ORACLE 10g

Database Administration

PART 1

مولف و مطالب از :
Encoder

لغت نامه (کتاب شماره ۱)

آموزش اوراکل

10g

ASHIYANE.ORG

CERTIFICATED

ABORT

ABORT یکی از حالتهایShutdown بانک اطلاعاتی می باشد. انتخاب ABORT زمانی انجام می شود که بانک اطلاعاتی با هیچکدام از حالتهای دیگر Shutdown نشود، Shutdown کردن بانک اطلاعاتی به حالت ABORT ممکن است به بانک اطلاعاتی آسیب برساند. در این حالت Instance بدون هیچ انتظار برای ثبت اطلاعات ناحیه SGA در فایلهای داده ای بانک اطلاعاتی و Redo Log File ها باعث Shutdown شدن می گردد.

حالت ABORT نیز همانندIMMEDIATE باعث قطع ارتباط تمامی کاربران با بانک اطلاعاتی می گردد اما باعث اعمال عملRollback نمی گردد. عمل Rollback درStartup بعدی انجام می شود و باعث کند شدن عملStart می گردد. دستور Shutdown بانک اطلاعاتی به حالت ABORT به صورت زیر است.

```
SHUTDOWN ABORT;
```

Account

در بانک اطلاعاتی User Account (شناسه کاربر) تنها یک ساختار فیزیکی محسوب نمی شود، بلکه ایجاد روابط مهم و اساسی بین اشیاء در بانک اطلاعاتی را نیز بر عهده داشته و صاحب آن اشیاء نیز می باشد. مهمترین کاربری که در هر بانک اطلاعاتی ORACLE به صورت پیش فرض به وجود می آیدکاربرSYS می باشد، که جداول Data Dictionary را در اختیار دارد. اطلاعات ذخیره شده در Data Dictionary درباره تمامی اشیاء بانک اطلاعاتی است. کاربر SYSTEM دارای View هایی است که به جداول Data Dictionary دسترسی دارد تا توسط کاربران در بانک اطلاعاتی مورد استفاده قرار گیرد. زمانیکه اشیایی در بانک اطلاعاتی ایجاد می شوند، این اشیاء تحت نظارتUser Accountساخته می شوند. می توان در زمان ایجاد یک User Account یک Tablespace پیش فرض در نظرگرفت تا تمامی اشیاء آن Account در آن فضا ایجاد شود.

همچنین می توان Account های بانک اطلاعاتی را به عنوان Account های سیستم عامل تعریف نموده و به کاربران اجازه داد تا بدون داشتن کلمات عبور ورودی، به بانک دسترسی داشته تا بتواند به اشیائی که دارند و یا به اشیائی که اجازهٔ دستیابی به آنها را پیدا کرده اند، دسترسی یابند.

Alert Log

مشکلات جدی که پرداشهای پیش زمینه را تحت تاثیر قرار می دهند، معمولاً در Alert Log بانک

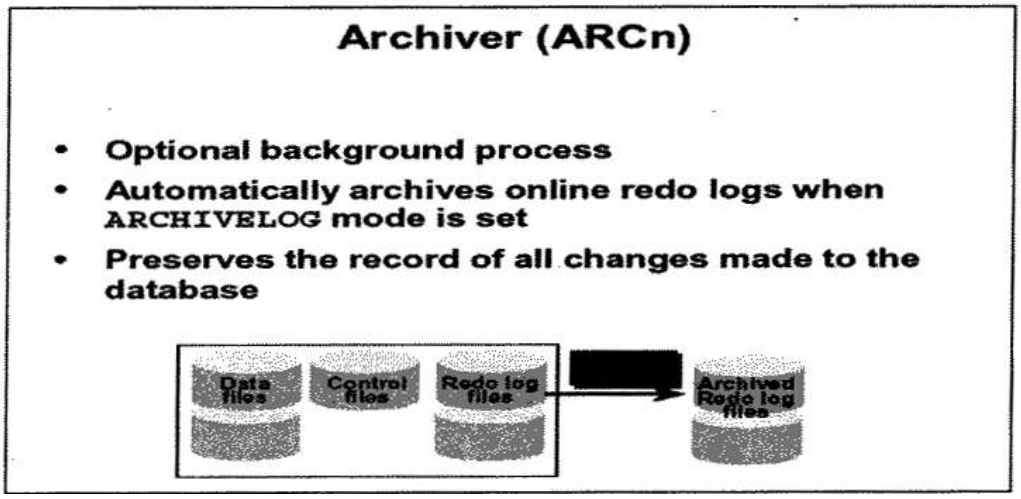
اطلاعاتی ثبت می شوند. هر یک از پردازشهای پیش زمینه در یک Instance با یک فایل ردیابی (Trace File) مرتبط به خود، اجرا می شوند. این فایل ردیابی شامل اطلاعاتی در زمینه وقایع خاصی است که پردازش پیش زمینه با آن مواجه می شود. ORACLE علاوه بر فایلهای ردیابی فایل دیگری به نام Alert Log دارد که دستورات و نتایج مربوط به وقایع بانک اطلاعاتی را ثبت می کند. به عنوان مثال، ایجاد Tablespace، سوئیچهای Redo Log، اعمال Recovery و Startup های بانک اطلاعاتی در Alert Log ثبت می شوند. فایل Alert Log یک منبع مهم اطلاعاتی، برای اداره روزانه بانک محسوب می گردد. فایلهای ردیابی نیز در زمان مشخص کردن علت یک خرابی ویا خطای سیستم بسیار مفید واقع می شوند.

Alert Log معمولاًدر دایرکتوریBACKGROUND_DUMP_DEST مشخص شده است که عمدتاً این دایرکتوری، دایرکتوریORACLE-BASE /admin/INSTANCE_NAME/bdump می باشد.

ARCH

پردازش پیش زمینه LGWR به روش چرخه ای در فایلهای Online Redo Log می نویسد. بدین ترتیب که بعد از پر کردن اولین Log File، شروع به نوشتن دومی می کند تا آن را هم پر کند، سپس نوشتن در سومینLog File را آغاز می کند. زمانی که آخرین فایل Online Redo Log پر شود، LGWR شروع به روی هم نوشتن محتویات اولین Redo Log File می کند. هنگامی که حالت ARCHIVE LOG در بانک اطلاعاتی انتخاب شود، بانک اطلاعاتی قبل از باز نویسی مجدد داده ها در Log File ها، نسخه برداری از هر فایل Redo Log را انجام می دهد.

این فایلهای Redo Log بایگانی شده معمولاً در یک دیسک ثبت می شوند. البته این امکان وجود دارد که مستقیماً در یک Tape نیز نوشته شوند. عمل بایگانی توسط پردازشهای پیش زمینه ARCH اجرا می شود. بانک اطلاعاتی که در حالت Archive Log Mode قرار داشته باشند امکان انجام Hot Backup را دارند.



ARCHIVE LOG

اگر بانک اطلاعاتی در وضعیت ARCHIVE LOG قرار داشته باشد، اطلاعات Online Redo Log File ها بعد از پر شدن در مقصد مشخص شده، کپی می شوند. پردازش زمینه ARCH به صورت خودکار عمل کپی اطلاعات را به فایلهای ARCHIVE LOG به عهده دارد.

قرار دادن بانک اطلاعاتی در وضعیت ARCHIVE LOG درگیری بیشتری برای مدیر بانک اطلاعاتی به همراه دارد اما باعث فراهم شدن لایه های محافظتی بیشترمی گردد. برای تغییر وضعیت بانک اطلاعاتی از حالت پیش فرض (NO ARCHIVE LOG) به حالت Archive Log مراحل زیر باید انجام شود:

۱. بانک اطلاعاتی را ابتدا Shutdown و سپس با دستور STARTUP MOUNT، Instance را MOUNT کنید.
۲. دستور ALTER DATABASE ARCHIVE LOG را برای قرار دادن بانک اطلاعاتی در حالت ARCHIVE LOG اجرا کنید.

همچنین برای قرار دادن بانک اطلاعاتی در حالت ARCHIVE LOG پارامترهای زیر از فایل INIT.ORA را از حالت پیش فرض به مقدار مشخص شده تغییر دهید:

پارامتر	شرح
LOG_ARCHIVE_START = TRUE	قرار دادن بانک اطلاعاتی در حالت ARCHIVE LOG
LOG_ARCHIVE_DEST = نام و مسیر فایل	تعیین نام و مسیر فایلهای ARCHIVE
LOG_ARCHIVE_FORMAT = فرمت فایل	فرمت فایلهای ARCHIVE

در وضعیت ARCHIVE LOG :

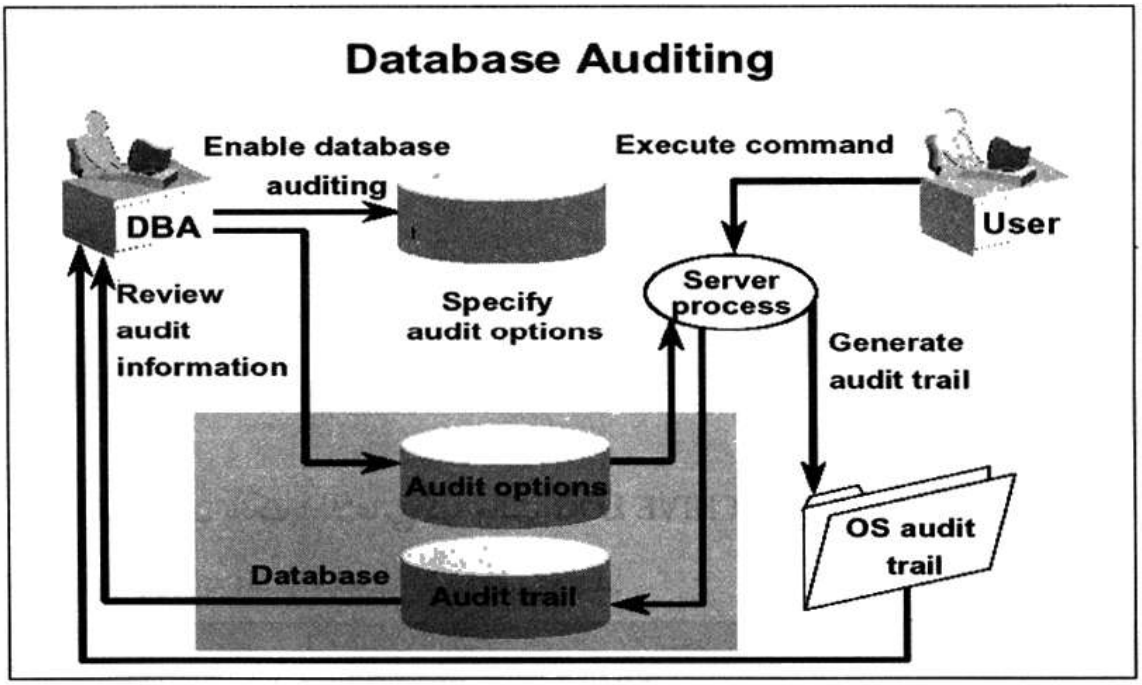
- فضای بیشتری بر روی دیسک مورد نیاز است.
- امکان انجام HOT BACKUP وجود دارد.
- در صورت وقوع Media Failure امکان Recovery کامل بانک اطلاعاتی وجود دارد.

AUDIT

بانک اطلاعاتی قادر به رسیدگی و بررسی تمامی اعمالی است که در آن اتفاق می افتد. به این عمل Audit اطلاق می گردد. تمامی این اعمال به صورت رکوردهایی در بانک اطلاعاتی ثبت می شوند رکوردهای Audit ممکن است در جدول SYS.AUD$ و یا در Operating System Trail (ردیابی بررسی

سیستم عامل) نوشته شوند. توانایی استفاده از ردیابی بررسی سیستم عامل، به سیستم عامل بستگی دارد. در بانک اطلاعاتی اراکل امکان Audit اعمال زیر وجود دارد :

- برقراری ارتباط با سیستم
- دسترسی به اشیاء
- اعمال بانک اطلاعاتی



Background process

روابط بین ساختارهای حافظه و ساختارهای فیزیکی بانک اطلاعاتی توسط پردازشهای پیش زمینه حفظ و اجرا می شوند. تعداد پردازشهای پیش زمینه بانک اطلاعاتی تا حدود زیادی به پیکربندی بانک اطلاعاتی بستگی دارد. این پردازشها عبارتند از:

- SMON
- PMON
- DBWR
- LGWR
- RECO
- ARCH

Backup

سه روش برای Backup از یک بانک اطلاعاتی ORACLE وجود دارد:

۱. Export
۲. Backup های Offline
۳. Backup های Online (ARCHIVELOG)

هر Export، یک Backup منطقی از بانک اطلاعاتی است و Backup های Offline وOnline به عنوان Backup های فیزیکی بانک اطلاعاتی می باشند. Backup های بانک اطلاعاتی اراکل به دو گروه زیر تقسیم می شوند:

الف: Backup های منطقی

در این نوعBackup به جای کپی از ساختارهای فیزیکی، از محتوای درون بلوکهای داده ای برای انجام Backup استفاده می شود. بنابراین تعاریف ایجاد تمامی اشیاء بانک اطلاعاتی به صورت دستورات DDL(Data Definition Language) در این نوع Backup ذخیره می شوند. این نوع Backup تنها می تواند از محتوی بلوکهای داده ای Data File های بانک Backup بگیرد و در واقع نمی توان از Control File ها و Redo Log File ها برای Backup های منطقی استفاده کرد.

ب: Backup های فیزیکی

به Backup ای گفته می شود که از ساختارهای فیزیکی بانک اطلاعاتی مانند Data Fileها، Redo Log File ها، Archive Log Fileها و Control Fileها انجام می شود. این نوع Backup ها به دو گروه زیر تقسیم می شوند :

۱. **Cold Backup**

به Backup ای گفته می شود که در زمان Shutdown بودن بانک اطلاعاتی از ساختارهای فیزیکی آن انجام می شود. چون در این نوع Backup بانک اطلاعاتی در حالت Shutdown قرار دارد، امکان تغییر محتوی بلوکهای داده ای Datafile های بانک اطلاعاتی وجود نداشته بنابراین می توان با استفاده از فرمان کپی سیستم عامل از ساختارهای فیزیکی بانک، کپی تهیه کرد. بانک اطلاعاتی می تواند به حالتهای Normal، Immediate و یا Transaction در حالت Shutdown قرار بگیرد.

قرار دادن بانک اطلاعاتی در حالت Shutdown باعث می شود که تمامی تغییرات اعمال شده با آخرین وضعیت ممکن ثبت شوند. در واقع هیچ تغییری در بانک اطلاعاتی وجود ندارد که

وضعیت آن مشخص نشده باشد. بنابراین در صورت نیاز به Restore کردن فایلهایBackup به Media Recovery نیاز نمی باشد. اما اگر زمان زیادی از انجام Backup گذشته باشد، نیاز به Media Recovery بعد از Restore کردن فایلهای Backup وجود دارد.

۲. **Hot Backup**

به Backup ای گفته می شود که در زمان Start بودن بانک اطلاعاتی از ساختارهای فیزیکی آن انجام می شود. در این نوع Backup بعلت Open بودن بانک اطلاعاتی امکان تغییر محتوی بلوکهای داده ای بانک توسط کاربران وجود دارد. بنابراین Hot Backup تنها رکوردهای Commit شده در بلوکهای داده ای را در نظر می گیرد.

برای انجام Hot Backup در یک بانک اطلاعاتی، نیاز به قرار دادن بانک در حالت Archive Log File می باشد. بنابراین اگر بانک اطلاعاتی در حالت Archive Log File قرار نداشته باشد تنها انجام Cold Backup از بانک اطلاعاتی امکان پذیر است و امکان انجام Hot Backup وجود ندارد.

Backup piece

Backup Piece شامل یک ساختار فیزیکی است که می تواند یک یا بیش از یک Datafile و یا Archive Log File را شامل شود. اندازه Backup Piece توسط عبارت MAXPIECESIZE در دستورات CONFIGURE CHANNEL و یا ALLOCATE CHANNEL قابل تنظیم می باشد.

Backup set

یک Backup Set شامل یک یا چند Datafile، Control File و Archive Log File بانک اطلاعاتی است که می تواند به دو یا چند فایل تقسیم شود. Backup Set یک ساختار منطقی با مشخصه های زیر است :

- شامل یک یا بیش از یک فایل فیزیکی است که Backup Piece گفته می شود.
- از طریق دستورBACKUP ایجاد می شوند و پارامترFILESPERSET تعداد Datafile های یک Backup Set را مشخص می کند.
- Backup Set می تواند بر روی دیسک و یا Tape نوشته شود.
- دستورRESTORE قبل از Recovery بانک ابتدا باید فایلهای موجود درBackup Set

را به دست آورده و استخراج کند.

- Archive Log File ها به صورت Incremental قابل Backup نمی باشند.
  (همواره باید به صورت Full باشد)
- Backup Set هیچگاه شامل بلوکهای داده ای که استفاده نشده اند، نمی باشد.

هر Backup Set همواره دارای یک Backup Piece می باشد. هر Backup Piece شامل یک ساختار فیزیکی است که می تواند یک یا بیش از یک Datafile و یا Archive Log File را شامل شود. اندازه Backup Piece توسط عبارت MAXPIECESIZE در دستورات CONFIGURE CHANNEL و یا ALLOCATE CHANNEL قابل تنظیم می باشد.

```
ALLOCATE CHANNEL ... MAXPIECESIZE=integer
CONFIGURE CHANNEL ... MAXPIECESIZE=integer
```

مثال :

```
RMAN>RUN{
      ALLOCATE CHANNEL t1 TYPE 'SBT_TAPE'
      MAXPIECESIZE=4G;
      BACKUP TABLESPACE Users;
      }
```

Bitmap Index

نوع دیگری از Index که از ORACLE 8 به بعد قابل تعریف می باشد، Bitmap Index است. در Index های نرمال، اراکل به منظور پیمایش Index (برای یافتن بلوک برگ) از روش B*Tree استفاده می کند. در یک Bitmap Index، یک Bitmap از RowID ها نگهداری می شود. این Bitmap، نشان می دهد کدام سطرها با ستون Index شده، تناظر برقرار می کنند. اگر بیت 1 باشد، نشان دهنده این است که سطر متناظر با آن حاوی مقدار کلید است، و اگر بیت 0 باشد، خلاف این موضوع درست می باشد.

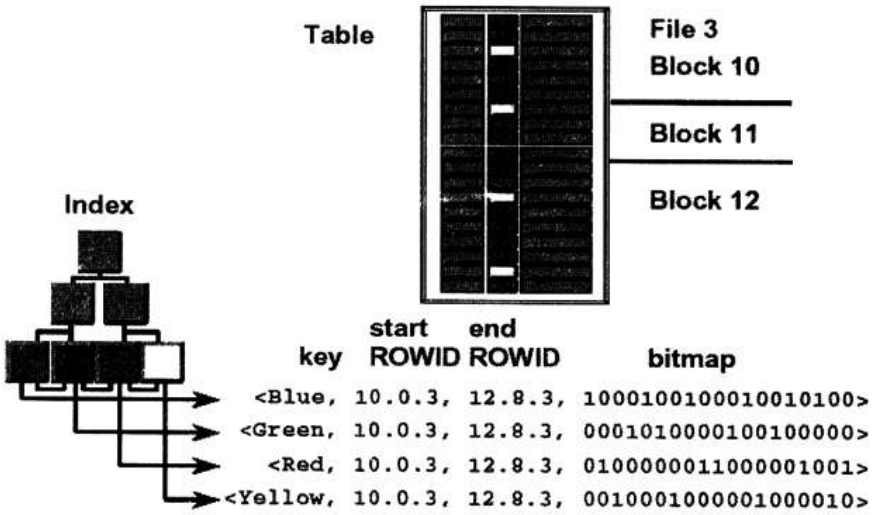
به سادگی می توان حدس زد که Index های Bitmap، احتمالاً می توانند تحت شرایط مناسب بسیار مفید باشند (و در شرایط مناسب چیزی جز سربار نباشند). مثالی از یک Bitmap Index در شکل زیر نشان داده شده است. در این شکل بر روی ستون deptno یک Bitmap Index ایجاد شده است.

استفاده از Bitmap Index، در خصوص عناصری بهتر است که تنوع آنها کم باشد. مثلاً ستونی که جنسیت را نشان می دهد تنوع کمی دارد، چرا که فقط دو مقدار (مرد و زن) برای آن امکان پذیر است.

empno	eName	deptno
7156	Scott	10
7192	Allen	20
532	Ford	30
3254	Smith	10
3955	King	20
1256	Jack	30

Bitmap on Index deptno 10	20	30
1	0	0
0	1	0
0	0	1
1	0	0
0	1	0
0	0	1

Bitmap Index



Check

هرگاه مقادیر یک ستون الزاماً ازیک بازه تعریف شده قابل انتخاب بوده و درخارج از آن بازه غیرمجاز از این Constraint استفاده می شود. اطلاعات مربوط به constraint های بانک اطلاعاتی در جدول USER_CONSTRAINTS از جداول Data Dictionary ذخیره می شوند.

CKPT

Checkpoint ها (ایستگاه های بازرسی) مدت زمان مورد نیاز برای اجرای Instance Recovery را کاهش می دهند و باعث می شوند DBWR تمامی بلوکهایی را که تغییر یافته از Data Block Buffer Cache به Data Fileها بر روی دیسک منتقل کند. بعد از انجام عمل DBWR و انتقال داده ها با موفقیت بر روی دیسک، Header های Data File ها و Online Redo Log File ها برای ثبت عملی که توسط Checkpoint اتفاق افتاده Update می شوند.

Checkpoint ها زمانی بصورت خودکار رخ می دهند که یک فایل Online Redo Log پر شود، این امکان وجود دارد که پارامتر LOG_CHECKPOINT_INTER در فایل Init.ora در Instance بانک اطلاعاتی برای اجرای یک Checkpoint که بیشتر مورد استفاده قرار می گیرد، استفاده شود.

Client/Server

معماری Client/Server متشکل از یک یا چند کامپیوتر سرویس گیرنده (Client) می باشد که برنامه کاربردی در آنها اجرا می شود. این کامپیوترها از طریق نوعی خط ارتباطی به کامپیوتر راه دور(Server) که پاسخگوی درخواست های سرویس گیرنده است متصل هستند.

در معماری Client/Server، سیستم مدیریت بانک اطلاعاتی رابطه ای (RDBMS) بر روی کامپیوتر Server قرار می گیرد. برنامه کاربردی واقع بر کامپیوتر Client با یک لایه نرم افزاری دیگر به نامMiddleware ارتباط برقرار می کند. این لایه مسئول برقراری ارتباط و تبادل درخواستها و نتایج آن میان برنامه کاربردی و RDBMS می باشد.

Cluster

Cluster ها که گاهی به آنها Cluster Index نیز گفته می شود، ساختاری اختیاری برای ذخیره جداول در یک بانک اطلاعاتی ORACLE است. در هر Cluster چند جدول مربوط به هم در کنار یکدیگر ذخیره می شوند و از این طریق، زمان دستیابی به اطلاعات مرتبط، بهبود می یابد. جداولی که یک ستون را به اشتراک می گذارند می توانند حول آن ستون Cluster شوند. بدین ترتیب، سرعت بازیابی سطرهایی که از روی ستون مذکور مورد دستیابی قرار می گیرند بیشتر می شود. وجود Cluster، از دید کاربران و برنامه های کاربردی پنهان است وآنها، تنها بر چگونگی ذخیره شدن داده ها اثر می گذارند.

باید توجه داشت که وقتی اراکل داده ها را از دیسک بازیابی می کند، عمل خواندن روی بلوکهای داده ای، ونه سطرها، انجام می شود. بنابراین، چنانچه داده های مرتبط در کنار یکدیگر ذخیره شده باشند، متفقاً (از دیسک) به بلوک داده های یکسانی در حافظه کپی خواهند شد. وقتی یک بلوک داده ای خوانده شود، عمل خواندن بر روی تمام داده های جداول Cluster شدهٔ موجود در آن بلوک داده ای صورت خواهد گرفت. در نتیجه، اگر در بیشتر مواقع بخواهیم داده های مرتبط به هم را از تمام جداول مورد درخواست قرار دهیم، استفاده از Cluster بسیار مفید خواهد بود.

در صورتی که دو جدول با داده های مربوط به هم داشته باشیم که مکرراً با هم مورد دستیابی قرار بگیرند،

Cluster کردن آنها می تواند باعث بهبود کارآیی شود. زیرا داده های مربوط به هم به اتفاق در SGA قرار می گیرند. از آنجا که داده های مزبور مکرراً با همدیگر مورد استفاده قرار می گیرند، وجود آنها در SGA (در هر زمانی که نیاز باشد) باعث کاهش زمان دستیابی و افزایش کارآیی می شود.

Cold Backup

به Backup ای گفته می شود که در زمان Shutdown بودن بانک اطلاعاتی از ساختارهای فیزیکی آن انجام می شود. چون در این نوع Backup بانک اطلاعاتی در حالت Shutdown قرار دارد، امکان تغییر محتوی بلوکهای داده ای Datafile های بانک اطلاعاتی وجود نداشته، بنابراین می توان با استفاده از فرمان کپی سیستم عامل از ساختارهای فیزیکی بانک، کپی تهیه کرد. بانک اطلاعاتی می تواند به حالتهای Normal، Immediate و یا Transaction در حالت Shutdown قرار بگیرد. قرار دادن بانک اطلاعاتی به هر کدام از حالتهای فوق در وضعیت Shutdown باعث می شود که تمامی تغییرات اعمال شده با آخرین وضعیت ممکن ثبت شوند. در واقع هیچ تغییری در بانک اطلاعاتی وجود ندارد که وضعیت آن مشخص نشده باشد. بنابراین در صورت نیاز به Restore کردن فایلهای Backup به Media Recovery نیاز نمی باشد. اما اگر زمان زیادی از انجام Backup گذشته باشد، نیاز به Media Recovery بعد از Restore کردن فایلهای Backup وجود دارد.

Commit

یکی از دستورات (TCL) Transaction Control Language بانک اطلاعاتی اراکل می باشد که برای اتمام یک Tranaction مورد استفاده قرار می گیرد. وضعیت داده ها بعد از commit به صورت زیر می باشد:

- تغییرات انجام شده در جداول ثبت دائم می گردد .
- امکان بازگرداندن تغییرات (داده های قبلی) وجود ندارد .
- تمامی کاربران امکان مشاهده تغییرات (داده های جدید) را دارند.
- Lock های انجام شده از روی رکوردها بر داشته می شوند.
- تمامی savepoint های تعریف شده حذف می شوند.

Constraint

برای اعمال محدودیت جهت درج مقادیر در یک جدول از Constraint استفاده می شود. Constraint

ها یکی دیگر از اشیاء بانک اطلاعاتی اراکل بوده و درData Dictionary در جدولی به نام USER_ CONSTRAINTS ذخیره می گردند. Constraint های بانک اطلاعاتی عبارتند از:

- PRIMARY KEY
- FOREIGN KEY
- CHECK
- UNIQUE
- NOT NULL

Context Areas

در ناحیه Shared SQL، دو محیط Public وPrivate وجود دارد. برای هر عبارت SQL که توسط یک کاربر صادر می شود نیاز به یک ناحیهٔ SQL اختصاصی است که تا زمان بستن شدن Cursor باید وجود داشته باشد. به این ناحیه Context Areas گفته می شود.

Control File

تمامی ساختارهای فیزیکی یک بانک اطلاعاتی توسط Control File های آن نگهداری می شود که اطلاعات مربوط به تمام فایلهای موجود در بانک اطلاعاتی را ثبت می کند. Control File باعث حفظ هماهنگی و پایداری در ساختار درونی شده و اعمال Recovery را هدایت می کند. هر بانک اطلاعاتی حداقل دارای یک Control File می باشد، اگر چه بیش از یک Control File برای افزایش امنیت در از دست دادن این فایل به صورت Mirror ایجاد می شود. Control File ها اطلاعات مربوط به نام بانک اطلاعاتی، تاریخ و زمان ایجاد آن، محل فیزیکی Data File ها و Redo Log File ها را شامل می باشند.

در زمان Start شدن بانک اطلاعاتی ابتدا فایل Init.ora خوانده شده و از طریق پارامتر Control_Files آن، نام و مسیر Conrol File های بانک اطلاعاتی مشخص می شود، اگر این فایل وجود نداشته باشد و یا محتویات درون آن با ساختارهای فیزیکی موجود مطابقت نداشته باشد، امکان Start شدن کامل بانک اطلاعاتی ممکن نمی شود. بنابراین حیات بانک اطلاعاتی به اطلاعات Control File های آن بستگی دارد.

از آنجایی که Control File ها برای بانک اطلاعاتی بسیار حائز اهمیت هستند، بطور پیوسته نسخه های متعددی از آنها ذخیره می شوند.

برای اینکه صدمات ناشی از خرابی یا خطای دیسک را به حداقل برسانند، عمدتاً بر روی دیسکهای مجزا ذخیره می شوند. بانک اطلاعاتی وظیفه ایجاد و نگهداری Control File ها را بر عهده دارد. نام Control File های بانک اطلاعاتی توسط پارامتر CONTROL_FILES درفایل Init.ora تعیین می شوند. اگرچه

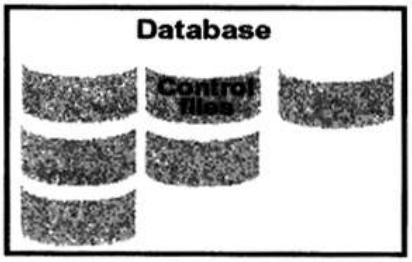
پارامتر CONTROL_FILES در فایل Init.ora معمولاً در فایل Cofig.ora معین می گردد، اما به ندرت تغییر می کند. در صورتیکه نیاز به افزودن یک Control File جدید به بانک اطلاعاتی باشد می توان Instance را Shut Down کرده و یکی از Control File های موجود را در مسیر جدید کپی و آن را به پارامتر CONTROL_FILES اضافه و Instance را دوباره Start کرد. اطلاعاتی که در Control File بانک اطلاعاتی ثبت می شود، عبارتند از :

۱. نام بانک اطلاعاتی
۲. ID یکتای بانک اطلاعاتی
۳. تاریخ ایجاد بانک اطلاعاتی
۴. نام و مسیر Data File های بانک اطلاعاتی
۵. نام و مسیر Redo Log File های بانک اطلاعاتی
۶. نام Redo Log File جاری
۷. آخرین شماره تغییر بانک اطلاعاتی (System Change Number)
۸. اطلاعات مربوط به Tablespace ها
۹. اطلاعات مربوط به Checkpoint ها
۱۰. اطلاعات مربوط به Archive Log File ها
۱۱. اطلاعات مربوط به Backup های RMAN

Control File

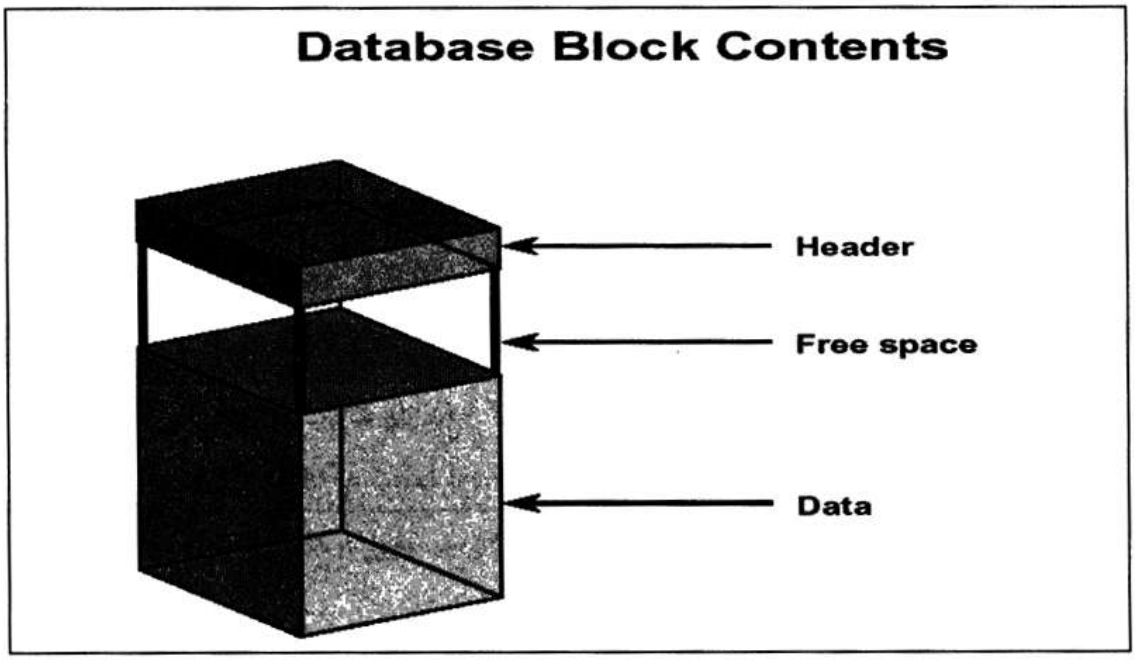
The control file is a binary file that defines the current state of the physical database..

- **Loss of the control file requires recovery**
- **Is read at MOUNT stage**
- **Is required to operate**
- **Is linked to a single database**
- **Should be multiplexed**
- **Maintains integrity of database**
- **Sized initially by CREATE DATABASE**



Data Block

Data Block ها کوچکترین واحد ذخیره سازی داده ها در بانک اطلاعاتی ORACLE بوده و بصورت فیزیکی بر روی دیسک ذخیره می شوند. در بیشتر سیستمها، اندازه بلاکها چهار کیلو بایت می باشد. اما بسته به عملکرد سیستم عامل و نیاز کاربر، می توان این اندازه را تغییر داد. اندازه Data Block در زمان ایجاد بانک اطلاعاتی مشخص گردیده و بعد از ایجاد بانک اطلاعاتی نیز امکان تغییر اندازه Data Block وجود دارد. Data Blockها علاوه برداده ها، حاوی اطلاعات مربوط به Header خود Block نیز می باشند.



Data Block Buffer Cache

Data Block Buffer Cache ناحیه ای است در SGA که برای نگهداری کپی از بلوکهای داده ای که از Data Segmentخوانده می شوند (مانند: جداول، Index ها و Cluster ها) استفاده می شود.

Data Block Buffer Cache کپی از داده های خوانده شده و یا تغییر یافته از فایلهای فیزیکی بانک اطلاعاتی را نگهداری می کند. Buffer های متعددی در ناحیهٔ Data Block Buffer Cache وجود دارند که عبارتند از:

- **Dirty Buffer** : Buffer هایی که داده های درون آنها تغییر کرده ولی به دیسک منتقل نشده اند.
- **Pinned Buffer** : Buffer هایی که در دسترس بوده و در درخواستهای بعدی به جای خواندن فایلهای فیزیکی از این Buffer ها استفاده می شود.
- **Free Buffer** : Buffer هایی که خالی بوده و قابل استفاده می باشند.

Data Dictionary

Data Dictionary یکی از مهمترین اجزای بانک اطلاعاتی اراکل می باشد. در زمان ایجاد یک بانک اطلاعاتی در اراکل جداول از قبل تعریف شده با ساختار مشخص در یک Tablespace به نام SYSTEM ایجاد و نگهداری می گردند. این جداول را جداول Data Dictionary می نامیم. از آنجایی که در بانک اطلاعاتی اراکل هر شیء باید در یک Schema قرار داشته باشد، این جداول در Schema ی کاربری به نام SYS قرار دارند. بنابراین کاربر SYS بعنوان owner جداول Data Dictionary می باشد.

اطلاعات هر شیء که در بانک اطلاعاتی ایجاد و یا تغییر می یابد، در Data Dictionary اراکل ثبت می گردد و همچنین در صورت حذف، اطلاعات آن شیء از Data Dictionary حذف می شود. Data Dictionary یک ساختار فیزیکی برای ذخیره سازی داده های بانک اطلاعاتی نمی باشد، بلکه به عنوان یک دیکشنری داده ها، نام و آدرس اشیاء بانک اطلاعاتی را برای دستیابی، نگهداری می کند.

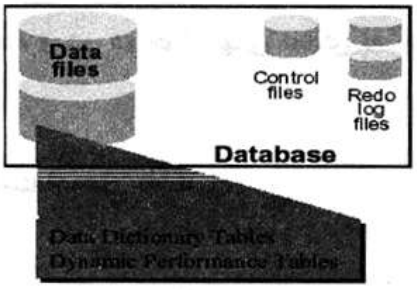
Data Dictionary از ۴ گروه view تشکیل می گردد که هر گروه پیشوند مشخصی دارد:

پیشوند	شرح
USER	شامل اشیائی است که به یک کاربر تعلق دارد. view هایی با این پیشوند امکان مشاهده اشیائی را می دهند که توسط آن کاربر ایجاد شده است.
ALL	مشاهده اشیائی که مربوط به یک کاربر بوده و همچنین اشیائی که مجوز دسترسی به آن واگذار گردیده است .
DBA	تنها کاربرانی که دارای نقش DBA می باشند به این view ها دسترسی داشته و از طریق آن می توان به تمامی اشیاء بانک اطلاعاتی متعلق به هر کاربر دسترسی داشت.
V$	برای تنظیم کارآیی بانک اطلاعاتی مورد استفاده قرار می گیرند.

Data Dictionary

During database creation, the Oracle server creates additional object structures within the data files.

- **Data dictionary tables**
- **Dynamic performance tables**



Data File

تنها واحد ذخیره سازی داده ها در بانک اطلاعاتی را Data File می نامند. از طریق این فایلها امکان نگهداری اشیاء Schema مانند جداول، View ها، Index ها و.... در بانک اطلاعاتی وجود دارد. در بانک اطلاعاتی ORACLE برحسب نیاز می توان از یک تا صدها Data File را ایجاد و مورد استفاده قرار داد.

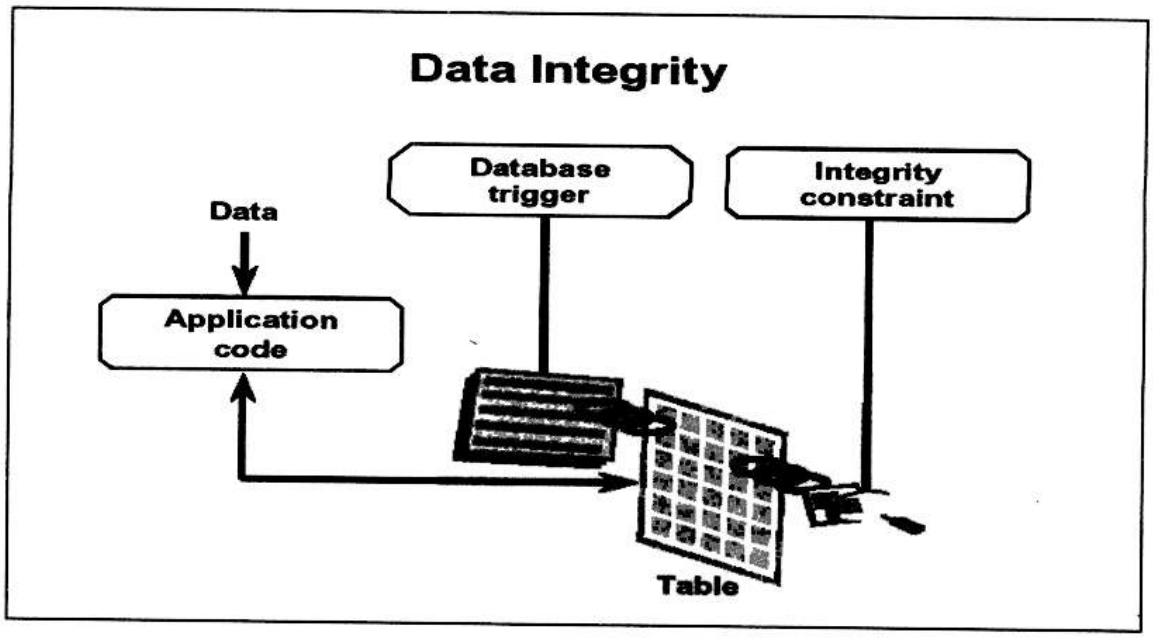
این فایلها یک بانک اطلاعاتی را به واحدهای کوچکتر برای مدیریت تقسیم می کنند. در واقع بانک اطلاعاتی ORACLE از طریق این فایلها به واحدهای کوچکتر که امکان توزیع آنها بر روی دیسکهای مختلف فراهم می شود، تقسیم می شود.

توزیع اطلاعات بین چند Data File، اثر قابل توجهی بر کارآیی سیستم خواهد گذاشت. Data File ها در بانک اطلاعاتی ORACLE مستقیماً مورد دستیابی قرار نمی گیرند، بلکه از طریق یک لایه منطقی به نام Tablespace مدیریت می شوند. هرData File تنها می تواند به یک Tablespace تعلق داشته باشد. امکان توسعه فضاهای Data File ها بعد از ایجاد آنها در بانک اطلاعاتی وجود دارد. در واقع Data File :

- تقسیمات فیزیکی بانک اطلاعاتی اراکل است.
- تنها واحد ذخیره سازی داده ها در بانک اطلاعاتی است.
- هرData File می تواند با یک نام مشخص در یک مسیر مشخص بر روی دیسک ایجاد شود.
- هرData File در زمان ایجاد با یک فضای اولیه مشخص ایجاد می شود. این فضای فیزیکی اولیه که در اختیار Data File قرار می گیرد از دید سیستم عامل یک فضای مصرفی و از دید بانک اطلاعاتی اراکل یک فضای رزرو شده اولیه برای ذخیره داده ها می باشد که می تواند در صورت نیاز رشد یابد.
- هرData File برای بانک اطلاعاتی اراکل همانند یک پارتیشن برای سیستم عامل می باشد.
- هر Data File یک فضای فیزیکی به هم پیوسته بر روی یک پارتیشن سیستم عامل می باشد. بنابراین نمی توان Data File ای ایجاد نمود که بخشی از فضای خود را از یک پارتیشن و بخشی از پارتیشن دیگر سیستم عامل دریافت کرده باشد.

Data Integrity

بر اساس تئوری رابطه ای، هر جدول دارای مجموعه صفاتی می باشد که هر کدام از ردیفهای موجود در آن جدول را منحصراً مشخص می سازد. ردیف تکراری در جدول غیر ممکن است. راه دیگر بیان این مطلب، آن است که هر جدول باید دارای یک کلید اصلی باشد، به این مفهوم جامعیت داده ها می گویند .



Data Segment

یکی از تقسیمات درونیData File های بانک اطلاعاتی که برای نگهداری بلوکهای داده ای جداول و Cluster ها مورد استفاده قرار می گیرد. اگر نوع Tablespace در زمان ایجاد از نوع Permanent انتخاب شود در آنصورت بلوکهای داده ای آن از نوع Data Segment خواهد بود.

Data Warehouse

اگر تعداد Transactionهای بانک اطلاعاتی کم ولی اندازه آنها بزرگ باشد، می توان بانک اطلاعاتی را به حالت Data Warehouse پیکربندی کرد. به این حالتDecision Support System (DSS) نیز اطلاق می گردد. در این حالت Query های پیچیدهٔ بسیاری که انجام می شود، پردازش زیادی از اطلاعات، با دسترسی زیاد، زمان پاسخگویی خوب و دقت را نیازمند است. در یک محیط DSS یا سیستم پشتیبانی از تصمیم گیری، عمدتاً بانک اطلاعاتی وظیفهٔ اجرای Query های بسیار پیچیده ای را به عهده دارد که منابع بسیار زیادی را برای اجرا نیازمند است. انتخاب حالت OLTP و یا Data Warehouse باعث تغییر اندازه Data Block های بانک اطلاعاتی می گردد. در حالت OLTP اندازه Data Block ها کوچکتر و در حالت Data Warehouse اندازه Data Block ها بزرگتر در نظر گرفته می شود.

Database

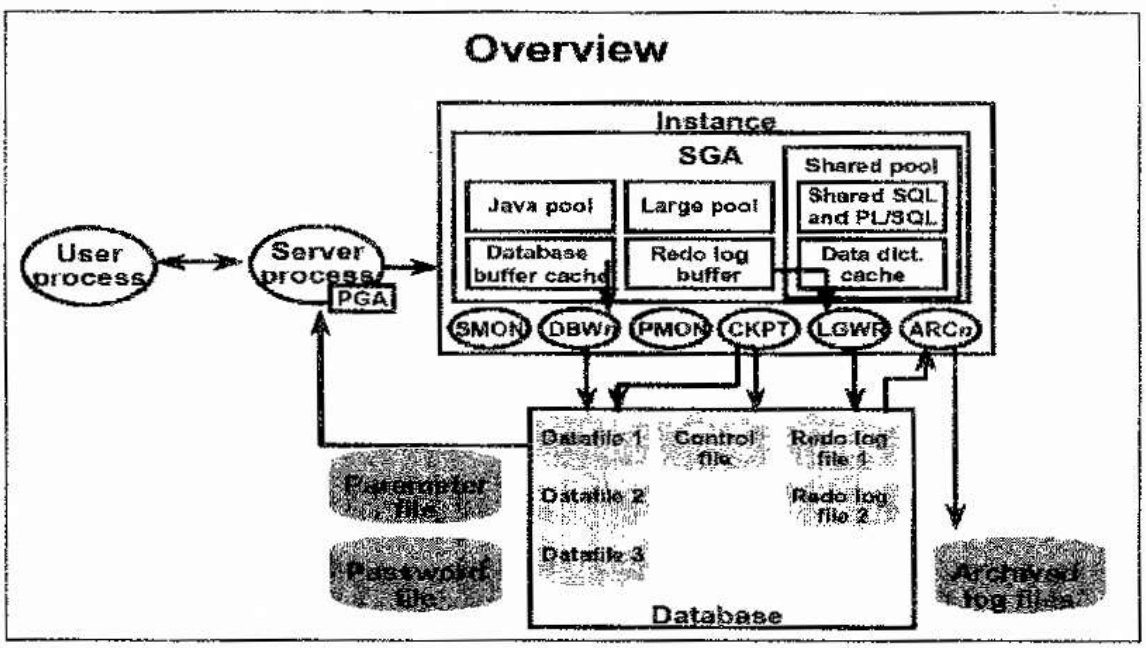
هر بانک اطلاعاتی اراکل بر روی کامپیوتر Server از دو ساختار زیر تشکیل می شود:

۱. Instance
۲. ساختار های فیزیکی بانک اطلاعاتی مانند Control File ها، Redo Log File ها و Data File ها

Instance روش دسترسی به داده های بانک اطلاعاتی اراکل در ساختارهای فیزیکی آن می باشد. هر Instance بانک اطلاعاتی خود از دو ساختار زیر تشکیل می شود:

۱. ساختار های حافظه ای

۲. ساختار های پردازشی



Database Link

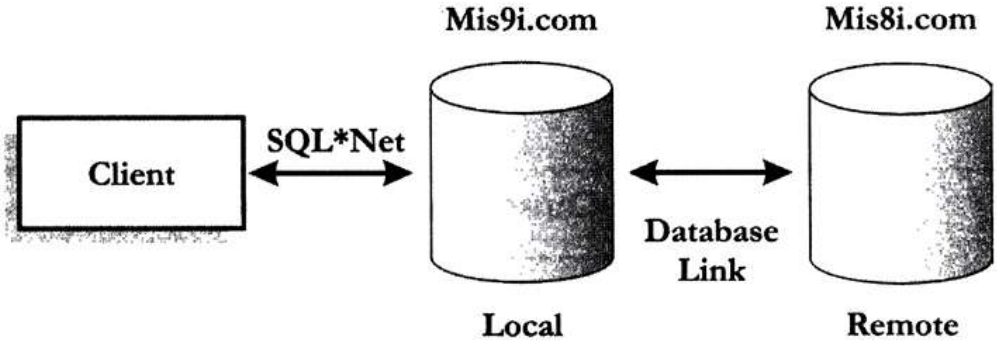
بانکهای اطلاعاتی ORACLE قادر به فراخوانی داده هایی می باشند که در خارج از بانک اطلاعاتی محلی آنها ذخیره شده باشد. یعنی هرگاه بخواهیم دو بانک اطلاعاتی طوری در کنار هم قرار بگیرند که درصورت Connect بودن به یک بانک بتوان Query در بانک دیگر اجرا نمود، از Database Link استفاده می شود. Database Link به صورت یک شیء در بانک اطلاعاتی ذخیره می شود.

به بانک اطلاعاتی که ارتباط اولیه با آن انجام شده، Local و به بانک دیگرRemote اطلاق می شود. در بانک اطلاعاتی Local برای برقراری ارتباط با بانک Remote اشیاء زیر باید تعریف می شوند:

- تعریف یک Net Service Name که به طور مستقیم به بانک اطلاعاتیRemote دسترسی داشته باشد.
- تعریف یک Database Link که با استفاده ازNet Service Name تعریف شده، امکان برقراری ارتباط را ممکن سازد.

برای ایجاد Net Service Name از برنامهNet Manager استفاده می شود و برای ایجاد Database Link باید به بانک اطلاعاتی متصل بود. برای ایجاد Database Link به مجوزی در بانک اطلاعاتی نیاز می باشد.

کاربرانی که دارای مجوز DBA می باشند، به صورت پیش فرض این مجوز را دارا می باشند.



ویژگی مهم Database Link آن است که یک کاربر با اتصال به بانک اطلاعاتی Local به سایر بانک ها در محیط شبکه بطور همزمان دسترسی دارد، در نتیجه این ویژگی مهم امکان تعریف Query هایی که در بیش از یک Server برای دستیابی به داده ها اجرا می شوند، را پشتیبانی می کند.

برای مشخص کردن راه دسترسی به یک شیء در بانک اطلاعاتی Remote می بایست از یک Database Link استفاده شود که می تواند بصورت Public (در دسترس تمام کاربران بانک اطلاعاتی) و یا Private (بدین معنا که تنها توسط یک کاربر ایجاد شده و تنها خودش می تواند از آن استفاده کند) باشد.

وقتی که یک Database Link ایجاد می شود باید نام کاربر مربوطه، کلمه عبور کاربر و نام سرویس مرتبط با بانک اطلاعاتی Remote را مشخص نمود. در صورتیکه نام کاربر معین نشود ORACLE از نام کاربر جاری و کلمه عبور آن برای ایجاد رابطه با بانک اطلاعاتی Remote استفاده خواهد کرد. به عنوان مثال در دستور زیر یک Public Link به نام Mis9i.com ایجاد شده است.

```
CREATE PUBLIC DATABASE LINK mis9i.com
connect to scott identified by tiger
using 'mis9i';
```

دراین مثال از Net Service Name ای به نام mis9i برای برقراری ارتباط با بانک Remote استفاده شده است. نام Database Link در تعریف Qurey هایی که باید در بانک Remote اجرا شوند، در قسمت from مشخص می شود. مانند مثال زیر:

```
select * from EMPLOYEE@mis9i.com;
```

امکان تعریف Synonym برای Database Link وجود دارد. در مثال زیر جدول EMPLOYEE در بانک اطلاعاتی Remote به همان نام EMPLOYEE توسط یک Synonym تعریف شده است.

```
create Synonym EMPLOYEE for EMPLOYEE@mis9i.com;
```

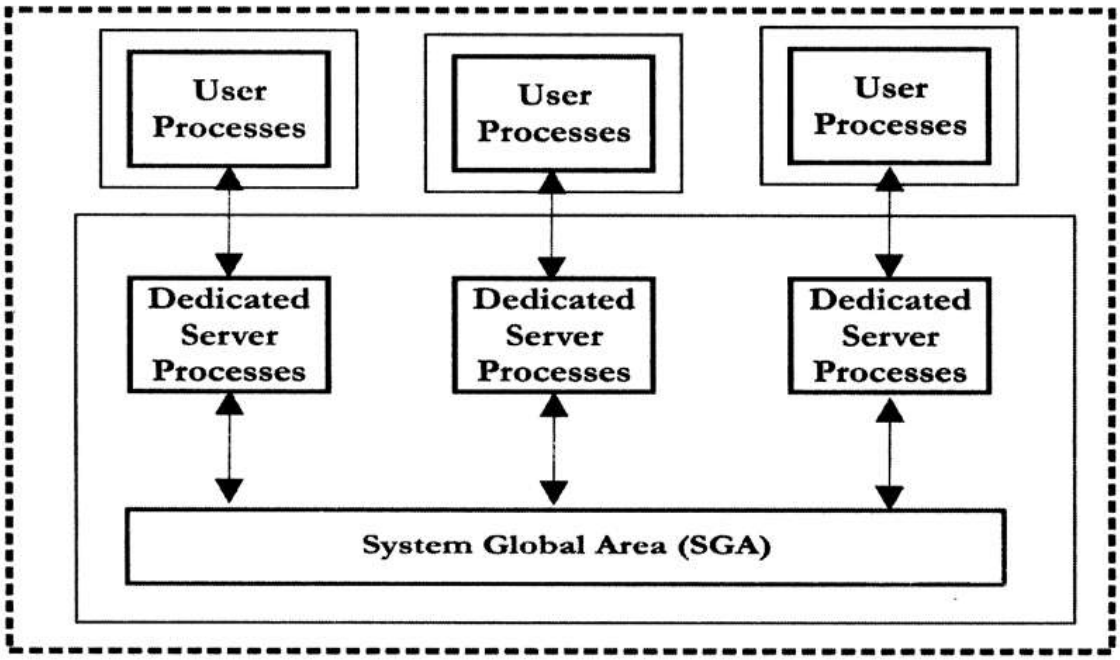
در صورتی که یک Procedure, Package ویا Trigger ذخیره شده، شامل یک ارجاع به یک Database Link باشد وجود این Link برای کامپایل کردن PL/SQL ضروری خواهد بود.

DBWR

پردازش پیش زمینه DBWR (Database WRiter) مدیریت محتویات Data Block Buffer Cache را بر عهده دارد. این پردازش وظیفه نوشتن بلوکهای تغییر یافته (Dirty Block) در ناحیه SGA به Data File ها را به عهده دارد. این پردازش همچنین وظیفه ایجاد فضا های مورد نیاز برایData Block Buffer Cache را بعهده دارد، بنابراین ابتدا بلاکهایی که کمترین استفاده را داشته از این ناحیه آزاد می شوند.

Dedicated Server Mode

یکی از تنظیمات بانک اطلاعاتی اراکل قرار دادن بانک در حالت Dedicated Server Mode می باشد. قرار دادن بانک در حالت Dedicated Server Process رابطه یک به یک میان پردازش کاربر و پردازش سرویس دهنده بوجود می آورد، زیرا هر پردازش سرویس دهنده، به یک پردازش کاربر اختصاص داده می شود.



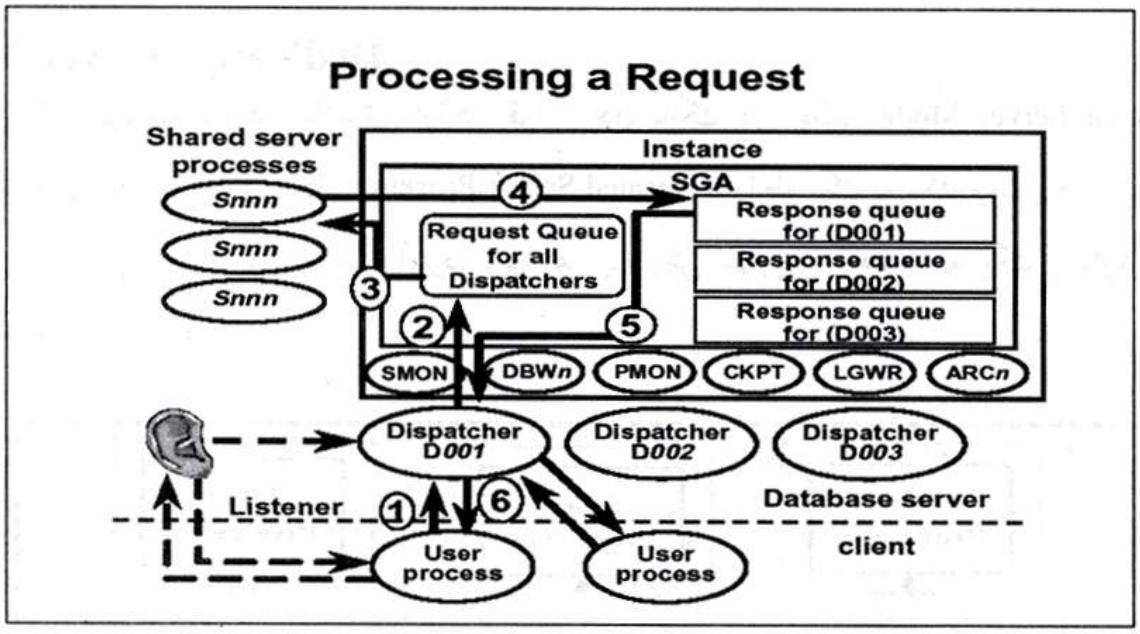
Dispatcher Pool

Dispatcher Pool وظیفه تجزیه دستورات DISPATCHER را بر عهده دارد. اندازهٔ Dispatcher Pool برحسب بایت، به وسیله پارامتر DISPATCHER_POOL_SIZE در فایل Init.ora که از ORACLE 8i معرفی شده، تعیین می شود.

Dispatcher

اگر ازحالت MTS استفاده شود، در آنصورت Dispatcher ها وظیفه برقراری ارتباط User Process ها به Server Processها را به عهده دارند.

تعداد Dispatcherها و همچنین پروتکلهایی که پشتیبانی می کنند، در فایل پارامتری Instance مشخص می شوند.



Dump File

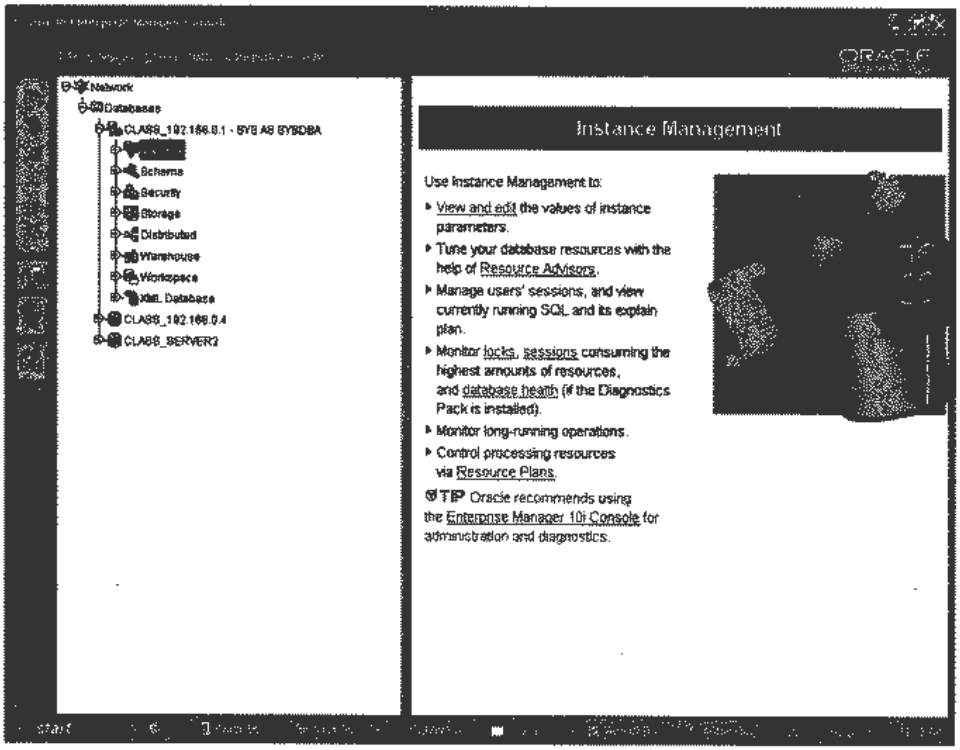
به فایلهایی که از Backup های منطقی اراکل ایجاد می شود Dump File می گویند. هر دستور Export ای که در بانک اطلاعاتی اجرا می شود یک فایل Dump برای نگهداری اطلاعات Backup خود ایجاد می کند. این فایل از طریق Import قابل خواندن بوده و مجدداً برای ایجاد اشیاء بانک مورد استفاده قرار می گیرد.

Enterprise Manager Console

محیطی گرافیکی که یکی از مهمترین محیط های مدیریتی بانک اطلاعاتی اراکل به شمار می رود. از این محیط گرافیکی امکان ایجاد، تغییر و یا حذف تمامی اشیاء بانک اطلاعاتی در هر اسکیمای بانک

امکان پذیر می باشد. کاربرانی می توانند از این محیط گرافیکی با بانک اطلاعاتی ارتباط برقرار نمایند که دارای مجوز DBA و یا SYSDBA باشند. بنابراین کاربران SYS و SYSTEM کاربران پیش فرضی هستند که بعد از نصب اراکل به این محیط گرافیکی برای مدیریت بانک دسترسی دارند. این محیط از ۴ قسمت اصلی زیر تشکیل می شود :

- Instance Manager
- Schema Manager
- Security Manager
- Storage Manager



EXPLAIN PLAN

ابزاری از اراکل که برای تنظیمات بانک اطلاعاتی مورد استفاده قرار می گیرد. از این ابزار می توان هزینه اجرای یک Query را در بانک اطلاعاتی مشخص کرد.

Export

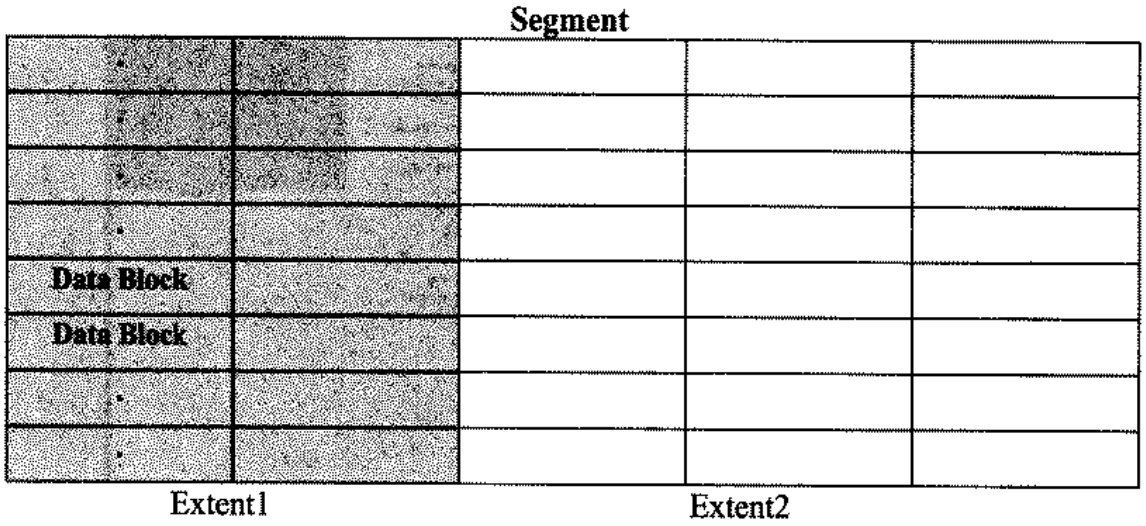
دستور EXPORT در ORACLE بانک اطلاعاتی را که شامل Data Dictionary می باشد، خوانده و خروجی آنرا در یک فایل باینری به نام Export DUMP می نویسد. می توان از تمامی اشیاء بانک اطلاعاتی به طور کامل، انتخاب یک یا چند کاربر و یا برخی از جداول کاربران Export گرفت. با استفاده از Export

می توان برخی از اشیاء وابسته به جداول از قبیل مجوزها، ایندکسها و constraint های مربوط به جداول را نیز به همراه جداول Export کرد. فایلی که توسط Export ایجاد می شود، برای ایجاد مجدد اشیاء Export شده، مورد استفاده قرار می گیرد.

Extent

مجموعه ای از Data Block ها تشکیل یک Extent را می دهند. Extent ها در بانک اطلاعاتی ORACLE به سه گروه تقسیم می شوند:

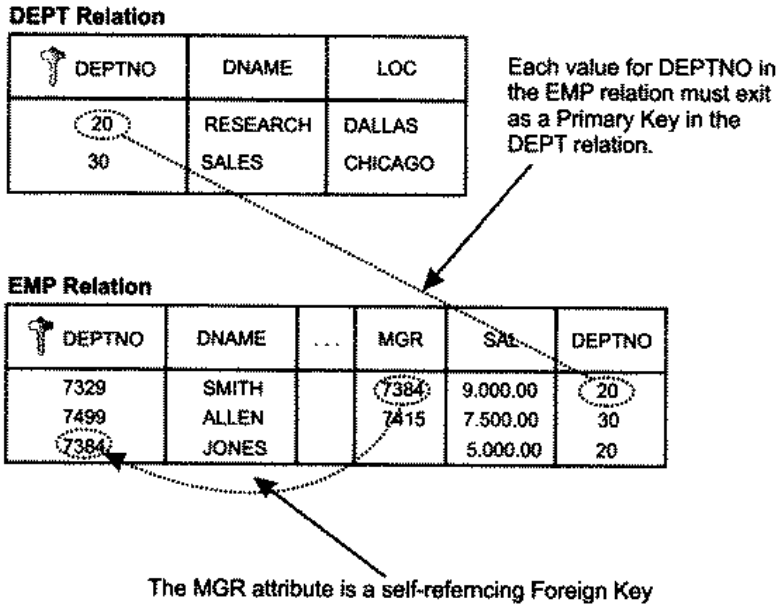
۱. **Initial Extent :** اولین Extent ای که در هر Segment ایجاد می گردد را Initial Extent می نامند.
۲. **Next Extent :** دومین Extent ای که در هر Segment ایجاد می گردد را Next Extent می نامند.
۳. **Extent سوم به بعد در هر Segment :** که درصدی از اندازه Next Extent می باشند.



تقسیمات Data Block ها و Extent ها در یک Segment

Foreign key

Foreign key ستونی از یک جدول می باشد که به primary key از جدول دیگر ارجاع می شود. ستونهای مشترک همواره primary key از یک جدول و Foreign key از جدول دیگر می باشند. قانون جامعیت ارجاعی مشخص می کند که مقادیر Foreign key در یک جدول به مقادیر primary key در جدول دیگر ارجاع می شود. همچنین Foreign key ممکن است به primary key از همان جدول ارجاع شود.



Foreign Key محدودیتهای زیر را اعمال می کند:

۱. جدول Master حذف نمی شود مگر آنکه رابطه (Foreign Key) حذف شده باشد.

۲. رکوردی از جدول Master حذف نمی شود مگر آنکه دارای رکوردهایی وابسته در ستون Foreign Key خود نباشد.

۳. رکوردی در ستون Foreign Key از جدول Details درج نمی شود مگر آنکه دارای مقداری متناظر در ستون Primary Key خود باشد.

Function

Function ها نیز همانند Procedure ها شامل بلوکهایی از PL/SQL می باشند که در بانک اطلاعاتی ذخیره می شوند. Functionها برخلاف Procedureها قادرند مقادیر را به برنامه فرا خوانندهCalling) Program) بازگردانند (دارای مقدار return می باشند). Functionها نیز همانند Procedureها دارای آرگومان های ورودی/خروجی می باشند. همچنین Function ها همانند Procedure ها از سه قسمت اصلی Exception , Body , Declration تشکیل می شوند.

تفاوت بین Function و Procedure در بازگرداندن مقدار بازگشتی است. هر Function علاوه بر آرگومانهای ورودی/خروجی یک دستور **return** نیزدر بدنه خود دارد، که باعث بازگشت این مقدار می شود. دستور **return** برای انتقال مقدار بازگشتی به برنامه فراخواننده (Calling Program) مورد استفاده قرار می گیرد. اگر نخواهیم هیچ مقداری به برنامه فراخواننده برگردد، از Procedure استفاده می کنیم.

Global Database Name

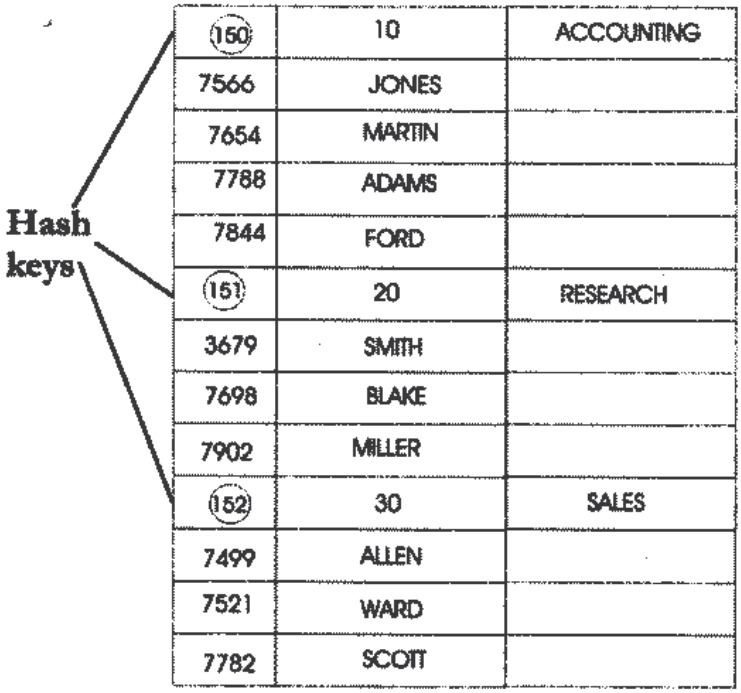
از Oracle 8i به بعد هر بانک اطلاعاتی علاوه بر نام SID دارای یک نام، به نام Global Database Name می باشد. از این نام برای برقراری ارتباط با بانک و پیکربندی SQL*Net استفاده می شود. این نام از دو قسمت که شامل نام بانک اطلاعاتی و Domain می باشد تشکیل می شود. انتخاب نام Domain برای Global Database Name اختیاری است. یعنی می توان SID و Global Database Name را هم نام در نظر گرفت.

Hash cluster

Hash Cluster شبیه به Index Cluster است، اما درآن به جای استفاده از یک Index برای مراجعه به Cluster key، از یک Hash Function استفاده می شود. Hash Function، تابعی عددی است که بلوک داده ای Cluster را بر اساس مقدار Cluster key محاسبه می کند و Hash Cluster ، داده ها را با توجه به خروجی Hash Function ذخیره می نماید. شکل زیر یک Hash Cluster را نشان می دهد. برای یافتن بلوک داده ای Cluster Index ، اول باید یک یا چند عمل I/O روی آن انجام گیرد تا بلوک داده ای صحیح پیدا شود. در Hash Cluster، خود Cluster key مشخص می کند که بلوک داده ای کجاست. بدین ترتیب، تنها یک عملیات I/O برای بازیابی سطر انجام خواهد شد.

بر خلاف Cluster Index که داده های مربوط را بر مبنای مقادیر Cluster key در کنار یکدیگر ذخیره می کند، Hash Cluster براساس مقادیری که از روی Hash Function به دست آمده اند، به ذخیرهٔ داده های مربوط در کنار یکدیگر می پردازد. تعداد مقادیری که Hash Function می تواند تولید کند، از روی پارامتر HASHKEYS در دستور CREATE CLUSTER تعیین می شود. تعداد و اندازه های Cluster key ها، بسیار مهم هستند و باید به دقت محاسبه شوند. Hash Cluster، نباید برای جداولی که عملیات پویش اغلب بر روی فقط یکی از آنها انجام می شود، مورد استفاده قرار گیرد؛ چرا که در این جداول، فضای اضافی مورد نیاز Cluster و عملیات I/O اضافی، کارآیی را کاهش می دهند.

همچنین، Hash Cluster نباید در جداولی که Cluster key آنها مکرراً از طریق برنامه کاربردی تغییر می کند، یا در جداولی که مدام در حال تغییر هستند به کار برده شود. در این نوع Cluster ، انجام مداوم محاسباتی که باید برای بدست آوردن بلوک داده ای بر روی کلید Cluster انجام شود، باعث تولید سربار قابل ملاحظه ای می گردد.



Hot backup

به Backup ای گفته می شود که در زمان Start بودن بانک اطلاعاتی از ساختارهای فیزیکی آن انجام می شود. در این نوع Backup به علت Open بودن بانک اطلاعاتی امکان تغییر محتوی بلوکهای داده ای بانک توسط کاربران وجود دارد. بنابراین Hot Backup تنها رکوردهای Commit شده در بلوکهای داده ای را در نظر می گیرد. برای انجام Hot Backup در یک بانک اطلاعاتی، نیاز به قرار دادن بانک در حالت Archive Log Mode می باشد. بنابراین اگر بانک اطلاعاتی در حالت Archive Log Mode قرار نداشته باشد تنها انجام Cold Backup از بانک اطلاعاتی امکانپذیر است و امکان انجام Hot Backup وجود ندارد.

Immediate

یکی از حالتهای Shutdown بانک اطلاعاتی می باشد. IMMEDIATE در دستور SHUTDOWN به صورت زیر Instance را Shutdown می کند:

۱. ارتباط تمامی کاربران را بلافاصله از بانک اطلاعاتی بدون هیچ پرسشی قطع می کند.
۲. تمامی Transaction های ثبت نشده را Rollback می کند.
۳. بانک اطلاعاتی را Shutdown می کند.

Import

Import فایل Export را خوانده و دستوراتی را که در آنجا ذخیره شده است را، مجدداً اجرا می کند.

Import ممکن است برای برگرداندن اشیاء و یا کاربران از فایل Export به صورت انتخابی، مورد استفاده قرار گیرد. بعد از Export داده ها، در صورت نیاز به Recovery بانک از Import استفاده می شود. Import فایلی که توسط Export ایجاد شده است را خوانده و دستورات SQL آن را مجدداً اجرا می کند. این عمل باعث بازسازی مجدد اشیاء بانک اطلاعاتی در صورت نیاز، می گردد.

Index

Index ساختاری است که برای دستیابی سریعتر داده ها در بانک اطلاعاتی، مورد استفاده قرارمی گیرد. Index ساختاری مستقل در بانک اطلاعاتی به شمار می رود. درصورت ایجاد یک Index یک Segment برای نگهداری از داده های Index ایجاد می شود. داده های Index مستقل از داده های جداول و یا Cluster های مربوط به Index می باشد. برای دستیابی سریع به داده ها، می توانیم از Index استفاده کنیم. ولی این امکان نیز وجود دارد که داده ها را مستقل از Index با جستجو از جدول بدست آوریم. ویژگی متمایز Index درسرعت دستیابی به داده های ذخیره شده در بانک اطلاعاتی می باشد.

وجود Index از دید کاربر یا برنامه کاربردی پنهان است و برای استفاده از آن نیازی به تغییر دادن برنامه کاربردی وجود ندارد. اما اگر از وجود Index مطلع باشیم، می توان برنامه کاربردی را چنان طراحی نمود که از Index های موجود، بهره بیشتری گرفته شود. بنابراین تنها نشانهٔ وجود Index ممکن است بهبود زمان دستیابی به داده ها باشد. وجود Index در بانک اطلاعاتی لزوماً باعث افزایش سرعت دستیابی به داده ها نمی گردد، اگر Index بر روی ستونهایی ایجاد شود که برای دستیابی به داده ها در قسمت شرط Query، مورد استفاده قرار نگیرد، وجود آن بی فایده خواهد بود. بعد از ایجاد Index بر روی یک جدول، نگهداری و مدیریت آن به عهده بانک اطلاعاتی خواهد بود. با انجام اعمالی چون درج، تغییر و حذف سطرهای جدول، Index های مربوط به طور خودکار Update می شوند.

هر جدول می تواند به تعداد دلخواه Index داشته باشد، اما هرچه تعداد Index ها افزایش یابد، سربار ناشی از اعمال تغییر، درج و حذف روی جدول بیشتر خواهد شد. این سربار بدان سبب بوجود می آید که تمامی Index های مرتبط باید هنگام تغییر داده های جدول Update شوند.

با توجه به بحث فوق، برای دستیابی به داده ها در بانک اطلاعاتی دو راه وجود دارد :

- خواندن تمامی ردیفهای جدول برای دستیابی به داده های مورد نیاز (full scan)
- استفاده از Index برای محدود نمودن داده های مورد نیاز (جلوگیری از full scan)

بنابراین Index مکانیزمی برای جلوگیری از full scan شدن جداول بانک اطلاعاتی فراهم می آورد و می تواند سرعت دستیابی به داده ها را افزایش دهد. برای تصمیم گرفتن در خصوص اینکه بهتر است بر روی کدام جدول Index ایجاد شود و یا ایجاد نشود، راهنمایی های زیر قابل استفاده خواهند بود:

- باید بر روی جداولی Index تعریف نمود که درخواستها، تعداد کمی ازسطرها را انتخاب کنند. درخواستهایی که سطرهای زیادی را انتخاب می نمایند، وجود Index را بی اثر می سازد. در حالت کلی، بهتر است زمانی از Index استفاده شود که درخواستها کمتر از ۵ درصد سطرهای جدول را مورد دستیابی قرار می دهند.
- بر روی ستونهایی که همواره در قسمت شرط Query، مورد استفاده قرار می گیرند.
- در مجموع، مناسب است بر روی جداولیIndex بگذاریم که با عبارت WHERE نسبتاً ساده ای مورد درخواست واقع می شوند عبارات WHERE پیچیده ممکن است از Index استفاده نکند.
- به صورت خودکار در زمان ایجاد یک Constraint از نوع Primary Key یک Index ایجاد می شود.
- به صورت خودکار در زمان ایجاد یک Constraint از نوعUnique یک Index ایجاد می شود.
- ستونهایی که در عملیات Join شرکت می کنند، برای Index شدن مناسب می باشند.
- بهتر است برای جداولی که داده های آنها زیاد تغییر می کنند، Index ایجاد نکنیم. در چنین جداولی، عملیات تغییر، درج و حذف باعث ایجاد سربار اضافی می شود. بهتر است، مبنای تصمیم گیری در خصوص ایجاد Index، تعداد عملیات تغییر، درج و حذف (در مقایسه با تعداد درخواستها) باشد.
- ستونهایی که دارای مقادیر NULL و تکرار زیاد باشند برای Index شدن مناسب نمی باشند.

Index Segment

یکی از تقسیمات درونی Data File های بانک اطلاعاتی که برای نگهداری بلوکهای داده ای Index ها مورد استفاده قرار می گیرد. اگر نوع Tablespace در زمان ایجاد از نوع Permanent انتخاب شود در آنصورت بلوکهای داده ای آن، امکان نگهداری اطلاعات مربوط به Index ها را خواهند داشت.

Index Table

جدول فقط Index، یکی از اشیاء بانک اطلاعاتی است که از ORACLE 8 به بعد معرفی شده است.

علت ترکیب کردن آنها) کاهش داده و بر میزان کارآیی خواهند افزود. کارآیی، بدان سبب افزایش می یابد که در این حالت به جای بازیابیRowID از Index های مرسوم، با خود داده ها مواجه هستیم. داده هایی که از طریق مقدار کلید اولیه مورد دستیابی قرار نگیرند، مناسب جداول فقط Index نیستند. همچنین، جداولی که مقدار کلید اولیه آنها تغییر کرده یا زیاد با عملیات درج سر و کار داشته باشند، برای این منظور مناسب نیستند.

Init.ora

هرInstance باید یک فایل پارامتری که آن را INIT.ORA می نامند، برای تنظیم پارامترها و وضعیت بانک اطلاعاتی خود داشته باشد. این فایل پارامتری در زمان Start شدن Instance توسط ابزارهای مدیریتی خوانده می شود. بعد از آنکه Instance، Start شد این فایل پارامتری دیگر مورد نیاز نمی باشد تا دفعه بعد که Instance دوباره Start می شود. این فایل پارامتری باید بر روی کامپیوتری قرار داشته باشد که ابزارهای مدیریتی Instance در آنجا قرار گرفته اند.

فایل پارامتری INIT.ORA برای مشخص کردن موارد زیر مورد استفاده قرار می گیرد:

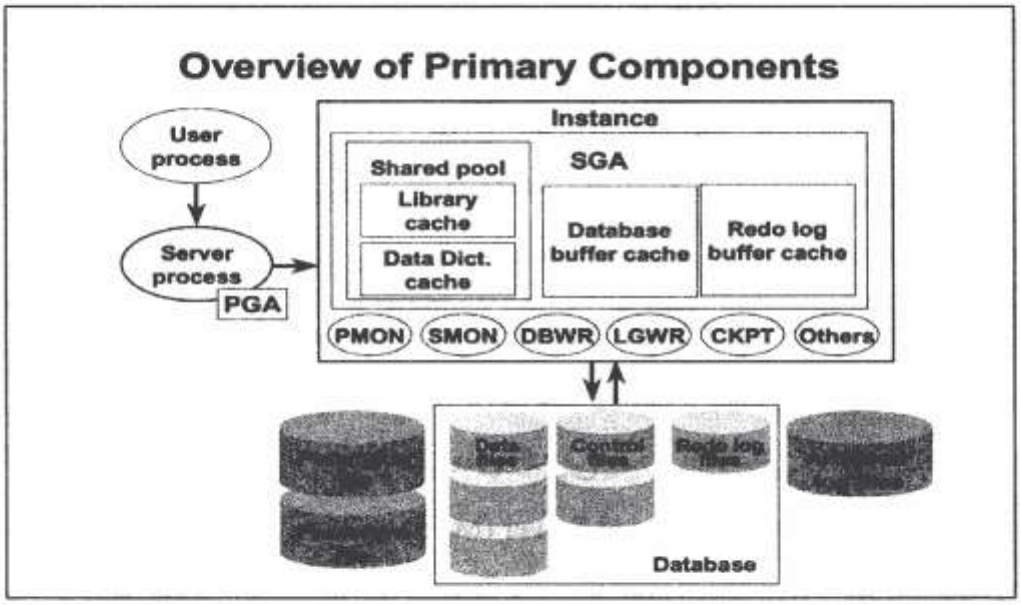
- میزان فضای قابل تخصیص به ساختار حافظه ای اراکل.
- Rollback Segment هایی که در Instance مورد استفاده قرار می گیرند.
- تنظیمات سایر زبانهایی که پشتیبانی می شود.
- تنظیمات اراکل برای حالت Parallel Server .
- بانک اطلاعاتی و Control File های آن که مورد استفاده هستند.
- زمانی که Checkpoint اعمال می شود.
- محدودیت برای ساختار های کنترلی بانک اطلاعاتی.
- تنظیمات راه اندازی Multi-Threaded Server (MTS).
- مقدار دهی پردازشهای زمینه غیر اجباری.
- نام و مسیر فایلهایی که بر روی سیستم عامل اطلاعات Trace، Dump و سایر فایلها را نگهداری می کنند.

PFILE Example

```
# Initialization Parameter File: initdb01.ora
db_name                 = db01
instance_name           = db01
control_files           = ( /u03/oradata/db01/control01db01.ctl,
                            /u03/oradata/db01/control02db01.ctl)
db_block_size           = 4096
db_block_buffers        = 500
shared_pool_size        = 31457280  # 30M Shared Pool
db_files                = 1024
max_dump_file_size      = 10240
background_dump_dest    = /u05/oracle91/admin/db01/bdump
user_dump_dest          = /u05/oracle91/admin/db01/udump
core_dump_dest          = /u05/oracle91/admin/db01/cdump
undo_management         = auto
undo_tablespace         = undtbs
. . .
```

Instance

به مجموعه ای از پردازشها و حافظه مشترک که برای دستیابی به داده های بانک اطلاعاتی مورد نیاز می باشد، Instance اطلاق می شود. هر Instance بانک اطلاعاتی از دو ساختار حافظه ای و پردازشی تشکیل می شود.

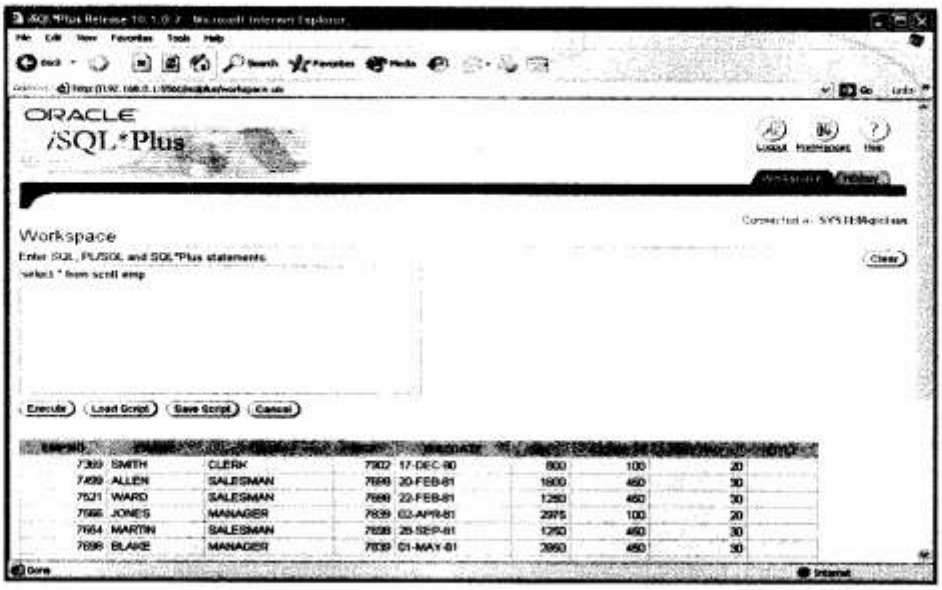


iSQL*Plus

برنامه iSQL*Plus یکی از محصولات اراکل برای اجرای دستورات SQL می باشد. برای فراخوانی محیط iSQL*Plus با استفاده از برنامه Internet Explorer آدرس زیر را در نظر می گیریم. تمامی کاربران بانک اطلاعاتی، که دارای حداقل مجوز CREATE SESSION می باشند، می توانند به این محیط گرافیکی

دسترسی داشته باشند.

HTPP://server کامپیوتر IP : 5560/isqlplus



سایر محیطهای دیگری که همانند iSQL*Plus امکان اجرای دستورات SQL را دارند، عبارتند از :

۱. SQL*Plus تحت DOS
۲. SQL*Plus تحت Windows
۳. SQL*Plus Worksheet

Join

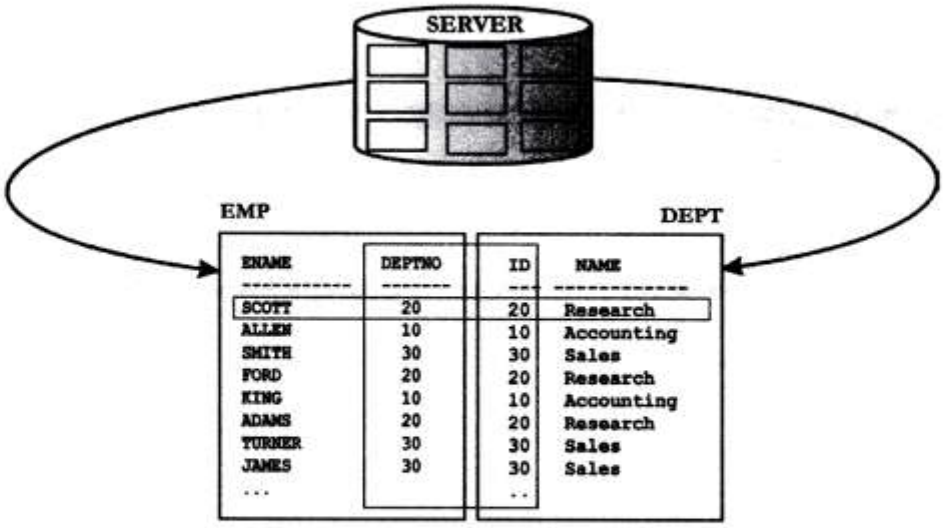
join امکان فراخوانی رکوردهای دو یا چند جدول که دارای مجموعه مقادیر یکسان هستند را فراهم می نماید. join پرس و جویی است که می تواند ردیفهای دو یا چند جدول را فراخوانی نماید.

join مکانیزمی جهت الحاق یک جدول به جدول دیگر می باشد. برای join دو جدول در قسمت WHERE ستونی از جدول اول (که اصولاً به صورت primary key تعریف می شود) با ستونی از جدول دوم (که اصولاً به صورت foreign key تعریف می شود) برابر می باشد.

در قسمت WHERE ممکن است شرط های بیشتری تعریف شود. قسمت شرط در دستور select که امکان join دو یا چند جدول را برقرار می نماید به صورت زیر تعریف می گردد :

WHERE table1.column=table2.column;

Equi Join



ارتباط دو جدول از طریق equi join

Large Pool

Large Pool یک ناحیه حافظهٔ انتخابی (Optional Memory Area) است. اگر از گزینهٔ Multi-Threaded استفاده شود و یا عمل Recovery مکرراً اجرا شود، زمانی می توان بطور مؤثر و کارآمد این امور را اداره نمود که یک فضای Large Pool ایجاد شود. این فضای Large Pool برای پشتیبانی از دستورات Large SQL اختصاص خواهد یافت که با استفاده از آن می توان از روی هم نوشته شدن ورودیها در Shared SQL Pool جلوگیری کرده و تعداد عباراتی که در Library Cache انباشته شده اند، را کاهش داد.

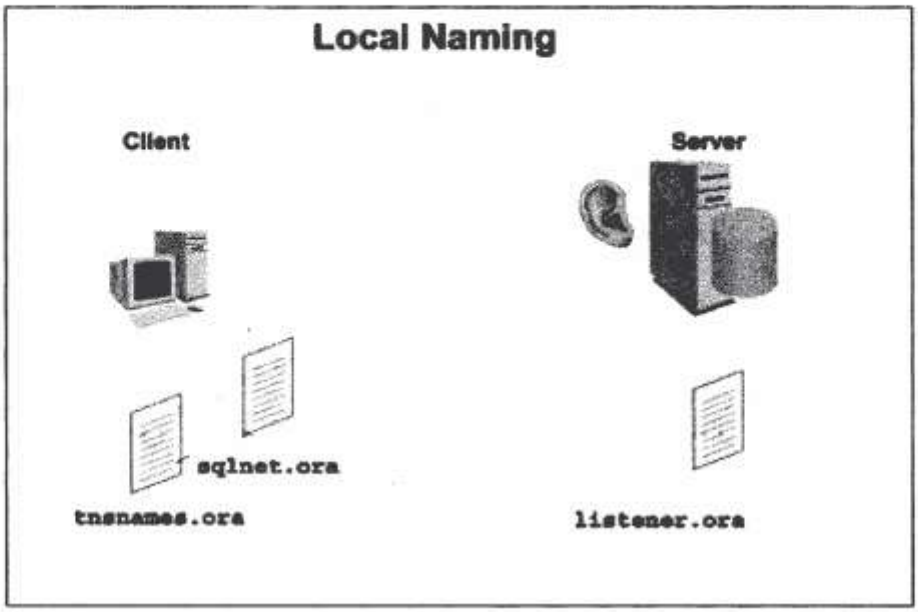
LGWR

پردازش پیش زمینه LGWR (Log Writer)، ثبت محتویات Redo Log Buffer را بر فایلهای Online Redo Log بر عهده داشته و ورودی های Log را به صورت گروهی در فایل های Online Redo Log می نویسد. ورودیهای Redo Log Buffer همواره شامل جدیدترین وضعیت بانک اطلاعاتی هستند، اما پردازش DBWR ممکن است تا زمان درخواست ثبت بلوکهای تغییر یافته از Data Block Buffer به Data File ها منتظر بماند.

Listener

پردازشی بر روی Server جهت گوش دادن به در خواست های ارتباطی از طرف Client به Server و مدیریت ترافیک Server. هرگاه یک Client یا یک Server همانند یک Client درخواست ارتباط با

Server را داشته باشند، یک Listiner ارتباط را به عهده دارد. اگر اطلاعات Client منطبق براطلاعات Listener باشد آنگاه Listener مجوز ارتباط با Server را صادر می نماید. به ازای هر پروتکل ارتباطی یک listener جهت برقراری ارتباط با Server مورد نیاز می باشد.



LOBs

در Oracle8 و بعد از آن برای مدیریت Large Object ها همانند تصاویر، صوت و .... چندین Built-In Type جدید تعریف گردیده است در زیر ۴ نوع از LOB ها لیست شده است:

- Binary Large Object : **BLOB**
- Character Large Object : **CLOB**
- National Character Set : **NCLOB**
- **BFILE** : داده باینری که در خارج از بانک اطلاعاتی در سیستم عامل نگهداری می شود.

LOB ها با داده های جداول دیگر ذخیره نمی شوند و باید برای هر ستون از نوع LOB یک Tablespace جدا در نظر گرفته شود. می توان در یک جدول از بیش از یک LOB استفاده نمود.

```
CREATE TABLE Emp_data
(
    Empono      NUMBER(10)PRIMARY KEY,
```

```
    Picture     BLOB,
    Bio         CLOB,
    Hr_fill     BFILE
)
TABLESPACE Ts_Emp_data
LOB (picture) STORE AS (TABLESPACE ts_pics
        Storage (initial 200M next 200M) CHUNK 16K),
LOB (bio) STORE AS (TABLESPACE ts_large_text
        storage (initial 2M next 2M) CHUNK 2K);
```

توجه شود که داده های BFILE در بانک اطلاعاتی ذخیره نمی شوند. بنابراین هیچ فضائی برای این Datatype تعریف نمی شود. کلمه کلیدی CHUNK مقدار داده برای هربار خواندن و نوشتن در ستون را مشخص می کند.

توابع(EMPTY_BLOB)، (EMPTY_CLOB) و(EMPTY_NCLOB) برای خالی کردن محتوی ستونهای LOB مورد استفاده قرار می گیرند. زیرا اگر ستونهای LOB دارای مقدار Null باشند نمی توان در آنها مقداری درج نمود. داده های BFILE در خارج از بانک اطلاعاتی ذخیره می شوند. بنابراین ستونهایی از این نوع می توانند بدون هیچ مشکلی دارای مقدار Null باشند. تابع(BFILENAME) برای مقدار دهی یک ستون از نوع BFILE جهت اشاره به یک فایل سیستم عاملی مشخص، مورد استفاده قرار می گیرد. مثال زیر استفاده از این توابع را نمایش می دهد.

```
INSERT INTO Emp_data
    VALUES (1001,EMPTY_BLOB(),EMPTY_CLOB(), NULL);

INSERT INTO Emp_data
    VALUES (1002, EMPTY_BLOB(),EMPTY_CLOB(),
BFILENAME('F:\ hrfiles','1002.doc'));
```

Type هایی از نوع LOB از طریق مشخص کننده هایی قابل دسترس می باشند و مستقیماً از طریق دستورات SQL و یا DML قابل دسترس نمی باشند. Package ای به نام DBMS_LOB دارای توابعی برای انجام اعمال DML بر روی ستونهای LOB است.

Media failure

یکی ازخرابی های بانک اطلاعاتی که به علت مشکل نوشتن بر روی دیسک به وجود می آید. زمانی که در یک دیسک یک فایل فعال بانک اطلاعاتی غیرفعال شود، Media failure رخ می دهد. Media failure ممکن است به دلیل از کارافتادگی یک دیسک و یا خطاهای خواندن در آن دیسک رخ دهد، که باید فایل های واقع بر دیسکها، جایگزین شوند.

Media recovery

ترمیم بانک اطلاعاتی که Media failure در آن رخ داده است. برای Media Recovery به داده های ذخیره شده در Redo Log File های بانک اطلاعاتی نیاز می باشد.

MOUNT

یکی از حالتهای Instance بانک اطلاعاتی می باشد که در زمان Start شدن بانک اتفاق می افتد. در این مرحله جستجو برای Control File بانک اطلاعاتی که در مرحله Nomount نام و مسیر آن از فایل پارامتری خوانده شده، انجام می شود. اگر تمامی Control File ها در دسترس Instance قرار بگیرند، در آن صورت مقادیر زیر ازControl File در این مرحله خوانده می شود.

- نام و مسیر Data File های بانک اطلاعاتی
- نام و مسیر Online Redo Log File های بانک اطلاعاتی
- وضعیت بانک اطلاعاتی از لحاظ قرار داشتن و یا قرار نداشتن در حالت Archive Log Mode
- آخرین شماره تغییر بانک اطلاعاتی
- اطلاعات مربوط به Tablespace ها

دستور قرار دادن بانک در حالت Mount به صورت زیر است :

```
STARTUP MOUNT;
```

Multiple Buffer Pool

می توان Multiple Buffer Pool ها را در SGA ایجاد کرده و از آنها برای جدا کردن مجموعه داده های بزرگ از دیگر برنامه های کاربردی استفاده کرد. این امر احتمال آنچه را که در Data Block Buffer Cache برای منابع مشابه وجود دارد، کاهش می دهد. برای هر Buffer Pool که ایجاد می شود، می بایست اندازه و تعداد LRU latches را مشخص کرد. تعداد Buffer ها باید دست کم، پنجاه بار از latche های LRU بیشتر باشد.

در زمان ایجاد Buffer Pool ها لازم است اندازهٔ Recycle Area و Keep Area را معین کرد. Keep Area همانند Reserved Area در Shared SQL Pool ورودی ها را نگه می دارد، در حالی که Recycle Area مکرراً به طور چرخشی عمل می کند.

پایان مقاله شماره دو

موضوع : دیکشنری اوراکل (بخش اول)

منابع :

کتاب آقای اسماعیل مومن

ویکیپدیا

NobodyCoder@ymail.com

www.ashiyane.org