



Oracle 体系结构

Oracle 核心竞争力



可扩展性:

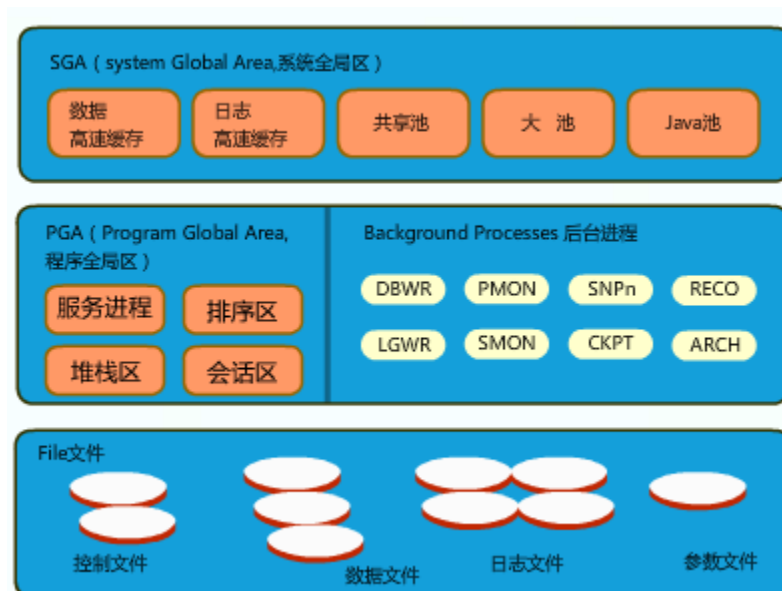
Oracle 系统有能力承担增长的工作符合，并且相应的扩充系统资源利用情况。

可靠性:

无论出现系统资源崩溃，电源断电还是系统故障的时候，我们都可以对 Oracle 进行配置。以保证检索用户数据事务处理的时候，不受到影响。

可管理性:

数据库管理员可以微调 Oracle 使用内存的方式，以及 Oracle 向磁盘写入数据的频率。并且管理可以调整，数据库为连接到数据库的用户分配操作系统进程的方式。



Oracle 总体结构分为三个部分

第一部分：系统全局区（SGA）

第二部分：程序全局区和后台进程

第三部分：Oracle 的文件

如果从功能上划分的话：

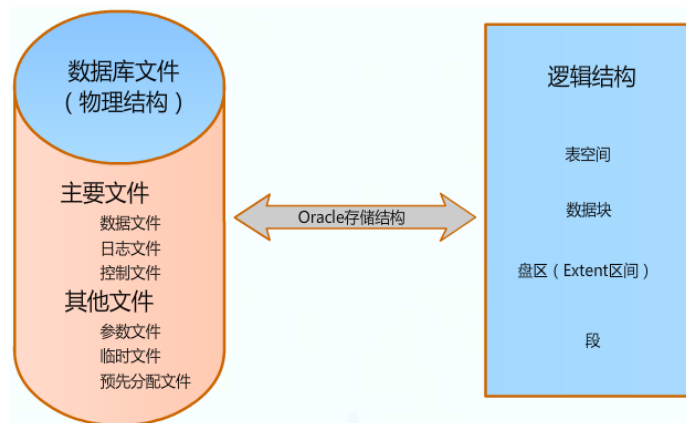
- ④ 存储结构：包括控制文件、数据文件、日志文件等文件
- ④ 内存结构：包括系统全局区（SGA）和程序全局区（PGA）。
- ④ 进程结构：包括前台进程、后台进程。

存储结构：由这些文件就构成了 Oracle 的物理存储结构。

内存结构：使用内存最多的是 SGA，也是影响数据库系统性能最大的一个参数。

进程结构：前台进程服务进程和用户进程，是根据实际需要而运行的，并在需要结束后立刻结束。后台进程是指 Oracle 数据库启动后自动启动的几个操作系统进程。

● 存储结构



包括物理结构和逻辑结构，既独立

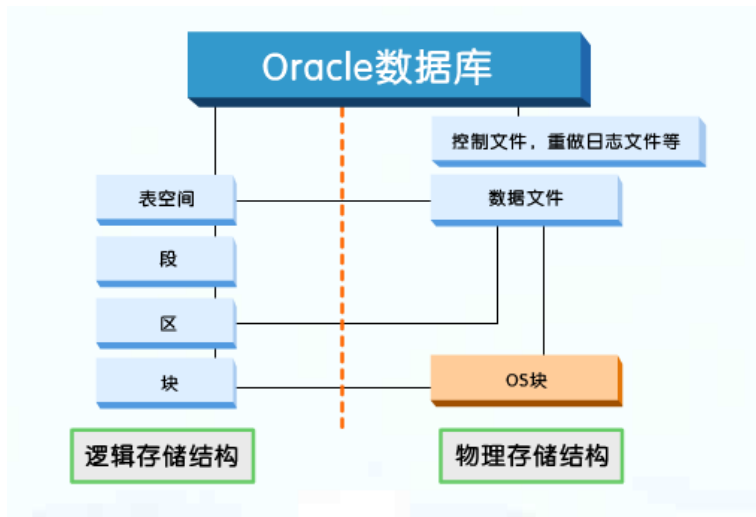
又相互联系的。

- ④ 物理存储结构主要描述 Oracle 数据库的外部存储结构，即在操作

物理结构：系统中如何组织、管理数据。是和操作系统平台有关的。

- ④ 逻辑存储结构主要描述 Oracle 数据库的内部存储结构，即从技术

逻辑结构：概念上描述在 Oracle 数据库中如何组织、管理数据。逻辑存储结构是和操作系统平台无关的。



这个图是两种存储结构中的对应关系

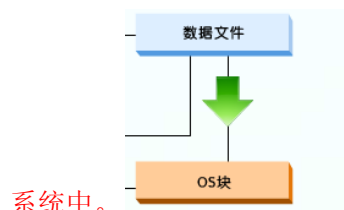
表空间包括系统表空间 and 用户表空间。表空间也是最大的逻辑单位。

块是最小的一个逻辑单位。



逻辑结构的表空间对应着物理结构的数据文件，也就是创建表空间的时候要给它指定数据文件，但是一个表空间可以对应多个数据文件。表空间的大小也就是他包含的数据文件大小的总和。

数据文件呢同时又对应着操作系统中的数据块，数据文件也是以数据块的形式，存在于操作



系统中。

逻辑结构中的块也对应着操作系统的数据块。



◆ 物理存储结构

Oracle 的物理存储结构，也就是 Oracle 的数据库文件

数据库文件又分为主要文件和其他文件

主要文件：

① 数据文件

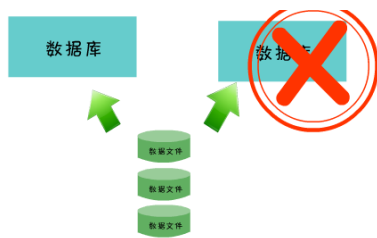
② 日志文件

③ 控制文件

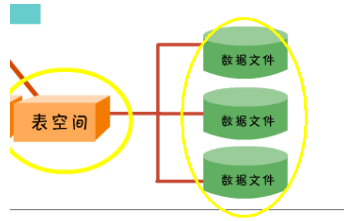
首先来看数据文件：

就是物理存储 Oracle 数据库数据的文件有以下特点：

- ◆ 每一个数据文件，只与一个数据库相关联。不能同时对应两个数据库。



- ◆ 一个表空间可以包含一个或多个数据文件，也就是说表空间的大小也就是他包含的数据文件大小的总和。



- ◆ 一个数据文件只能属于一个表空间



表空间有用户默认的表空间和临时表空间

用户默认的表空间可以由一个或者多个数据文件组成。

临时表空间就是我们用 `tmpfile` 选项指定的一个临时文件。用于存放排序操作产生的临时数据。当执行排序操作时如果临时数据超过排序区尺寸，那么服务器进程会将临时数据放到临时表空间中。但是不管是用户默认的表空间还是临时表空间，数据最终存储在数据文件中。

日志文件

🔍 重做日志文件 (Redo Log)

🔍 归档日志文件 (Archive Log)

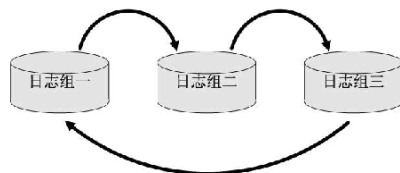
- ◆ 重做日志文件

用于记录数据库变化，是用户的事务处理日志。

当进行介质恢复和介质恢复的时候，我们需要使用重做日志，如果没有重做日志，用户能够执行的唯一恢复手段就是从最后的完整备份中复原。

数据库至少要包含两个重做日志组，并且这些重做日志组是循环使用的。

假定数据库中有三个日志组



在初始阶段日志文件写入进程，会将事务变化写入日志组

一，当日志组一写满后，Oracle 会自动进行日志切换，并且循环使用这三个日志组。当所有的日志写满后，如果需要归档，就会生成归档日志（是非活动重做日志的备份）

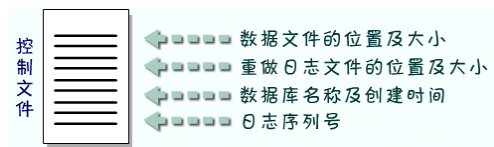
通过使用归档日志可以保留所有的重做历史记录。当数据库出现介质损坏时使用数据文件副

本和重做历史记录可以完全恢复数据库。

控制文件

④ 控制文件（Control Files）是Oracle服务器在启动期间用来标识物理文件和数据库结构的二进制文件。

在控制文件中主要记录一下信息：



Oracle 也会在常规的操作期间更新控制文件，以便准备下一次使用。

当用户建立控制文件的时候，建议用户建立多个副本，如果可能的话将这些副本放到不同的物理驱动器上（不同的磁盘下）。假如当设备崩溃的时候，用户就有完好的控制文件来启动和恢复数据库，如果没有恢复就会很复杂。

如何查询数据库中主要文件（视频）

其他文件

- ④ 参数文件
- ④ 临时文件
- ④ 预先分配文件

◆ 参数文件

参数文件（Parameter File）用于在启动实例的时候配置数据库。

位置：安装目录下 DATABASE 目录中。

参数文件有两种：

✚ 第一种是当建立数据库的时候，用户就可以运行初始化文件（也就是一种参数文件），规定数据库中所使用的各种设置值。

文本参数文件：

例如：数据库实例名称，数据库主要文件的位置，实例所使用的主要内存区域的大小。

作用就是：确定存储结构的大小，设置数据库的全部缺省值，设置数据库的范围，设置数据库的各种物理属性，以及优化数据库性能。

✚ 文本参数文件的名称格式为init<SID>.ora

SID 为数据库实例名称。

服务器参数文件的名称格式为SPFILE<SID>.ora，它可以管理数据库参数和值。

服务器参数文件：

它是静态文本初始化文件的替代物。

✚ 参数文件中参数的数据类型包括：整型、字符型、逻辑型



示例

```
DB_BLOCK_SIZE = 2048
DB_NAME = 'ora9'
CHECKPOINT_PROCESS = true
```

◆ 临时文件

Oracle中临时文件 (Temporary File) 处理方式与标准数据文件稍有不同。这些文件确实包含数据，但是只用于临时的操作。一旦建立它的会话，完成了操作，就会从数据库中将这些数据完全删除

如何创建临时文件：

```
SQL> create temporary tablespace temp_tbspace
```

```
tempfile 'd:/oracle/product/10.2.0/oradata/orcllib/temp_tbspace.dbf'
```

```
size 10M
```

```
extent management local
```

```
uniform size 512k;
```

◆ 预先分配文件

```
SQL> create tablespace MY_APPLICATION_TABLESPACE
2 datafile 'd:/oracle/product/10.2.0/oradata/orcllib/myapp.dbf' size 20M
3 autoextend on next 10M maxsize 1000M
4 extent management local uniform size 1M;
```

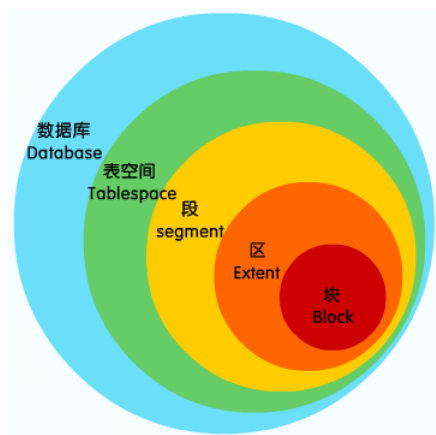
CREATE TABLESPACE

ALTER TABLESPACE

当用户创建表空间或修改表空间的大小，也就是使用 **ALTER TABLESPACE** 等 sql 命令，通常要在 size 子句中告诉 Oracle 数据文件的大小。

◆ 逻辑存储结构

Oracle 逻辑存储结构的组成



Oracle数据库的逻辑存储结构由一个或0多个表空间组成

一个表空间(tablespace)由一组段组成

一个段(segment)由一组区组成

一个区(extent)由一批数据库块组成

一个数据库块(block)对应一个或多个物理块

物理块也

就是操作系统块。

◆ 表空间

🔗 表空间是最大的逻辑存储结构，它对应一个或多个数据文件，表空间的大小是它所对应的数据文件大小的总和。

Oracle 建议将不同类型的数据部署到不同类型的空间，一方面可以提高数据访问性能，另一方面便于数据管理、备份恢复等操作。

- 🔗 SYSTEM表空间用于存放系统数据（数据字典信息）
- 🔗 SYSAUX表空间是Oracle Database 10g的新特性，（辅助系统表空间）用于减少系统负荷，提高系统的作业效率
- 🔗 用户表空间，永久存放用户对象和私有信息，也被成为数据表空间。
- 🔗 索引表空间用于存放索引数据
- 🔗 临时表空间用于存放临时数据
- 🔗 UNDO表空间（重做表空间）用于存放 UNDO数据

🔗 用户可以使用一个默认表空间和一个临时表空间。

默认表空间：是在默认情况下存储用户对象的表空间，当用户建立表的时候，就可以选择通知 Oracle 将表、数据存储在哪一个表空间中，如果用户没有规定表空间，那么 Oracle 就会将表数据存储在用户的默认表空间中。**用户的临时表空间**：是写入临时数据的地方，当用户进行查询，将数据块交换到磁盘上的时候，就会将所交换的数据存储到用户的临时表空间中。

表空间的特性

🔗 1. 控制数据库数据磁盘分配

就是说表空间可以拥有多个数据文件，各个数据文件可以存储在不同的磁盘中。从而达到了分配磁盘的目的。

🔗 2. 限制用户在表空间中可以使用的磁盘空间大小

🔗 3. 表空间具有 online, offline, readonly, readwrite属性

也就是说可以修改表空间的属性，来设置这个表空间是否可以被访问。

修改表空间属性的语法：

```
--修改表空间的属性：
SQL> alter tablespace 表空间名称 属性;      online      offline
--查询表空间状态：
SQL> select tablespace_name,status from dba_tablespaces;
```

注意的是：system undo temp 等表空间不能设置为 offline 属性。

🔗 4. 完成部分数据库的备份与恢复

在完成数据库的备份与恢复的时候，我可以只针对其中的一个表空间来完成相应的操作。

🔗 5. 表空间通过数据文件来扩大，表空间的大小等于构成该表空间的所有数据文件的大小之和

```
--查询表空间与数据文件对应关系：
SQL> select tablespace_name,bytes,file_name from dba_data_files;
```

Oracle10g 中自动创建的表空间

🔗 Example (实例表空间)

安装 Oracle 时创建的实例用户模式的数据信息。

🔗 Sysaux(辅助系统表空间)

减少系统负荷，提高系统的作业效率，该表空间是 Oracle10g 的一个新特性。

🔗 System(系统表空间)

存放关于表空间的名称，控制文件，数据文件等管理信息（最重要的表空间），属于 sys 和 system 这两个用户，并且只被这两个或者是其它具有足够权限的用户使用。但是所有用户都不可以删除或者重命名这个表空间。

🔗 Temp(临时表空间)

存放临时表和临时数据，主要是用于排序的过程中

🔗 Undotbs(重做表空间)

存放 Undotbs 数据

🔗 Users(用户的表空间)

用于永久存放用户对象和私有信息，也被称为数据表空间。

一般情况下系统用户使用 system 表空间，非系统用户使用 Users 表空间。

◆ 块

🔗 块是数据库使用的I/O最小单元，又称逻辑块或ORACLE块

一个数据块对应一个或多个物理块

块的大小由 db_block_size 确定,块的大小是操作系统块大小的整数倍，

🔗 数据块的结构包括块头和存储区两个部分

块头

数据块标题：在标题中存储着数据类型以及块的物理位置等信息。

表目录：在一个数据块中可以存储多个表的数据。表目录用于记录数据块中存储了那些表。

行目录：用于记录数据块中各行的物理地址。

存储区

空闲区：当第一次分配数据块的时候，它只有空闲区，没有数据行。随着行被插入，空闲区会越来越小。

行数据区：这是数据块中存储实际行的地方。

◆ 盘区（区，extent）

🔗 我们知道块是Oracle逻辑存储结构中最小的单位，盘区是由连续的数据块集合构成的，而盘区构成了段，段又构成了表空间，表空间构成了数据库。

🔗 盘区数量和盘区大小可以在正在建立的对象storage子句中规定。

例如：

```
SQL> create table my_hash_table(  
2   name varchar2(30),  
3   value varchar2(4000))  
4   tablespace users  
5   storage(  
6     initial 1M  
7     next 512K  
8     pctincrease 0  
9     minextents 2  
10    maxextents unlimited);
```

🔗 initial. 设置为对象建立的第一个盘区的大小。

🔗 next. 这是随后的盘区的大小。

🔗 minextents. 这是立即分配的盘区数量。

🔗 maxextents. 这是能够为这个表建立的盘区的最大数量。

◆ 段

📌 段是表空间中一个指定类型的逻辑存储结构，它由一个或多个区组成，段将占用并增长存储空间。

📌 段的四种类型：

- 📌 数据(表)段 (Data Segment) 存储表中的所有数据
- 📌 索引段 (Index Segment)：存储表上最佳查询的所有索引数据
- 📌 临时段 (Temporary Segment) 存储表排序操作期间建立的临时表的数据
- 📌 回滚段 (Rollback Segment) 存储修改之前的位置和值，是Oracle的撤销机制。

重点讲解回滚段

是负责存储回滚用户对数据库进行的修改所有必须的信息，他会记录事务处理对数据修改之前的数据副本。当用户回滚事务处理的时候，就可以使用回滚段中的信息，撤销用户对数据库所做的改变。

Oracle 内存结构

📌 内存结构是Oracle数据库体系中最重要的一部分，内存也是影响数据库性能的第一因素。

SGA(System Global Area)

📌 系统全局区：是所有用户都可以访问的实例的共享内存区域。

数据块、事务处理日志、数据字典信息等都存储在 SGA 中

PGA(Program Global Area)

📌 程序全局区：是一类设有共享的内存，它专用于特定的服务器进程，并且只能够由这个进程访问。

UGA (User Global Area)

📌 用户全局区：这个内存区域会为我们用户进程存储会话状态。

分别介绍：

SGA 系统全局区

📌 SGA是ORACLE系统为实例分配的一组共享缓冲存储区，用于存放数据库数据和控制信息，以实现数据库数据的管理和操作。

其中每个实例都只能有一个系统全局区，它是不同用户进程与服务进程进行通信的中心。数据库的各种操作主要在系统全局区进行。

SGA 所包含的数据：

- ☺ 缓存数据块
- ☺ 在数据库上执行的SQL语句
- ☺ 用户执行的存储过程、函数和触发器等

- 🔗 创建实例或启动实例时，Oracle为SGA分配内存
- 🔗 终止实例时，释放SGA所占用的内存

SGA 的组成

- ☺ DB高速缓存
- ☺ 共享池
- ☺ 重做日志缓存
- ☺ 大型池

数据高速缓存：

- 🔗 数据高速缓存database buffer cache (DBC)
- 🔗 DBC的作用：保存最近从数据文件中读取的数据块，其中的数据被所有用户共享

🔗 影响DBC的两个参数（主要影响读取效率）

- ☺ DB_BLOCK_SIZE 缓冲区数据块大小
- ☺ DB_BLOCK_BUFFERS 缓冲区数据块个数

- 🔗 DBC工作原理
 - ☺ 在我们初次访问数据时，Oracle将读取磁盘上的数据文件，将数据放入数据高速缓存，再处理数据
 - ☺ 如果数据已经位于数据高速缓存中时，Oracle就可以直接操作数据

共享池：

- 🔗 共享池保存了最近执行的SQL语句、PL/SQL程序和数据库字典信息，是对SQL语句和PL/SQL程序进行语法分析、编译、执行的内存区。

会被分割为许多分离的区域：

库缓存/Library Cache)区包括：解析用户进程提交的SQL语句或PL/SQL程序和保存最近解析过的 SQL语句或PL/SQL程序。

数据字典区(Dictionary cache)用于存放数据字典信息行。

重做日志高速缓存：

- 🔗 重做日志高速缓存就是用于存储重做记录的缓存

- ☺ 重做记录并不直接写入磁盘的重做日志文件，而是先写入重做日志缓存
- ☺ 当重做日志缓存中的重做记录达到一定数量时，由（日志写入进程）LGWR进程写入重做日志。（先内存后磁盘）

大型池：

- 🔗 大型池用于需要大内存操作提供相对独立的内存空间，以便提高性能

大型池是可选的内存结构，数据库管理员可以决定是否在系统全局区中创建大型池。

需要大型池的操作有：数据库备份和恢复，用于共享服务进程的会话内存（例如：大量排序的SQL语句），并行化的数据库操作等

PGA 程序全局区

PGA是为单独的服务器进程存储私有数据的内存区域

PGA是非共享的，而SGA是共享的

PGA大小由操作系统决定，并且分配后保持不变；会话终止时，自动释放PGA所占的内存

PGA组成

排序区

会话区

游标区

堆栈区

排序区：

排序区保存执行order by、group by等包含排序操作的SQL语句时所产生的临时数据

先将要排序的数据放到排序区中排序，再将排好序的数据返回给用户。

会话区：

会话区保存会话所具有的权限、角色、性能统计信息

游标区：

游标区保存执行带有游标(course)的PL/SQL语句时所产生的临时数据

游标就是指针，就是说指针的上下文存储在游标区中。

堆栈区：

堆栈区保存会话中的绑定变量、会话变量以及SQL语句运行时的内存结构信息。

例如：

```
select * from emp where emp=:a
```

在这个示例中，等待用户输入的时候，系统会先将变量存入堆栈区，以便可以同时运行其它语句。

系统全局区和程序全局区的区别

PGA与SGA类似，都是Oracle数据库系统为会话在服务器内存中分配的区域。不过两者的作用不同，共享程度也不同

SGA系统全局区是对系统内的所有进程都是共享的

PGA程序全局区则主要是为了某个用户进程所服务的

Oracle 的进程结构

- ① 进程是操作系统中的一种机制，它可执行一系列的操作步骤。
- ② Oracle实例有两种类型：单进程实例和多进程实例。

dos 下是单进程可视化操作就是多进程。

进程分类

- ① 操作系统会使用多个进程来执行ORACLE的不同部分，并且对于每一个连接的用户都有一个进程。
- ② Oracle的进程可分为：用户进程和Oracle进程

◆ 用户进程

当用户运行应用程序时，操作系统会为用户运行的应用建立一个进程。那么这个进程就是我们所说的用户进程。

在用户请求连接到数据库时启动，用户进程是一个需要和服务交互的程序。当应用程序向数据库服务器发送请求时即创建用户进程。

◆ Oracle 进程

Oracle进程又可以分为服务器进程和后台进程。

服务器进程

- ① 服务器进程用于处理连接到该实例的用户进程的请求。

当应用程序和 Oracle 服务器是在同一台机器上运行，而不在通过网络来连接时，服务器一般将用户进程和他相应的服务器进程组合成单个进程，可以降低系统开销。

然而当应用程序和 Oracle 服务器运行在不同的机器上时，用户进程将通过一个分离服务器进程与 Oracle 通信。这个时候服务器进程将执行一下几项任务：

服务器进程执行的任务：
对应用程序所发出的SQL语句进行语法分析和执行。
从磁盘中读入必要的数据块到SGA的共享数据库缓冲区
将结果返回给应用程序处理。

（第二个红框）前提是该数据块不在缓冲区

时，会执行从磁盘中读入的操作。

后台进程

后台进程是在实例启动时启动，启动Oracle数据库的实例时会自动启动后台进程。

是为所有数据库用户异步完成各种任务，不

论用户是否连接数据库，后台进程都作为数据库的一部分运行，每个后台进程都有它的职责，如果进程崩溃数据库也会崩溃。

主要的后台进程：

- 👤 PMON 进程监控进程；
- 👤 SMON 系统监控进程；
- 👤 DBWR 数据库写入进程；
- 👤 LGWR 日志写入进程；
- 👤 ARCH 归档进程；
- 👤 CKPT 检查点；
- 👤 CJQO 作业队列协调器
- 👤 RECO 恢复进程；

👤 监控服务器进程，以确保能够销毁发生损坏或者出现故障的进程，释放它们的资源。

PMON: 👤 在主机操作系统上使用Oracle监听器注册数据库服务器。

👤 在出现实例故障的情况下，SMON负责重新启动系统，执行崩溃恢复。

👤 SMON将会清除已经分配但还没有释放的临时段。

SMON: 👤 SMON也会在表空间管理中执行盘区结合。

👤 该进程将缓冲区用户所使用的数据写入数据文件。

DBWR: 👤 使用户进程总是可以得到未用的缓冲区。

👤 LGWR（日志写入进程）进程主要职责：是向在线重做日志文件中记录所有数据库的已提交事务处理。

LGWR: 日志写入进程会在以下4种情况执行写入操作

👤 事务处理进行提交

👤 重做日志缓存已经填充了1/3

👤 重做日志缓存中的数据数量达到了1MB

👤 每三秒的时间

👤 ARCH（归档进程）进程主要职责：将重做日志的事务变化写入到归档日志。

ARCH: 在重做日志文件中的事务处理被覆盖之前保存重做日志文件

👤 CKPT（检查点进程）进程主要职责：该进程在检查点出现时，对全部数据文件的标题进行修改。

在通常的情况下，检查点进程是不启动的它的工作由日志写入进程LGWR执行。然而，如果检查点明显地降低系统性能时，CKPT进程就会运行，将原来由LGWR进程执行的检查点的工作分离出来，由CKPT进程实现。

CKTP: 检查点进程不将数据写入磁盘。

👤 CJQO（作业队列协调器进程）进程主要职责：在Oracle中规划将要在数据库后台运行的进程或者作业（job）。

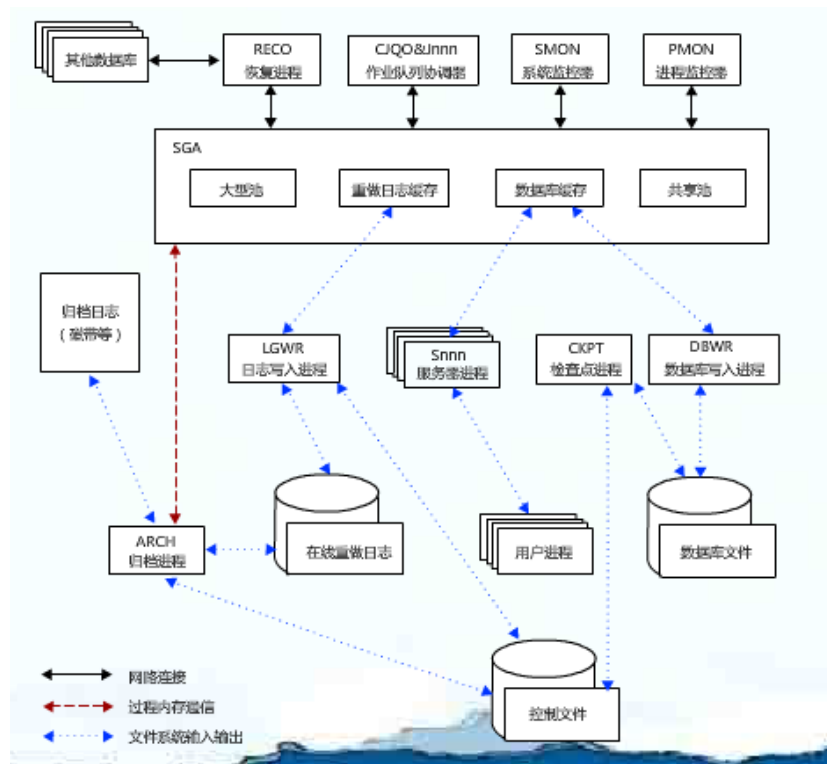
CJQO: 例如：用户可以告诉数据库在每天晚上12:00建立汇总表。通过采用这种方式，就可以在第二天报告汇总信息。作

业队列表中看到需要运行的作业时才会启动。

👤 RECO（恢复进程）进程主要职责：在分布式数据库环境中恢复进程自动恢复那些失败的分式事务。当每个分布式事务由于网络连接故障或者其他原因而失败时，RECO进程将尝试与该事务相关的所有数据库进行联系，以完成对该失败事务的处理工作。

RECO:

进程和内存结构、数据文件间的协作关系



容易混淆的概念

🕒 实例和数据库

🕒 用户和模式

🕒 用户和角色

🕒 数据字典和表

◆ 实例和数据库

🕒 实例：进程+内存结构

🕒 Oracle实例是SGA和后台进程的组合

🕒 数据库只有调入到实例所包括的内存和进程中，才可以使用

🕒 数据库：数据文件+重做文件+控制文件+临时文件等

实例和数据库区别

- 👉 **Oracle实例是SGA（系统全局区）和一组后台进程的组合**
- 👉 **Oracle数据库是指运行数据库所需的所有数据库文件**
- 👉 **Oracle服务启动时先启动实例，然后才装载数据库文件，也就是装载数据库**

◆ 用户和模式

🕒 **用户的传统解释：能够唯一标识一组信任凭证的名称和密码组合**

🕒 **Oracle中的用户其实就是用于登录Oracle的命名帐号**

🕒 **用户可以拥有自己的数据库对象**

🕒 **模式（SCHEMA）是对用户所创建的数据库对象的总称**

- 👉 **在Oracle数据库中任何数据库对象都属于一个特定用户，一个用户及其所拥有的对象即称为模式**

模式中的对象包括 **表、视图、索引、同义词、序列、过程、程序包**

每一个用户都与相同名称的模式相关联。因此模式又称为用户模式。

🕒 **模式本身不是对象，模式只是一个用来描述属于特定用户的对象集合的术语**

🕒 **用户和模式是一一对应的**

◆ 用户和角色

🕒 **用户就是用于登录Oracle的命名帐号**

- 👉 **sys、system、scott、hr**

- 👉 **我们在使用Oracle的时候要为我们的数据建立自己的用户帐号**

```
CREATE USER user1 ... .. ;
```

```
GRANT CREATE SESSION TO user1;
```

```
ROVOKE CREATE SESSION FROM user1;
```

🕒 **Oracle数据库角色就是一组相关权限的命名集合。**

- 👉 **使用角色最主要的目的是简化权限管理**

- 👉 **例如：CONNECT（连接角色） RESOURCE（资源角色） DBA（数据库管理员角色）等等。**

```
CREATE ROLE role1;
```

```
GRANT CREATE SESSION TO role1;
```

```
ROVOKE CREATE SESSION FROM role1;
```


👉 角色是一组权限的组合，我们可以将角色赋给用户，从而让用户拥有角色所拥有的权限。

◆ 数据字典和表

🕒 表是数据库中用来存储数据的对象；包含了行和列。

👉 列用来存储表中特定数据行的数据属性

列的名称 & 数据类型

长度 范围 精度

👉 行包含了在表定义中定义的列值

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON

```
select * from scott.dept;
```

👉 表是其他对象的基础

🕒 数据字典是Oracle存储所有实例信息的表和视图的集合

👉 数据字典的所有者为sys用户，数据存放在System表空间中

👉 数据字典只向用户赋予它们的只读权限

👉 所有的视图都使用了前缀，前缀DBA表示无论执行权限如何，都可以显示所有对象，前缀ALL表示将会显示用户具有权限的所有对象，而前缀USER表示只会显示用户所拥有的对象

👉 数据字典用来记录数据库的系统信息；用户不能修改数据字典中的信息

总结：

🕒 Oracle文件：控制文件、数据文件、参数文件

🕒 Oracle 逻辑结构：表空间、段、区、块

🕒 SGA的功能及其组成

🕒 Oracle主要的后台进程及其功能

🕒 Oracle中容易混淆的概念