Oracle数据字典以三种形式的对象类型存在：基表、用户访问视图、同义词

基表：

存储数据库及其对象的定义信息，

大部分数据是加密存储的，

只有数据库本身才能读写这些基表，

用户不直接访问。

用户访问视图：

是用户平时看到的数据字典，

是基于基表创建的，

汇总了基表内的信息，

以可读的形式提供给用户使用。

为了管理方便，Oracle又把它们分为几组不同的类型，以及不同的前缀命名，

有的只有dba才能访问，有的可以供所有用户访问，但不同的用户访问到的数据却有所不同。

同义词：为了方便用户访问，Oracle为大部分用户访问视图创建了同名的public同义词。

Oracle用户访问视图的名称由前缀和后缀组成，使用\_连接，其代表的含义如下：

Dba\_：系统管理员视图，整个数据库的全局视图，包含所有用户所拥有的模式对象的定义信息，只有数据库管理员才能访问。

User\_：用户视图，只包含当前用户所拥有的模式对象的定义信息，所有用户都可以访问，但不同用户所看到的数据可能不同。

All\_：扩展视图，除了包含的当前用户所拥有的模式对象的定义信息，还包含当前用户被授权访问的其他用户所拥有的模式对象的定义信息。

动态性能视图：存储在内存中

也是由基表、视图和同义词组成

基表：前缀：X$，不像常规表一样存储在数据库中。而是构建在Oracle实例的内存结构中，所以是虚拟表，不允许普通用户访问基表

视图：前缀V\_$，是基于基表创建的动态性能视图，别称V$视图，只有sys用户才能访问，与常规标不同，其结构定义及其基表中的数据都不能由用户修改，因此动态性能视图又称为固定表。

同义词：前缀V$，Oracle为V$视图创建的public同义词，解决了只有sys用户才能访问V\_$的问题，数据库管理员及其它用户，一般是通过同义词访问视图的。

动态性能视图和静态数据字典的区别：存储位置不同、命名形式不同、内容更新及频率和可访问时间不同。

常用数据字典：

就是英语翻译，例如：

Dba\_tablespaces：所有表空间

dba\_free\_space：空闲空间

dba\_segments：所有段分配的存储空间的信息

dba\_extents：所有区

dba\_data\_files：所有数据文件

dba\_temp\_files：所有临时文件

dba\_ts\_quotas：列出用户在表空间上分配的限额，ts是tablespace，quota是限额

dba\_objects：列出所有数据库对象

dba\_source：所有源代码

后面都大差不差，直接翻译就行了。Privs是权限，全名是privilege，特权的

常用动态性能视图：V$开头

V$version：版本号

前缀基本是这样的，会出选择题

Option选项功能是否可用

Database数据库信息

Instance实例状态信息

Process进程信息

Bgprocess后台进程信息

Fast\_start\_transactions实例恢复的进度

Sga显示sga汇总信息，查询分配的内存总量

Sgainfo显示sga不同组件的大小，粒度尺寸和空闲内存量

动态性能视图本质上是一个观察数据库状态的观察者

数据库实例有哪几部分组成：（sga和进程集合组成）sga、pmon、smon、dbwr、lgwr、ckpt组成

数据库的存储结构是逻辑存储结构和物理存储结构

Oracle系统体系结构：逻辑结构、物理结构和实例

pmon、smon、dbwr、lgwr、ckpt是用来将数据保存到参数文件、密码文件、数据文件、控制文件、重做日志文件、归档日志文件中的进程。后面会有解释。。。。。。

表空间：是一个逻辑存储容器，位于逻辑存储结构的顶层，用于存储数据库中的所有数据。表空间内的数据被物理存放在数据文件中，一个表空间可以包含一个或者多个数据文件。

Oracle创建表空间会默认创建这些表空间以及这些表空间的作用：

System：系统表空间，主要用于存放Oracle系统内部表和数据字典的数据。System01.dbf

Sysaux：system表空间的辅助表空间，用于存储一些组件和产品的数据，以减轻system表空间的负载。Sysaux01.dbf

Temp：临时表空间，用来存储SQL语句处理过程中产生的临时表和临时变量，用于排序和汇总等。Temp01.dbf

Undotbs1：还原表空间，Oracle数据库用它存储还原信息，实现回滚操作等。Undotbs01.dbf

Users：用于存储永久用户对象和数据，可以在这个表空间上创建各种数据对象。Users01.dbf

段就是占用存储空间的数据库对象。区是数据库存储单元的最小单位，范围是一个数据块到2GB

数据库物理文件：数据文件，控制文件，重做日志文件，参数文件，口令文件，警告日志文件，跟踪文件。

重做日志文件：记录了数据库产生的所有变化信息。在实例或者介质失败时，可以用重做日志恢复数据库。如果数据库运行在归档模式下，在发生日志文件切换后，填满的重做日志文件被复制到其他地方保存，这些日志文件副本被称为归档日志文件。

物理存储结构和逻辑存储结构的关系：

1. 一个表空间可以包含一个或多个数据文件，在一个表空间内可以存储一个或多个段，所以段数据可以存储在一个数据文件上，也可以存储在一个表空间内的多个数据文件上。
2. 每个段中包含一个或多个区，每个区由一个或多个数据块组成。
3. 向段分配数据文件内的空闲空间是以区为单位的。
4. Oracle数据块是操作系统块的整数倍。一个表空间内的所有数据文件只能使用同样的块尺寸

Oracle数据库的实例由两部分组成：sga系统全局区和进程集合组成

Sga系统全局区：是所有用户进程共享的一块内存区域。可以减少io次数，提高系统性能。

进程集合：共同实现对Oracle数据库的管理功能。

高速数据缓冲区对应进程：dbwr

重做日志缓冲区对应进程：lgwr

Exp：

数据库写入进程：dbwr

日志写入进程：lgwr

归档进程：arch可选

检查点进程：ckpt

进程监视进程：pmon

系统监视进程：smon

数据库实例的启动与关闭：

打开和关闭数据库需要有sysdba或者sysoper管理权限

数据库状态：close、nomount、mount、open

SQLplus使用startup开启实例，可以让数据库从close状态直接启动到其他三种状态，startup nomount，startup mount，startup open（或者startup），但是启动之后不能使用startup来改变状态，只能使用alter命令，例如alter database open

这么多状态是为了让dba执行相应的管理工作

Nomount状态：可以打开初始化参数文件

Mount：打开控制文件

Open：打开数据文件、重做日志文件

Close：可以编辑和修改初始化参数文件

Nomount：可以创建新的数据库，或者重新创建控制文件。

Mount：重命名或移动数据文件、恢复数据库、改变数据库归档模式

Open：分为受限和不受限

不受限：普通用户可以访问

受限：只有具有create session和stricted session才能访问数据库，管理员也不能远程访问，主要用于执行数据的导入导出操作，数据的装载，数据库的迁移和升级操作

没启动数据库之前：startup的restrict和nomount、mount和open组合使用，就可以启动到受限模式，eg：startup nomount restricted session

启动之后：alter system enable/disable restricted session

数据库的关闭：

使用shutdown命令，四种方式：normal，transactional，immediate，abort

normal：正常关闭

transactional：事务关闭

immediate：立即关闭

abort：异常关闭

正常关闭eg：shutdown normal正常关闭，normal可以省略

关闭之后禁止新建连接，并等待已连接的用户主动断开之后再关闭数据库，不主动断开就会导致该命令超时。

事务关闭：禁止新建连接，禁止已连接的用户启动事务，但会等待已启动的事务执行完成，然后断开用户连接，关闭数据库。

立即关闭：禁止新建连接，禁止已连接的用户启动事务，未提交的事务会立即回滚，然后关闭连接关闭数据库。处于一致性状态。

异常关闭：abort，通过立即终止数据库实例的方式关闭数据库，禁止新建连接，禁止已连接的用户启动新的事务，正在执行的SQL语句立即中止，不回滚未提交的事务，也不执行检查点，直接关闭数据库实例。因此这种方式数据库会处于不一致状态，下次启动需要做实例恢复。

初始化参数文件有两种：文本初始化参数文件，服务器参数文件。

控制文件是一个二进制文件，它记录Oracle数据库的当前状态以及物理结构信息，其中包括（但不仅限于）：

1. 数据库名称、数据库的唯一标识（DBID）以及数据库创建的时间戳
2. 组成数据库的表空间信息和数据文件信息，每个数据文件的存储路径和文件名。
3. 联机重做日志文件的名称和位置。
4. 归档日志文件的名称和位置。
5. 当前日志序列号
6. 控制点信息等。

Lgwr什么情况下会将重做日志缓冲区中的重做日志写入重做日志文件中：

1. 用户提交事务时
2. 联机重做日志切换时
3. Lgwr上次写入3s后
4. 重做日志缓冲区达到1/2满，或者缓存的重做日志达到1mb时
5. Dbwn把脏数据块写入数据文件之前。

归档进程与归档重做日志文件：

数据库可以在两种模式下运行：archivelog归档模式，noarchivelog非归档模式

区别：在lgwr需要重新使用联机重做日志文件时，在归档模式下，只有在原来的重做日志得到复制归档之后，lgwr才能重新使用该组重做日志文件。而在非归档模式下没有限制，lgwr在需要重新使用重做日志文件时可以直接覆盖原来的重做日志。

查看Oracle的运行模式：archive log list，也可以使用动态性能视图v$database的log\_mode列。

Oracle可以运行启动30多个归档进程，ARCn，n是编号，可以是0-9,和a-t

归档模式下，重做日志文件没有归档，会导致数据库运行被暂时挂起。

非归档模式下不会，如果需要使用之前的重做日志文件，直接覆盖掉当前的重做日志文件即可

归档分为手工和自动两种，自动归档启动后，后台进程ARCn在日志切换后自动完成归档操作。使用手工归档时，只有具有管理员权限，并且数据库处于mount或open状态才能归档。手工归档SQL语句：alter system archive log all/next，all表示将所有没有归档的重做日志文件全部归档，next只会归档下一组。

Alter system enable/disable restricted session

自动归档模式下，alter system archive log logfile ‘文件名’ to ‘位置’也能进行手动归档，前提是这个重做日志文件是inactive状态。

添加重做日志文件成员代码：

Alter database 数据库名 add logfile member ‘文件’，‘文件’ to group 1

移动、重命名重做日志文件代码：

1. 关闭数据库：shutdown immediate
2. 使用命令移动或者重命名日志文件：move 文件1 文件2
3. 启动到mount状态：startup mount
4. 修改控制文件：alter database rename file 文件1 to 文件2
5. 完全打开数据库：alter database open

管理归档重做日志文件：管理归档重做日志文件使用的命令都是alter system set

管理涉及设置归档日志的位置，归档日志文件的命名方法和设置归档进程数量等。

设置归档位置：在管理归档日志时，需要指定日志文件的归档位置。Oracle可以将重做日志归档到一个或多个位置，归档位置既可以是本地文件系统，Oracle数据库的快速恢复区，也可以是OracleASM磁盘组或者是远程Oracle数据库（备用数据库）。

这个归档位置由初始化参数指定，dba可以在创建数据库的时候设置相应的初始化参数规划好归档位置，也可以在数据库运行期间通过alter system语句动态修改初始化参数，改变归档日志的位置。这样修改之后只有在下一次日志切换时才能改变归档位置。

指定归档位置的参数：

1、LOG\_ARCHIVE\_DEST：主归档位置

LOG\_ARCHIVE\_DUPLEX\_DEST：辅助位置，可选项

这两个参数只能指定本地文件系统。并且指定的归档位置不能超过两个。

Eg：alter system set log\_archive\_dest = 文件位置；

alter system set log\_archive\_duplex\_dest = 文件位置；

2、log\_archive\_dest\_n：n的取值是1-31的整数，取1-10时，用于指定本地或远程归档位置，取11-31时，只能用于指定远程归档位置。使用这个参数需要配合location和service使用。Location是设置本地归档位置，service是设置远程归档位置。

本地文件系统：alter system set log\_archive\_dest\_1= ’location=位置’

OracleASM磁盘组：alter system set log\_archive\_dest\_1= ’location=+dgroup1’

快速恢复区：alter system set log\_archive\_dest\_1= ’location=use\_db\_recovery\_file\_dest’

远程Oracle数据库：alter system set log\_archive\_dest\_1= ’service=数据库名称’

设置归档日志文件命名格式：

作用：保证归档时能给每个文件唯一的文件名，也是通过初始化参数设置。参数是log\_archive\_format。这个参数有默认值，是arc%s\_%r.%t，可以保证各个归档日志文件名的唯一性。%S表示在文件名中包含日志序列号，%R表示包含重置日志编号，%T表示包含线程编号。长度固定，如果长度不一，在前面添0，如果是小写就不会固定长度。

Eg：alter system set log\_archive\_format = ‘%R %T%S.arc’ scope=spfile

不能动态修改，设置之后需要重新启动数据库实例才能生效。

调整归档进程数量：

设置初始化参数log\_archive\_max\_processes，11.2版本中默认值是4，如果满足不了重做日志归档的要求，Oracle会自动启动额外的归档进程。

该参数的取值范围是1-30的整数，是动态参数，不需要重新启动。

Eg：alter system set log\_archive\_max\_processes = n

改变归档模式：

要改变归档模式，需要处于mount状态。

Alter database noarchivelog 非归档模式

Alter database archivelog 自动归档模式

Alter database archivelog manual 手动归档模式

手动归档：alter system archivelog all/next

Big eg：

在线设置归档位置，文件命名格式：

使用sysdba连接数据库，然后修改初始化参数：

整体步骤：设置参数，关闭数据库，打开数据库到mount状态，设置归档模式，打开到open

alter system set log\_archive\_dest = 文件位置；

alter system set log\_archive\_duplex\_dest = 文件位置；

alter system set log\_archive\_format = ‘%R %T%S.arc’ scope=spfile；

然后关闭数据库：shutdown immediate

启动数据库到mount状态：startup mount

设置数据库为自动归档：alter database archivelog

打开数据库：alter database open，注意这里不能用startup命令，因为数据库已经启动了，只是是mount状态，需要改成open状态要用alter database命令

最后检查是否能够正确归档：alter system Switch logfile；强制切换日志，触发日志文件的归档。

归档模式的优点：

1. 无论出现实例故障还是介质故障，利用数据库备份，以及联机和归档重做日志文件能够确保恢复所有已提交的事务。
2. 能够在数据库打开和正常使用的情况下进行备份
3. 能将归档日志应用到备用数据库，可以使其与原数据库保持同步

Big eg：

转为非归档模式操作：

管理员权限登录，关闭数据库：shutdown immediate

启动到mount状态：startup mount

修改为非归档模式：alter database noarchivelog

打开数据库：alter database open

非归档模式下，重做日志组状态变为inactive后，即可被lgwr重新使用。只要重做日志文件完好，出现实例故障能实现实例恢复，但是介质故障时只能使用最近一次所做的数据库完整备份恢复到备份时的状态，由于没有归档日志，所以最近一次备份后所提交的事务无法恢复。除此之外在非归档模式下，也不能执行联机表空间备份。

查询归档重做日志相关的信息：

V$database：log\_mode列值说明数据库模式

V$archive\_processes：显示归档进程的状态信息

V$archived\_log：显示历史归档日志信息，取自数据库的控制文件

V$archived\_dest：显示当前实例的所有归档目标位置，以及它们当前值、模式、状态等

V$log：显示数据库的所有重做日志组，指出那些需要归档

使用方法：

Select log\_mode from V$database;

Select \* from v$archive\_processes

Select name,sequence#,completion\_time from v$archive\_log

Select dest\_id,dest\_name,destination from v$archive\_dest

Select group#,archived,status from v$log

表空间分类：临时表空间、永久表空间、还原表空间。

创建表空间的方式和语句：

可以使用oem创建表空间，也可以使用SQL语句。

SQL语句：create database

创建数据库的时候会自动创建system和sysaux两个永久表空间，一个临时表空间temp和一个还原表空间undotas1.还有一个users表空间。

SQL语句2：create tablespace

语法：

Create [bigfile/smallfile] tablespace [datafile 数据文件定义，数据文件定义]

[blocksize 整数[k]] [logging/nologging] [force loging] [online/offline]

[区存储管理子句] [段管理子句]

Extent management dictionary/local管理方式 autoallocate/uniform分配方式

Segment space management auto/manual分配方式

Autoextend on next 大小 maxsize unlimited

Bigfile：表空间类型，只能包含一个数据文件，最多容纳4G(232)个数据块，当表空间数据块大小是8k的时候，改表空间的存储容量可达32TB。bigfile表空间的区不能采用字典管理方式。

Smallfile是Oracle传统表空间，最多包含1022个数据文件，每个数据文件可以容纳4m(222)个数据块。缺省默认表空间。

Datafile：定义数据文件，指定数据文件的名称和位置

Blocksize：指定非标准数据块大小，单位是千字节kb，就是不使用默认的数据块大小，使用我指定的大小。

Logging/nologging：日志属性，logging是默认设置，要求把数据库对象的创建及操作日志写入重做日志文件中。Nologging相反，无日志会改善性能，但是无法进行介质恢复。

Force logging：忽略logging/nologging参数，进入强制日志模式，该表空间内所有对象的所有修改都会记录到重做日志文件中。

Online/offline：指定表空间创建之后是联机还是脱机状态。默认联机

区/段管理子句：指定区/段的存储分配情况。

Eg：create tablespace 表空间名 datafile ‘数据文件路径’ size 大小 reuse

Autoextend on next 1280k maxsize unlimited;

Reuse表示如果表空间存在就覆盖，autoextend表示自动扩展大小，next指定每次扩展1280k，最大不限制。

区管理方式代码：

create tablespace 表空间名 datafile ‘数据文件路径’ size 大小

extent management 管理方式 分配方式 size 大小

大小单位可以是k、g、m、t

管理方式：

字典管理dictionary：区的分配信息集中放在数据字典中

本地管理local：在表空间自身内用位图记录区的分配信息。建议使用这个，降低system表空间的并发访问，减少IO争用，提高性能，不需要合并空闲区碎片。

分配方式：

Autoallocate自动分配：默认，每次分配区的最小尺寸是64kb

Uniform统一分配：就是指定每次分配区的大小，如果省略size关键字每次分配区为1m

段空间管理代码：

create tablespace 表空间名 datafile ‘数据文件路径’ size 大小

segment space management auto/manual

段空间管理方式有两种，一种是自动一种是手动。

自动是默认，使用位图的方式管理表空间内空闲空间。手动管理方式数据库将用空闲列表方式管理表空间段内的空闲空间。只有在创建本地永久表空间的时候才能使用段空间管理子句。但是system表空间时候不能使用。

脱机表空间：

数据库打开状态能改变表空间的状态，从联机到脱机，从脱机到联机。

需要将表空间脱机的情况：

1. 重命名或移动表空间的数据文件
2. 执行脱机表空间备份
3. 在升级或维护应用程序过程中临时关闭其对应的表空间

在脱机过程中部分数据可用，部分数据不可用。并且system表空间、临时表空间和还原表空间不能脱机。

脱机表空间语句：

Alter tablespace 表空间名 offline [normal/temporary/immediate]

Normal正常脱机：脱机前对表空间内的所有数据文件执行检查点，并检查所有数据文件成功写入后才能成功脱机。默认设置，下次联机时不需要做介质恢复。

Temporary临时方式脱机：脱机前执行检查点，但不检查数据文件是否成功写入。所以下次联机之前需要看情况是否做介质恢复，如果脱机的时候数据文件写入成功，以后联机表空间的时候不需要做介质恢复，否则在联机表空间之前需要对写入失败而脱机的数据文件做介质恢复。

Immediate立即脱机方式：不会在数据文件上执行检查点，更不检查数据文件。使用这个参数脱机的表空间下次联机一定要对数据文件做介质恢复，所以数据库运行在noarchivelog模式，就不能用这种方式脱机表空间。

联机表空间：

如果使用的是temporary或者immediate脱机，并且出现了需要介质恢复的情况，需要执行

Recover datafile 文件编号或者文件路径 语句来进行介质恢复。文件编号查看方法：无法正常联机，会提示文件几需要介质恢复。恢复之后重新联机表空间即可。

Alter tablespace 表空间名 online；

创建临时表空间代码：

Create [smallfile/bigfile] [default] temporary tablespace 临时表空间名 tempfile ‘数据文件路径’ size 大小 reuse

段/区空间管理子句

Extent management local/dictionary autoallocate/uniform

Segment space management auto/manual

上面的default关键字表示设置创建的临时表空间为默认的临时表空间。

Alter database default temporary tablespace 他妈的就是直接翻译

设置默认临时表空间：alter database default temporary tablespace 临时表空间名;

还原表空间：看一遍，有个印象就行了

Undo数据使用情况：回滚事务，恢复事务，提供读一致性，闪回查询闪回事务处理和闪回表

添加数据文件：

Alter tablespace 表空间名

Add datafile ‘数据文件’ size 大小 reuse 区/段空间管理子句

非归档模式脱机数据文件：

非归档模式脱机之后不能再联机，调用语句：alter database datafile ‘数据文件名’ offline for drop，没有删除数据文件，数据文件仍然保留在数据字典中，要删除就需要alter database drop datafile语句或drop tablespace 表空间名 including contents and datafiles

归档模式下脱机联机数据文件：

Alter database datafile 数据文件名或者文件号 offline/online

查看数据文件状态：select status from v$datafile

如果是recover就说明需要介质恢复

Recover datafile 文件编号或者文件路径

创建用户代码：

create user 用户名 identified { by 密码 / externally / globall }

[password expire] [account lock/unlock]

[temporary tablespace 表空间] [default tablespace 表空间]

[quota 大小/unlimited on 表空间] [profile 概要文件]

Identified是验证方式，有三种：口令验证，外部验证，全局验证

Password expire表示第一次登录必须修改密码

Account表示用户的账户状态是否锁定，默认开放

Temporary tablespace表示指定临时表空间

Default tablespace表示指定默认表空间

Quota表示在指定的表空间上可为用户分配多大内存

Profile为用户指定概要文件，以限制分配给用户的数据库资源量

修改用户：

Alter user 用户名

Identified by 新密码

Account lock/unlock

Quota 大小 on 表空间

Password命令：password 用户 旧密码 新密码。如果不指定用户就是修改当前用户密码

禁止用户访问优先考虑锁定用户，删除用户需要带上cascade

Drop user 用户 cascade

授权操作：

授予系统权限：

Grant 系统权限 to user/public [with admin option]

表示授予指定权限给指定用户或者所有用户，with admin option表示将授权的权限同时授予指定用户。

授予对象权限：

Grant 对象权限(列名) on 对象名称 to 角色 [with grant option] with admin option

Eg: grant update(username) on student to fucker with admin option

还能指定列名，表示指定对象的指定列名的指定权限，with grant option表示将对象权限授予别人，admin表示将系统权限授予别人。

创建角色代码：

Create role 角色名 [no identified / identified { by 口令 / externally }]

Not identified表示该角色使用数据库认证方式，在启动时不需要口令（set role）

Identified表示调用set role激活角色时，用户必须按照指定的方式去验证口令，也能使用口令验证，全局验证，外部验证。

给角色授权代码：

Grant 系统权限/对象权限/角色 to 角色/用户/public with admin option

Admin表示被取消的权限不是级联的，grant表示被取消的权限是级联的。