می‌شود
۴) قبل از صدا زدن DBWR

هرگاه تغییری در پایگاه داده انجام شود تغییرات در Redo Log Buffer ثبت می‌شود و طبق زمان بندی مشخصی که در بالا اعلام شد تغییرات را به Redo Log File منتقل می کند در Redo Log

Buffer داده‌های commit شده و commit نشده وجود دارد ، با استفاده از این کامپوننت (LGWR) قابلیت های ذیل فراهم خواهد شد :

۱) رجوع به Redo log File برای نوشتن کمتر خواهد بود .

۲) اوراکل تنظیم می کند که هیچ داده ای از بین نخواهد رفت و می‌تواند برای Instance Recovery از داده‌های موجود در LGWR استفاده می شود.

۳- SMON :

وظایف :

۱) Instance Recovery

۲) ادغام extent های خالی

۳) حذف Temporary Segments

۴- PMON :

وظیفه پاکسازی پردازشهای ناتمام را برعهده دارد که شامل موارد ذیل است:

۱) Rollback

۲) Release Locks

۳) Release Resource

۵- CKPT :

توسط Checkpoint مدت زمان لازم برای Instance Recovery کاهش پیدا می‌کند ، بعد از صدا شدن Checkpoint عملیات ذیل اتفاق می افتد:

۱) Call DBWR (Dirty Buffer ها را از DBBC بر روی Data File ها می‌نویسد)

۲) Call LGWR (محتویات Redo Log Buffer را بر روی Redo Log File می‌نویسد)

۳) یک رکورد CKPT در فایل Redo Log File ثبت می شود.

۴) اطلاعات موجود در header ، Data File ها و CF (Control File) ها را همسان (sync) می‌کند

در مواقع زیر عملیات CKPT اتفاق می‌افتد

۱) Shutdown (اتوماتیک)

۲) Switch Log File (اتوماتیک)

۳) Alter system checkpoint (دستی)

Background Process های اختیاری:

۱- ARCn :

LGWR (LoG WRiter) اطلاعات موجود در Redo Log Buffer را بر روی Redo Log File می نویسد ، وقتی که Redo Log File جاری پر شد به سراغ Redo Log File بعدی می رود و تا بر روی آن بنویسد و بصورت چرخشی این کار را انجام می دهد

اگر پایگاه داده در حالت Archive Log باشد بعد از اینکه Redo Log File جاری پر شد و قبل از اینکه LGWR (LoG WRiter) شروع به نوشتن بر روی فایل بعدی بکند یک کپی از آن گرفته شده و در مسیری که مشخص شده است کپی می‌شود ، با استفاده از این قابلیت امکان Recovery به هر نقطه زمانی وجود دارد.

اگر تعداد تراکنش های پایگاه داده زیاد باشد و پایگاه داده در حالت Archive Log باشد ممکن است کارایی سیستم به دلیل اینکه کپی فایل قبلی تمام نشده است کاهش پیدا کند ، بطور کلی امکان بازنویسی در فایل Redo Log File بدون آنکه در آن کپی تهیه شده باشد وجود ندارد ، برای این مشکل سه راه حل وجود دارد :

۱) افزایش Size ، Redo Log File

۲) افزایش تعداد فایل های Redo Log File ها

۱) افزایش تعداد پردازش های ARCn از طریق پارامتر Log_Archive_Max_Processes

۲- MMAN :

هر یک ساعت یکبار فعال شده و اطلاعات آماری را از SGA خوانده و در AWR (Automatic Work Repository) قرار می دهد جمع آوری آمار و نوشتن بر روی snapshot می گویند.

هر وقت یک snapshot اتفاق می افتد فرآیند دیگری به نام ADDM (Automatic Database Diagnostic Manager) فعال شده و دو snapshot جاری را باهم مقایسه کرده و پیشنهاداتی جهت بهبود کارایی ارائه می دهد.

مدیریت خودکار حافظه اوراکل AMM (Automatic Memory Management)

از اوراکل ۹ به بعد این امکان فراهم شده که مدیریت حافظه PGA توسط اوراکل انجام شود ، در اوراکل ۱۰ نیز این امکان فراهم شد که مدیریت SGA توسط اوراکل انجام شود و در اوراکل ۱۱ می‌توان مدیریت دو حافظه PGA و SGA را توسط اوراکل انجام داد.

مدیریت خودکار PGA

به ازای هر دستور SQL یک حافظه غیر مشترک به session جاری تخصیص داده می شود که عملیاتی مثل مقدار دهی متغیرها، مرتب سازی خروجی ، گروه بندی داده ها و غیره در آن انجام می شود ، در حالت مدیریت دستی PGA حافظه ای که به PGA تخصیص داده می شود تا انتهای فعالیت session جاری رزرو می‌شود ولی اگر AMM را برای PGA راه اندازی کنید در صورتی که به حافظه فوق نیازی نباشد حافظه تخصیص یافته را در اختیار سیستم قرار خواهد داد

برای تنظیم AMM در PGA پارامترهای ذیل وجود دارد :

- Work_area_size_policy
- Pga_aggreate_target

مدیریت خودکار SGA

برای مدیریت خودکار حافظه SGA باید مقادیر پارامترهای زیر را به صفر و یا مقدار پیش فرض تغییر دهید پارامترها شامل

DB_Cache_Size
Shared_Pool_Size
Large_Pool_Size
Java_Pool_Size
Stream_Pool_Size

0 | Default Value

همچنین پارامتر sga_max_size باید مقداری بزرگتر از صفر داشته باشد و sga_target نیز باید مقداری بزرگتر از صفر داشته باشد و همواره باید SGA_derget <= SGA_Max_Size باشد.

AMM در SGA و PGA

هر دو حافظه (SGA و PGA) بصورت اتوماتیک توسط خود اوراکل مقدار دهی می شود برای این کار باید پارامتر Memory_Target با مقداری بزرگتر از صفر مقدار دهی شده باشد و همواره باید

Memory_Target <= Memory_Max_Size

بصورت خلاصه برای مدیریت خودکار SGA و PGA باید :

۱- Memory_Target > ۰ باشد

۲- پارامترهای اشاره شده در دو قسمت قبلی یا صفر باشد یا مقدار پیش فرض داشته باشند

View های موجود در اوراکل

به بخش ذیل تقسیم می‌شوند

۱- Static

a. View هایی که با _DBA شروع می‌شود

b. View هایی که با _ALL شروع می‌شود

c. View هایی که با _USER شروع می‌شود

۲- Dynamic

View هایی که با v$ شروع می‌شوند ، بعد از راه اندازی instance امکان استفاده از این view ها وجود دارد ، از جمله این view های می‌توان به v$session ، v$database و غیره اشاره کرد

SCN (System Change Number):

هر تغییری که در اوراکل بوجود می‌آید مقدار SCN تغییر می‌کند و می‌توان از آن برای Recovery بانک اطلاعاتی استفاده کرد.