Oracle体系结构：

Oracle有自己的文件结构、缓冲区结构、进程结构以及自身的优化功能。

Oracle服务器主要有两部分组成：Oracle数据库软件和Oracle实例。也就是说我们的Oracle数据库只是一个数据文件和相关数据库管理系统的集合，真正为用户提供服务的是Oracle实例。Oracle实例的内存结构组织主要包含在系统全局区（SGA:System Global Area）的内存区域中，SGA可以被所有用户进程共享。

**·Oracle服务器的内存结构：**

Oracle服务器有两种内存结构：系统全局区（SGA）和程序全局区（PGA）

系统全局区是一组共享内存结构，存放一个Oracle数据库实例的控制信息和各个共享用户的数据。程序全局区是用户进程连接到数据库并创建会话时，由*Oracle服务器进程进行分配的、专门用于当前用户会话的内存区*，存储该用户连接期间与SQL语句执行相关的信息，该区域是私有的。

**·Oracle服务器的进程结构：**

Oracle数据库系统中的进程主要分为：用户进程和服务器端进程，服务器端进程又分为服务器进程和后台进程。

用户进程在客户端工作，它向服务器进程发出请求信息。SQL\*Plus、Oracle Forms Builder等都是用户进程。

服务器进程接收用户发出的请求，并根据请求与数据库通信（通过SGA），通过这些通信完成用户进程对数据库中数据的处理请求，而具体的数据读写、日志写入等操作则是由后台进程完成。

**服务器端后台进程：包括数据库写入进程（DBWn）,日志写入进程（LGWR）,日志归档进程（ARCn），检查点进程（CKPT），系统监控进程（SMON）和进程监控进程（PMON）等。**

1. **数据库写入进程**（DataBase Writer，DBWn）：作用是将已经更改的数据块从内存写入到实际的数据文件。默认情况下在数据库实例启动时只启动了一个数据库写入进程，即DBW0，通过设置初始化参数DB\_WRITER\_PROCESSES可以最多定义20个（编号分别是0-9和a-j）数据库写入进程来执行写入操作。从而获得高度动态的数据库性能。在Oracle数据库系统中只有两种进程可以对数据文件执行写入操作：数据库写入进程和检查点进程。
2. **日志写入进程**（Log Writer，LGWR）：作用是将重做日志缓冲区中的内容写入到重做日志中。重做日志是对数据库进行的所有事务的一个拷贝，这样做的目的是使Oracle能从不同的错误中回复。日志写入进程是唯一一个能够读写重做日志的进程。
3. **日志归档进程**（Archive Process，ARCn）：作用是将重做日志的事务变化复制到归档日志文件中。重做事务日志以顺序方式记录，填满一个日志文件之后，就会通过日志切换转向另一个可用的重做事务日志。该后台进程只有在“归档日志”（ARCHIVELOG）模式下才有效。默认情况下只有2个归档日志（ ARC0 和ARC1），通过初始化参数LOG\_ARCHIVE\_MAX\_PROCESSES最多可定义30个日志归档进程。通常情况下，DBA不需要设置，日志写入进程（LGWR）会根据归档任务的需要自动启动适当数量的归档进程。
4. **检查点进程**（Checkpoint Process，CKPT）：作用是发出检查点，实现同步数据库的数据文件、控制文件和重做日志。通俗的讲，就是在发出检查点之后数据库写入进程和日志写入进程都会将缓存中的数据写入到对应的数据文件和重做日志，从而保证数据文件、控制文件和重做日志文件的一致性。
5. **系统监控进程**（System Monitor，SMON）：是在数据库系统启动时执行恢复工作的强制性进程，它负责许多内部的操作。它的作用是执行数据库实例的恢复、释放控件碎片并释放临时段。即当前一次数据库系统异常关闭，再次启动时的恢复工作。在并行服务器模式下，系统监控进程还可以恢复另一台计算机中失败的数据库。
6. **进程监控进程**（Process Monitor，PMON）：作用是监控服务器进程的执行，并在服务器进程失败时清除该服务器进程。进程监控进程是用于恢复失败的数据库用户的强制性进程。
7. **锁进程**（Lock，LCKn）：用于并行服务器模式，帮助数据库通信。
8. **调度进程**（Dispatcher，Dnnn）：在使用多线程服务器时出现，负责所连接的用户进程到可用服务器进程的路由请求，并把响应返回到何时的用户进程。
9. **作业队列进程**（Job Queue，Jnnn）：作业队列进程和作业队列监视进程。10g引入
10. **闪回恢复进程**（Recovery Writer，RVER）：9i提出10g完善的闪回技术。闪回数据库是进行时间点恢复的新方法。能够快速的将Oracle数据库恢复到以前的时间点，以正确更正由于逻辑数据损坏或用户错误引起的任何问题。当需要恢复时，闪回恢复进程可以将数据库恢复到错误前的时间点，并且只恢复改变的数据块。

数据库启动后，我们可以通过系统动态性能监视视图v$bgprocess查看启动了的后台进程，查询代码如下：

conn /@orcl as sysdba --以管理员身份登录

column description format A30 --设置查询列显示格式

set pagesize 200 --设置单页显示条数

select paddr, pserial#,name,description from v$bgprocess order by name

**·服务器进程结构模式：**

Oracle服务器进程结构模式决定了系统如何响应用户进程请求。

1. 专用服务器模式：在专用服务器模式下，Oracle为每一个连接到数据库实例的客户进程启动一个专门的前台服务进程，一个客户进程的前台服务进程只为它的客户进程执行数据库访问操作。

专用服务器模式一般只用在密集的批操作，能让服务器进程大部分时间保持繁忙。当预期客户机连接总数较少，或客户机向服务器发出的请求持续时间较长时，应采用专用服务器模式。局域网环境开发的C/S结构应用系统，采用专用服务器模式有较好的性能。

1. 共享服务器模式：多线程服务器模式是Oracle用于支持客户连接的典型的进程结构，这些服务器端进程合在一起可以有效的支持大量的用户群。其组件包括（调度器、共享服务器进程和队列）。

多线程服务器配置对典型的多用户应用环境非常有效，少数共享服务器进程执行了许多连接客户的数据访问操作，只要有很少的进程开销就可以满足大量的用户群。

采用共享服务器模式时应指定在启动数据库实例时需要创建的服务器进程数，可以通过初始化参数SHARED\_SERVERS设置。

**·Oracle数据库的物理结构：**

Oracle数据库的存储结构包括数据的物理存储结构和逻辑存储结构，物理存储结构是实际的数据存储单元，而逻辑存储结构是数据概念上的组织。

Oracle的物理存储结构主要包括：

1. 一个或多个数据文件（oracle/product/10.2.0/oradata/orcl文件夹下的.DBF文件）
2. 两个或多个重做日志文件（oracle/product/10.2.0/oradata/orcl文件夹下的REDO01.LOG文件）
3. 一个或多个控制文件（oracle/product/10.2.0/oradata/orcl文件夹下的.CTL文件）
4. 其他OS文件

数据文件：用户存储数据库数据，包括系统数据（数据字典）、用户数据（表、索引等）、撤销数据、临时数据等。在Oracle数据库中，至少包含一个数据文件，在逻辑上数据库是由表空间组成，而表空间物理上则是由一个或者多个数据文件组成，一个数据文件包括多个OS上的物理磁盘块。

重做日志文件：用于记录数据库变化，目的是为了在出现数据库实例失败或者介质失败时恢复数据库。Oracle需要至少两个重做日志文件组。当Oracle数据库运行在归档模式时，所有的重做日志文件在被覆盖前都会为其创建一个副本，即对数据库进行的所有事务都有一个备份，是一种最安全的数据库工作方式，并允许扩展恢复功能（包括指定时间恢复），但是这种方式会增加系统的开销。非归档模式下，当一个重做日志文件被覆盖式，之前的所有事务记录则都被清空了，这种模式下只有有限的恢复能力，只要被设计在例行故障时保护数据。

控制文件：用于记录和维护数据库的物理结构。数据库必须包含至少一个控制文件，有与控制文件至关重要，所以应在联机时多保存及格备份，这些文件存储在多个磁盘上，以降低磁盘失效引起的潜在危险。控制文件主要记录：①数据文件的大小、位置②重做日志文件的大小、位置③数据库名、创建时间④日志序列号。

其他文件：

1. 归档日志文件（重做日志文件的备份文件）
2. 参数文件（用于定义数据库实例启动时需要初始化的参数，参数文件默认存放在ORACLE\_HOME/db\_1/database中的.ora文件；文本参数文件的名称格式为inti<SID>.ora，服务器参数文件的名称格式为SPFILE<SID>.ora;SID为数据库实例）
3. 口令文件（存放特权用户信息，特权用户例如SYSDBA,SYSOPER；db\_1/database/pwdorcl.ora）
4. 警告文件（由连续的消息和错误信息组成，按照时间顺序存放；通过警告文件可以查看Oracle内部错误也可以监视特权用户操作，警告文件的位置由初始化参数background\_dump\_dest确定，名称格式为：oracle/product/10.2.0/admin/orcl/bdump/alert\_orcl.log）
5. 后台进程跟踪文件（记载后台进程的警告和错误信息，每个后台进程都相应的跟踪文件。存放位置由初始化参数background\_dump\_dest确定，名称格式为：<数据库实例名>\_<后台进程名>\_<进程对应的OS进程号>.trc）
6. 服务器进程跟踪文件（用于跟踪SQL语句，诊断SQL语句的性能，并做出响应的调整。存放位置由初始化参数user\_dump\_dest确定，名称格式为：<数据库实例名>\_ora\_<进程对应的OS进程号>.trc）

**·Oracle数据库的逻辑结构：**

数据库的逻辑结构是面向用户的，它描述了数据库在逻辑上是如何组织和存储数据的。数据库的逻辑结构支配一个数据库如何使用系统的物理空间。数据库中的数据物理上存储在数据文件中，而逻辑上则存放在表空间中。Oracle数据库中使用表空间、段、区间、数据块等逻辑结构来管理空间对象。层次包含关系如下图：

**数据文件**

**OS磁盘块**

**数据库实例**

**表空间**

**段**

**区间**

**数据块**

1. 表空间（TableSpace）是数据库的逻辑组成部分，将相关的逻辑结构进行分组。一个Oracle10g数据库中至少包含一个SYSTEM表空间和一个SYSAUX表空间，还应该包含数据表空间、索引表空间、临时表空间和UNDO表空间等。一个表空间只能属于一个数据库实例。Oracle数据库中的表空间大部分都是永久表空间，也允许创建临时表空间。表空间的控件信息可以直接保存在数据文件中，也可以保存在数据字典中。
2. SYSTEM表空间：每一个数据库实例至少要包含SYSTEM表空间，该表空间为系统表空间，存储数据库的数据字典，以及数据表以外的其他对象（如：视图、序列等）的定义。

由于SYSTEM表空间被保留用于存放系统信息，因此用户数据对象不应保存在这个表空间中，否则对系统的运行性能和安全造成危害。

1. SYSAUX表空间：在Oracle10g新增的辅助系统表空间，用于减少系统表空间的负荷，提高系统的作业效率。该表空间由系统创建，作为存储数据的默认位置。
2. 非系统表空间：Oracle数据库通常还有几个表空间，用于数据的逻辑和物理存储，下列表空间是大多数数据库必备或者常见的。
3. 撤销表空间 UNDOTBS1，用于保存事务回退信息（rollback）
4. 用户表空间 USERS，存放用户的私有信息
5. 临时表空间 TEMP，用于存放临时表和临时数据
6. 样例表空间 EXAMPLE，包含一些如人力资源、订单输入等的实例方案

实际上系统中只要有SYSTEM,SYSAUX和TEMP表空间就可以正常工作了，查询数据字典DBA\_TABLESPACES可以看到系统中表空间的状态。

1. 表空间在实际工程中的应用：对于一个Oracle数据库应用方案来说，方案用户应当建立自己专用的存放用户数据的数据表空间和索引表空间，同时还应当创建专用的临时表空间。目的是提高数据访问性能，另一方面也便于数据管理、备份、恢复等工作。
2. 表空间的状态和工作模式
3. 联机表空间与脱机表空间：联机表空间中的数据对于数据库而言是可以访问的，脱机表空间的数据是不可访问的，必要时数据库管理员可以将某个表空间（临时）脱机以阻止用户对应用数据的访问。注意：SYSTEM必须始终保持联机，因为数据字典要时刻处于可被访问的状态。
4. 只读表空间与可读写表空间：新创建的表空间总是可读写的。当一个表空间的数据不能被改变时（如用于数据仓库应用的历史数据），可以将其设置为只读表空间。
5. 段（Segment）是一个对象的物理表示，由一个或多个区间组成，用于存储特定对象的所有数据，包含表空间中一种指定类型的逻辑存储结构，段不能够跨表空间，一个段只能属于一个表空间。段的类型：**表、表分区、簇、索引、按索引组织的表、索引分区、回退段、临时段、LOB段、嵌套表**等。可分为数据段、索引段、临时段和回退段。

簇：将多个表的数据按照关键字存储在一起，一个簇表可以包括多个表的数据。

索引：存储表关键字及对应记录的rowid，索引本质上也是一张表。

Lob段：针对大数据

1. 区间（Extent）是Oracle进行空间分配的逻辑单元，是Oracle数据库中最小的存储分配单元。
2. 数据块，也称为Oracle块，是Oracle管理数据文件中存储空间的单位。它是Oracle在数据文件上执行I/O操作的最小单位。可以通过修改初始化参数DB\_BLOCK\_SIZE用于定义标准块尺寸。一个数据块分为了三个区域（Header：保存数据块的地址/表目录/行目录/和为事务保留的空间，FreeSpace：保留用于以后数据更新，Data：），这中设计虽然浪费了一部分空间，但是却换来了更新系统开销的显著减少，从而极大的改善了系统性能。

·**Oracle10g网格基础架构：**

网格计算提供了一个自适应的软件基础架构，它有效地利用了低成本的服务器和模块化存储器，通过增加少量的小型服务器来向外扩展，便可以获得高性能和可靠性。

网格计算消除了应用程序、服务器、数据库、计算机、存储器----网格中的每一个组件之间的固定连接。智能系统也将网格中的所有东西都当作虚拟化的服务，从而能够优化资源的利用率和响应性。与其他计算模型相比，以网格形式设计和实现的系统可以提供更高的可靠性、更低的成本和更大的灵活性。

网格计算基于五个基本的属性：虚拟化、动态供应、资源集中、自适应系统和统一管理。

Oracle的网格计算能力如下：

1. 服务器虚拟：Oracle实时应用集群（RAC）可以是一个数据库运行在多个节点上，将多个计算机的处理资源集中使用，并且Oracle在垮计算机分配工作负载的能力方面具有独特的灵活性。
2. 虚拟存储：Oracle10g的自动存储管理提供了数据库和物理存储之间的一个虚拟层，使得多个此片可以看作是一个单独的磁盘组。在保证数据库在线的情况下，磁盘可以动态的加载和移除。
3. 网格管理：Oracle10g的网格控制特性提供了将多系统集成管理为一个逻辑组的控制台，可以管理网格中独立的节点，集中维护各组系统的配置和安全设置。

**·数据字典：**

在数据库创建期间，Oracle服务器将在数据文件中创建两个类型的对象，他们是数据库字典和动态性能监视视图。

1. *数据字典：*

数据字典主要存放以下几方面的信息：①数据库的物理结构和逻辑结构；②数据库对象的定义和空间的分配情况；③完整性约束；④用户；⑤角色；⑥权限；⑦审计。

各数据字典的前缀含义：

user 记录当前用户所拥有的所有对象信息

all 记录用户可以访问的所有信息，包括user前缀、授权给public以及该用户的所有信息

dba 记录数据库系统中所有数据库对象的信息

1. *动态性能监视视图：*

Oracle服务器包含一组基础视图，这些视图由服务器进行维护，主要记录与数据库系统性能相关的内容，提供的是系统运行过程中的动态信息，它们不允许用户直接进行修改，但是用户可以对这些视图进行查询，以便对系统进行管理及优化。

动态性能监视视图的名称以“V$”为前缀，系统管理员sys可以访问它们。数据字典和动态性能监视视图以及相应同义词的定义都保存在oracle\_home/rdbms/catalog.sql中。

1. *常用数据字典：*

|  |
| --- |
| ***--用户***  *--查看当前用户的缺省表空间*  select username,default\_tablespace from user\_users;  *--查看当前用户的角色*  select \* from user\_role\_privs;  *--查看当前用户的系统权限和表级权限*  select \* from user\_sys\_privs;  select \* from user\_tab\_privs;  *--显示当前会话所具有的权限*  select \* from session\_privs;  *--显示指定用户所具有的系统权限*  select \* from dba\_sys\_privs where grantee='GAME';  *--显示特权用户*  select \* from v$pwfile\_users;  *--查看用户下所有的表*  select \* from user\_tables;  *--查看用户下所有的表的列属性*  select \* from USER\_TAB\_COLUMNS where table\_name=:table\_Name;  *--显示用户信息(所属表空间,用户名区分大小写)*  select default\_tablespace,temporary\_tablespace  　　from dba\_users where username='SCOTT';  *--显示用户的PROFILE*  select profile from dba\_users where username='SCOTT';  *--锁定用户*  alter user scott account lock  *--解锁用户*  alter user scott account unlock  ***--表***  *--查看用户下所有的表*  select \* from user\_tables;  *--查看名称包含log字符的表*  select object\_name,object\_id from user\_objects  　　where instr(object\_name,'LOG')>0;  *--查看某表的创建时间*  select object\_name,created from user\_objects where object\_name=upper('&table\_name');  *--查看某表的大小*  select sum(bytes)/(1024\*1024) as "size(M)" from user\_segments  　　where segment\_name=upper('&table\_name');  *--查看放在Oracle的内存区里的表*  select table\_name,cache from user\_tables where instr(cache,'Y')>0;  ***--索引***  *--查看索引个数和类别*  select index\_name,index\_type,table\_name from user\_indexes order by table\_name;  *--查看索引被索引的字段*  select \* from user\_ind\_columns where index\_name=upper('&index\_name');  *--查看索引的大小*  select sum(bytes)/(1024\*1024) as "size(M)" from user\_segments  　　where segment\_name=upper('&index\_name');  ***--序列号***  *--查看序列号，last\_number是当前值*  select \* from user\_sequences;  ***--视图***  *--查看视图的名称*  select view\_name from user\_views;  *--查看创建视图的select语句*  set view\_name,text\_length from user\_views;  set long 2000; 说明：可以根据视图的text\_length值设定set long 的大小  select text from user\_views where view\_name=upper('&view\_name');  ***--同义词***  *--查看同义词的名称*  select \* from user\_synonyms;  ***--约束条件***  *--查看某表的约束条件*  select constraint\_name, constraint\_type,search\_condition, r\_constraint\_name  　　from user\_constraints where table\_name = upper('&table\_name');  select c.constraint\_name,c.constraint\_type,cc.column\_name  　　from user\_constraints c,user\_cons\_columns cc  　　where c.owner = upper('&table\_owner') and c.table\_name = upper('&table\_name')  　　and c.owner = cc.owner and c.constraint\_name = cc.constraint\_name  　　order by cc.position;  ***--存储函数和过程***  *--查看函数和过程的状态*  select object\_name,status from user\_objects where object\_type='FUNCTION';  select object\_name,status from user\_objects where object\_type='PROCEDURE';  *--查看函数和过程的源代码*  select text from all\_source where owner=user and name=upper('&plsql\_name'); |