

13

存储结构和关系

ORACLE®

Copyright © Oracle Corporation, 2001. All rights reserved.

目标

完成这一课的学习后，您应该能达到下列目标：

- 描述数据库的逻辑结构
- 列出段类型及其用途
- 列出控制块空间使用率的关键字
- 从数据字典获取有关存储结构的信息

ORACLE®

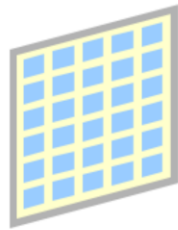
存储和关系结构

Database										test																	
Tablespace																											
system				sysaux						users						undo1				temp							
datafiles				d:\test\						d:\test\			d:\test\			d:\test\			d:\test\								
d:\test\sys01.dbf				sysaux01.dbf						user01.dbf			user02.dbf			undo01.dbf			temp01.dbf								
d.d.	d.d.	Rb	Rb	Log	strea	Inter	Ultra	Dept	Emp		emp	dept		Rb1	Rb2	Rb3	T1	t2									
table	Index	seg	seg	miner	m	media	search	Data	Data		(cont)	Index		Rb	Rb	Rb	Temp	Temp									
Data	Index							seg	seg		Data	Seg		seg	seg	seg	seg	Seg									
seg	seg										seg																
extents																											
orac				le				da				ta				base				blo				cks			

数据库体系结构

上一课讨论了数据库、其表空间和数据文件的存储结构。本课通过分析段、区和数据块来继续讨论数据库存储。

段类型



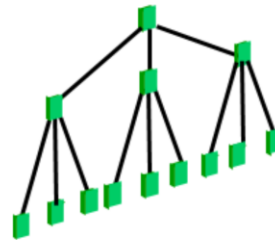
表



表分区



簇



索引

ORACLE

9-4

Copyright © Oracle Corporation, 2001. All rights reserved.

段类型

段是数据库内占用空间的对象。它们使用数据库中数据文件内的空间。本部分介绍不同类型的段。

表：

表是在数据库内存储数据的最常用方法。表段用于存储非集簇且未分区的表中的数据。表段中的数据并不按特定顺序存储，因此，数据库管理员 (DBA) 很难控制表中块内行的位置。表段中的所有数据都必须存储在一个表空间内。

表分区：

当数据库内表的并发使用率很高时，主要关注点将是伸缩性和可用性。在这种情况下，表内数据可以存储在几个分区内，每个分区驻留在不同的表空间。Oracle 服务器当前支持通过键值范围、散列算法以及值列表来分区。表分区后，每个分区都是一个段，可以指定存储参数单独对它们进行控制。

段类型（续）

簇：

簇与表一样，是一种数据段类型。簇内的行是基于键列值存储的。一个簇可以包含一个或多个表。一个簇内的表属于同一个段并且共享相同的存储特性。可以通过索引或者散列算法来访问集簇表内的行。

索引：

一个特定索引的所有条目都存储在一个索引段内。如果一个表有三个索引，则使用三个索引段。使用索引段的目的是根据指定的关键字来查找行在表内的位置。

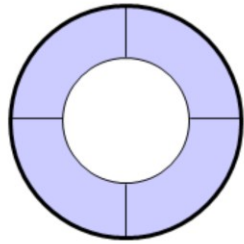
段类型



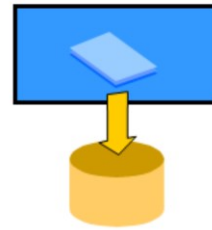
按索引组织的表



索引分区



还原段



临时段

段类型

按索引组织的表:

在按索引组织的表内，数据基于键值存储在索引内。按索引组织的表无需在表中进行查找，因为所有数据都可以直接从索引树中检索到。

索引分区:

索引可以分区并跨多个表空间。在这种情况下，索引内每个分区都对应一个段并且无法跨越多个表空间。分区索引的主要用途在于，通过分散索引输入/输出 (I/O) 来最大限度地降低争用。

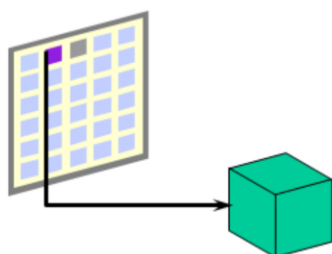
还原段:

还原段由正在对数据库进行更改的事务处理使用。在更改数据或者索引块之前，旧值存储在还原段内。因此，允许用户还原所做的更改。

临时段:

当用户执行 CREATE INDEX、SELECT DISTINCT 和 SELECT GROUP BY 等命令时，Oracle 服务器就会在内存中执行排序。如果排序所需空间大于内存中的可用空间，则将中间结果写入到磁盘上。临时段用来存储这些中间结果。

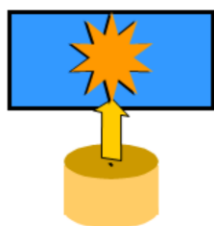
段类型



LOB 段



嵌套表



引导程序段

ORACLE

9-7

Copyright © Oracle Corporation, 2001. All rights reserved.

段类型

LOB 段：

表中的一列或者多列可以用来存储大型对象 (LOB)，如文本文档、图像或者视频。如果列很大，Oracle 服务器将把这些值存储在独立的段（称为“LOB 段”）中。表中只包含一个定位器或者指针，指向对应的 LOB 数据所在的位置。

嵌套表：

表中的列可以由用户定义的表构成，如订单中的项。在这种情况下，内表（即嵌套表）将存储为独立的段。

引导程序段：

引导程序段，也称为高速缓存段，是在创建数据库时由 sql.bsq 脚本创建。在例程打开数据库时，该段可帮助初始化数据字典高速缓存。

引导程序段无法查询或者更新，并且不需要数据库管理员维护。

其它段类型

- 表子分区
- 索引子分区
- 大对象索引
- 大对象分区
- 大对象子分区
- 新回滚段(TYPE2 UNDO)

ORACLE

9-8

Copyright © Oracle Corporation, 2001. All rights reserved.

表子分区：表的二级分区

索引子分区：索引的二级分区

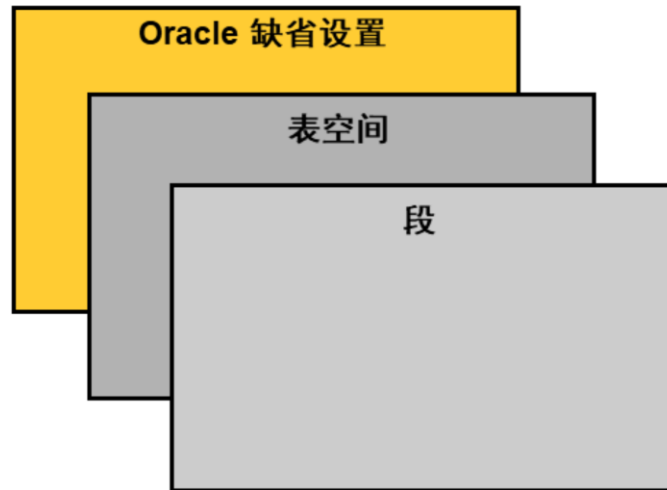
大对象索引：有大对象段就会生成此段

大对象分区：对大对象字段可分区

大对象子分区：对大对象字段可进行二级分区

新回滚段(TYPE2 UNDO)：原回滚段为ROLLBACK，新回滚段为这个类型

存储子句优先级



ORACLE

9-9

Copyright © Oracle Corporation, 2001. All rights reserved.

存储参数

可以在段级别指定存储子句，以控制如何向段分配区。

- 除了 MINIMUM EXTENT 和 UNIFORM SIZE 表空间参数外，在段级别指定的任何存储参数覆盖在表空间级别设置的相应选项。
- 如果没有在段级别明确设置存储参数，那么存储参数缺省设置为在表空间级别所设置的值。
- 如果没有在表空间级别明确设置存储参数，那么将使用 Oracle 服务器系统的缺省设置。

其它注意事项：

- 如果存储参数改变，新选项只适用于尚未分配的区。
- 某些参数无法在表空间级别指定，而只能在段级别指定。
- 如果指定了表空间的最小区大小，则该大小适用于将来分配给该表空间内段的所有区。

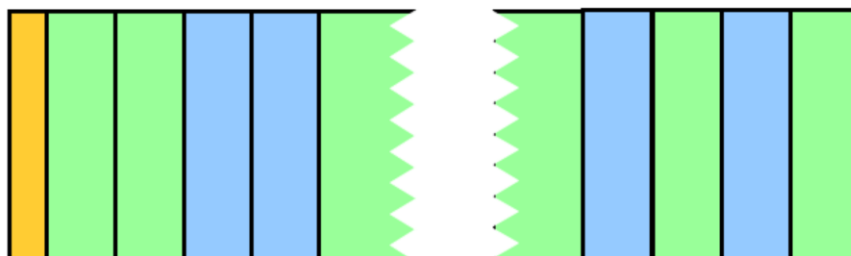
区的分配与回收

- 区是表空间内某个段使用的一块空间。
- 当段处于以下情况时分配区：
 - 已创建
 - 已扩展
 - 已改变
- 当段处于以下情况时回收区：
 - 已删除
 - 已改变
 - 已截断

ORACLE

已用区和空闲区

数据文件



文件标头



已用区



空闲区

ORACLE

9-11

Copyright © Oracle Corporation, 2001. All rights reserved.

区

创建表空间时，表空间中的数据文件包含一个标头，它是文件中的第一个块或前几个块。创建段时，从表空间的空闲区为这些段分配空间。段所使用的连续空间称为“已用区”。当段释放空间时，将把所释放的区添加到表空间的可用空闲区池中。

数据库块

- I/O 的最小单位
- 由一个或多个操作系统块组成
- 在创建表空间时设置
- `DB_BLOCK_SIZE` 指定了缺省块大小

ORACLE

多种块大小支持

- 数据库既可以按照标准的块大小创建，也可以按照非标准的块大小创建，非标准的块大小最多为四种。
- 块大小可以是介于 **2 KB** 和 **32 KB** 之间的 **2** 的幂值。

ORACLE

9-13

Copyright © Oracle Corporation, 2001. All rights reserved.

多种块大小支持

Oracle 支持创建具有多种块大小的数据库。此功能在下列情况下非常有用：

- 将表空间从联机事务处理 (OLTP) 数据库传送到企业数据仓库时。使用 Oracle，可方便地在具有不同块大小的数据库之间传送数据
- 要求能够在具有相应块大小的表空间中定位对象以最大限度地提高 I/O 性能时

SYSTEM 表空间的块大小指的是标准块大小。它是在创建数据库时设置的。使用 Oracle，除了标准的块大小外，还可以指定最多四种非标准的块大小。在初始化文件中，可以在缓冲区高速缓存中为每个这样的块大小配置子高速缓存。也可以在例程运行过程中配置子高速缓存。可以创建具有其中任意块大小的表空间。标准块大小用于系统表空间和大多数其它表空间。

标准块大小

- 在创建数据库时使用 **DB_BLOCK_SIZE** 参数设置；除非重新创建该数据库，否则无法更改
- 用于 **SYSTEM**、**SYSAUX**、还原、临时表空间
- **DB_CACHE_SIZE** 指定标准块大小的 **DEFAULT** 缓冲区高速缓存大小：
 - 最小大小 = 一个粒组 (4 MB、8 MB、16 MB)
 - 缺省值 = 48 MB

ORACLE

9-14

Copyright © Oracle Corporation, 2001. All rights reserved.

标准块大小

DB_BLOCK_SIZE 初始化参数用于指定数据库的标准块大小。该块大小用于 **SYSTEM** 表空间，以及任何临时表空间。除非进行指定，否则，标准块大小还用作表空间的缺省块大小。Oracle 最多支持四种附加的非标准块大小。

应该将最常用的块大小设置为标准块大小。在很多情况下，这是唯一需要指定的块大小。通常，将 **DB_BLOCK_SIZE** 设置为 4 KB 或 8 KB。如果未指定，则使用缺省数据块大小，缺省数据块大小取决于所用的操作系统，并且通常是合适的块大小。

除非重新创建数据库，否则在创建数据库后将不能更改块大小。

DB_CACHE_SIZE 初始化参数替代以前版本中使用的 **DB_BLOCK_BUFFERS** 初始化参数。**DB_CACHE_SIZE** 参数用于指定标准块大小缓冲区的高速缓存大小，其中的标准块大小是由 **DB_BLOCK_SIZE** 指定的。

为了保持向后兼容性，**DB_BLOCK_BUFFERS** 参数仍然可以使用，但它是一个静态参数，不能与任何可动态调整大小的参数一起使用。

注：粒组是一个连续虚拟内存分配单位。粒组的大小取决于估算的 SGA 的总大小，这个总大小是根据 **SGA_MAX_SIZE** 的参数值计算的：如果估算的 SGA 的大小 < 128 MB，则为 4 MB；否则为 16 MB。

非标准块大小

- 使用以下动态参数配置附加高速缓存：
 - DB_2K_CACHE_SIZE 用于 2 KB 块
 - DB_4K_CACHE_SIZE 用于 4 KB 块
 - DB_8K_CACHE_SIZE 用于 8 KB 块
 - DB_16K_CACHE_SIZE 用于 16 KB 块
 - DB_32K_CACHE_SIZE 用于 32 KB 块
- 如果 nK 是标准块大小，则不允许使用 DB_nK_CACHE_SIZE
- 每个高速缓存的最小大小：一个粒组

ORACLE

9-15

Copyright © Oracle Corporation, 2001. All rights reserved.

非标准块大小

数据库缓冲区高速缓存初始化参数决定了 SGA 数据库缓冲区高速缓存组件的大小。可以使用这些参数为数据库使用的各种块大小指定高速缓存大小。如果要在数据库中使用多种块大小，则必须设置 DB_CACHE_SIZE 和至少一个 DB_nK_CACHE_SIZE 参数。每个参数为相应的块大小指定了缓冲区高速缓存大小。DB_nK_CACHE_SIZE 参数的缺省值为零。如果存在块大小为 n KB 的联机表空间，则不要将此参数设置为零。

平台特定的块大小具有一些限制。例如，如果平台上的最大块大小小于 32 KB，则不能设置 DB_32K_CACHE_SIZE。此外，如果最小块大小大于 2 KB，则不能设置 DB_2K_CACHE_SIZE。

注：这些参数不能用于调整标准块大小的高速缓存大小。例如，如果 DB_BLOCK_SIZE 的值为 2 KB，则设置 DB_2K_CACHE_SIZE 是非法的。标准块大小的高速缓存大小始终由 DB_CACHE_SIZE 的值确定。

粒组是一个连续的虚拟内存分配单位。粒组的大小取决于估算的 SGA 的总大小，这个总大小根据 SGA_MAX_SIZE 参数值进行计算。

创建非标准块大小的表空间

分配非标准块大小缓存区

```
Alter system set db_4k_cache_size=1m;
```

```
CREATE TABLESPACE tbs_1  
DATAFILE 'd:\test\tbs_1.dbf'  
SIZE 10M BLOCKSIZE 4K;
```

ORACLE

9-16

Copyright © Oracle Corporation, 2001. All rights reserved.

创建非标准块大小的表空间

使用 BLOCKSIZE 子句为表空间指定非标准块大小。可以用字节或千字节（使用 K 后缀）指定大小。

要指定该子句，必须设置 DB_CACHE_SIZE 和至少一个 DB_nK_CACHE_SIZE 参数，在该子句中指定的整数必须与某个 DB_nK_CACHE_SIZE 参数的设置值对应。

限制：不能为临时表空间（即，如果同时指定了 TEMPORARY，或者要将该空间作为任何用户的临时表空间）指定非标准块大小。

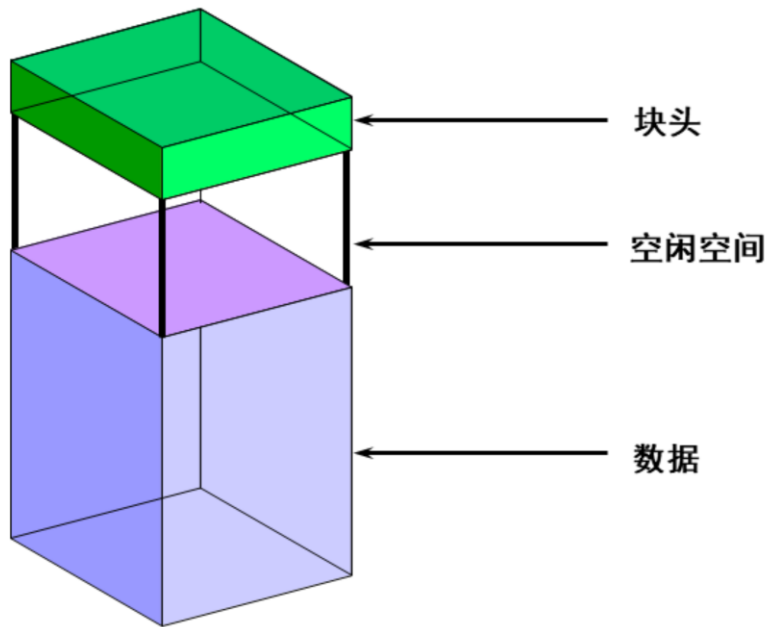
上面的第一条语句创建一个名为 tbs_1 的表空间，使用的数据文件 tbs_1.dbf 具有 4 KB 的块大小。要成功执行这条语句，当前必须在缓冲区高速缓存中配置了 4 KB 大小的缓冲区。

多种块大小的规则

- 分区对象的所有分区必须位于具有相同块大小的表空间中。
- 所有临时、还原表空间必须采用标准块大小。
- 按索引组织的表溢出和外部 **LOB** 段可以存储在块大小与基表不同的表空间中。
- 非标准块大小必须得到操作系统支持

ORACLE

数据库块内容



ORACLE

9-18

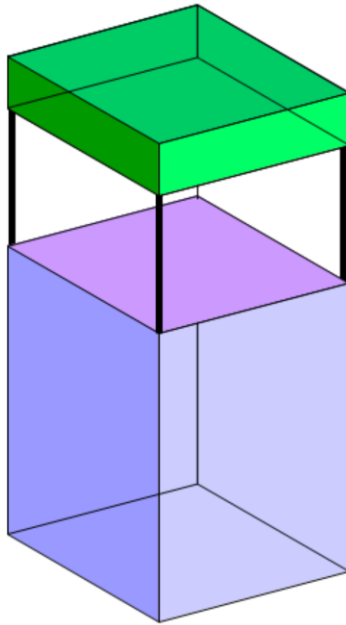
Copyright © Oracle Corporation, 2001. All rights reserved.

数据块

Oracle 数据块包含：

- 块头：块头包含数据块地址、表目录、行目录和事务处理对块中的行进行更改时所用的事务位置。块头从上往下增长。
- 数据空间：行数据从下往上插入到块中。
- 空闲空间：块内空闲空间位于块的中部。因此，必要时块头和行数据空间可以增长。最初，块内的空闲空间是连续的。但是，删除和更新会使块内的空闲空间产生碎片。Oracle 服务器可以在需要时合并块内的空闲空间。

块空间使用参数



INITTRANS

MAXTRANS

PCTFREE

PCTUSED

ORACLE

9-19

Copyright © Oracle Corporation, 2001. All rights reserved.

块空间使用参数

块空间使用参数可用来控制对数据段和索引段空间的使用。

控制并发性的参数：

INITTRANS 和 MAXTRANS：指定初始的和最大的事务位置数，这些事务位置在索引块或者数据块内创建。事务位置用来存储在某一时间点对块进行更改的事务处理的有关信息。一个事务处理只占用一个事务位置，即使它正在更改多行或者多个索引条目。

INITTRANS：保证最低级别的并发性。对于数据段和索引段，INITTRANS 的缺省值分别为 1 和 2，以保证最低级别的并发性。例如，如果 INITTRANS 设为 3，则保证至少有 3 个事务处理可以同时对该块进行更改。如果需要，也可以从块空闲空间内分配更多事务位置，以允许更多的事务处理并发修改块内的行。

MAXTRANS：缺省值为 255，它用于设置可更改数据块或者索引块的并发事务处理数的限制。进行设置后，该值将限制事务位置对空间的使用，从而保证块内有足够的空间供行或者索引数据使用。

块空间使用参数（续）

控制数据空间使用的参数：

PCTFREE：对于数据段而言，此参数用于指定每个数据块中保留空间所占的百分比，保留空间用于因更新块内的行而导致的增长。**PCTFREE** 的缺省值为 10%。

PCTUSED：对于数据段而言，此参数表示 Oracle 服务器试图为表内每个数据块所保持的已用空间的最低百分比。如果一个块的已用空间低于 **PCTUSED**，则将该块放回到空闲列表中。段的空闲列表是容纳将来所插入内容的候选块的列表。缺省情况下，每个段在创建时都有一个空闲列表。通过设置存储子句的 **FREELISTS** 参数，可以创建有更多空闲列表的段。**PCTUSED** 的缺省值为 40%。

PCTFREE 和 **PCTUSED** 都按可用数据空间百分比来计算，可用数据空间是从整个块大小减去块头空间后剩余的块空间。

当空闲列表中的某块在增加记录后，块的可用空间的百分比低于**PCTFREE**参数所指定的百分比时，此块从空闲列表中移走；

当不在空闲列表中的某块在删除记录后，块的已用空间的百分比低于**PCTUSED**参数所指定的百分比时，此块重新加入空闲列表。

注：“管理索引”一课详细讨论了这些参数在索引中的使用。

数据块管理

可以使用两种方法来管理数据块：

- 自动段空间管理
- 手动管理

ORACLE

自动段空间管理

- 一种在数据库段内管理空闲空间的方法。
- 对段内空闲和已用空间的跟踪是使用位图完成的（与使用空闲列表相对）。
- 此方法提供了：
 - 更方便的管理
 - 更高的空间使用率
 - 改进的并发 INSERT 操作性能
- 限制：不能用于包含 **LOB** 的表空间。

ORACLE

9-22

Copyright © Oracle Corporation, 2001. All rights reserved.

自动段空间管理

易于使用

PCTUSED、FREELISTS、FREELIST GROUPS 均是自动管理的。

更高的空间使用率

所有对象都可以更有效地使用空间，尤其是行大小变化很大的对象。

改进的并发处理性能

改进了对并发访问变化的运行时调整。

自动段空间管理

- 位图段包含一个位图，它描述了与段中的可用空间相关的每个块的状态。
- 该映射包含在单独的一组块中，这些块称为“位图块”(BMB)。
- 插入新行时，服务器就会在该映射中搜索具有足够空间的块。
- 当块中的可用空间数量发生变化时，位图中就会反映出它的新状态。

ORACLE

配置自动段空间管理

- 自动段空间管理仅能在表空间级别启用，用于在本地管理的表空间。

```
CREATE TABLESPACE data02  
DATAFILE '/u01/oradata/data02.dbf' SIZE 5M  
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 64K  
SEGMENT SPACE MANAGEMENT AUTO;
```

- 创建表空间后，这些规格将应用于在该表空间中创建的所有段。

ORACLE

9-24

Copyright © Oracle Corporation, 2001. All rights reserved.

配置自动段空间管理

位图段是通过 CREATE TABLESPACE 命令的 SEGMENT SPACE MANGEMENT AUTO 子句指定的，此后不能更改这些段。如果定义了 PCTUSED、FREELIST 和 FREELIST GROUPS，则将其全部忽略。

可以用位图管理的段为：规则表、索引、按索引组织的表 (IOT) 以及 LOB。

手动数据块管理

- 允许使用参数手动配置数据块
 - PCTFREE
 - PCTUSED
 - FREELIST
- 表空间是大文件模式
 - 段空间管理只能是自动

ORACLE

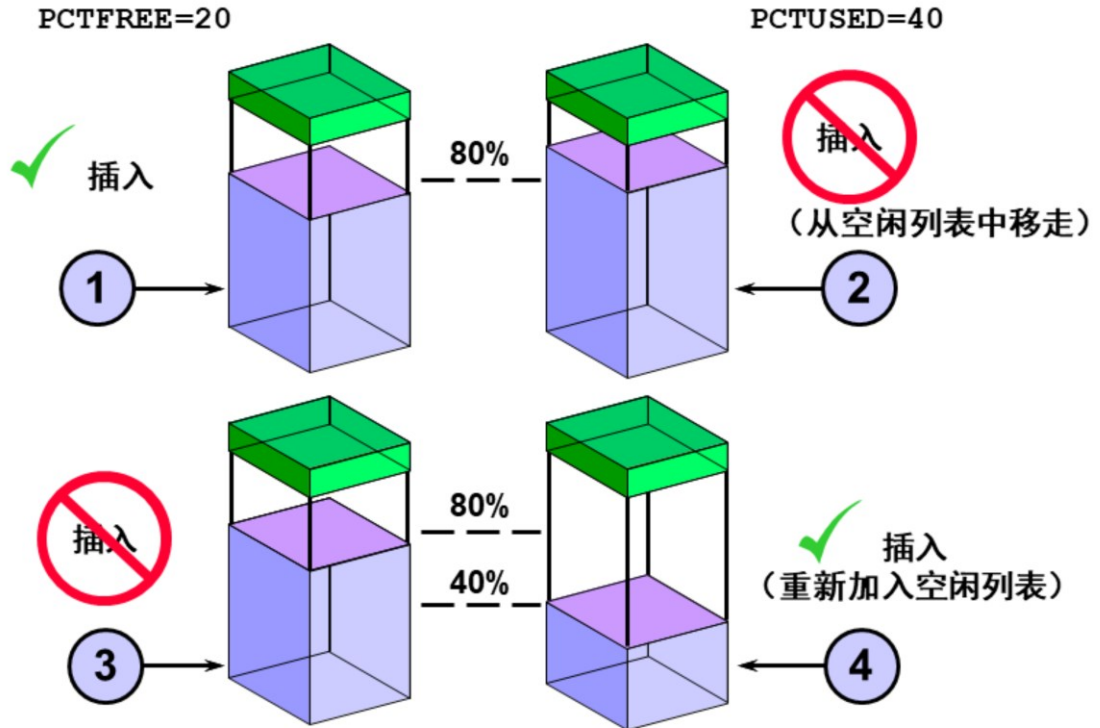
9-25

Copyright © Oracle Corporation, 2001. All rights reserved.

手动数据块管理

使用手动数据块管理，可以配置如何使用块空间以及块何时可用。可以在手动管理中使用 PCTFREE、PCTUSED 和 FREELIST 等参数。在以前的版本中，这是管理数据块的唯一方法。

块空间使用率



9-26

Copyright © Oracle Corporation, 2001. All rights reserved.

ORACLE

块空间使用率

下面步骤介绍对于 $PCTFREE=20$ 且 $PCTUSED=40$ 的数据段如何管理块内的空间：

1. 向块中插入行，直到块内的空闲空间等于或者小于 20%。当行所占用的块内可用数据空间达到 80% ($100 - PCTFREE$) 或者更多后，则无法在该块内插入。
2. 剩余的 20% 可在行大小增长时使用。例如，更新初始为 NULL 的列并分配一个值。这样，更新后的块使用率可能超过 80%。
3. 如果由于更新，删除了块内的行或者行大小减少，块使用率可能跌至 80% 以下。但是，仍然无法向块中插入，直到块使用率跌至 $PCTUSED$ 以下，在本例中 $PCTUSED$ 为 40%。
4. 当块使用率跌至 $PCTUSED$ 以下时，该块可用于插入。随着向块内插入行，块使用率增长，此时又重复从步骤 1 开始的循环。

注：设置 $PCTFREE$ 和 $PCTUSED$ 参数的原则将在关于表和索引的“管理表”和“管理索引”两课中分别进行详细介绍。

获取存储信息

可以通过查询以下视图来获取有关存储的信息：

- **DBA_EXTENTS**
- **DBA_SEGMENTS**
- **DBA_TABLESPACES**
- **DBA_DATA_FILES**
- **DBA_FREE_SPACE**

ORACLE®

9-27

Copyright © Oracle Corporation, 2001. All rights reserved.

查询数据字典

表空间、数据文件、段和空闲区与已用区之间的关系可通过查询数据字典查看。

当创建具有一个或多个文件的表空间时，系统将在 DBA_TABLESPACES 中添加一行。对于数据库中的每个文件，系统都会在 DBA_DATA_FILES 中添加一行。在此阶段，每个数据文件内的空间（除了文件标头）在 DBA_FREE_SPACE 中均显示为一个空闲区。

创建一个段后，在 DBA_SEGMENTS 内将看到一行。可从 DBA_EXTENTS 查看为该段中的区分配的空间，同时调整 DBA_FREE_SPACE 以显示为段创建的区所在文件内的空闲空间减少了。

文件内的所有空间（除了标题块）必须在 DBA_FREE_SPACE 或 DBA_EXTENTS 中说明。

查询数据字典（续）

DBA_SEGMENTS 视图：

查询 DBA_SEGMENTS 视图以获得分配给某个段的区和块的数目。

```
SQL> SELECT segment_name,tablespace_name,extents,blocks
2 FROM dba_segments
3 WHERE owner = 'HR';

SEGMENT_NAME TABLESPACE EXTENTS BLOCKS
-----
REGIONS SAMPLE 1 8
LOCATIONS SAMPLE 1 8
DEPARTMENTS SAMPLE 1 8
JOBS SAMPLE 1 8
EMPLOYEES SAMPLE 1 8
JOB_HISTORY SAMPLE 1 8
5 rows selected.
```

DBA_EXTENTS 视图：

使用 DBA_EXTENTS 视图以检查给定段的区。

```
SQL> SELECT extent_id,file_id,block_id,blocks
2 FROM dba_extents
3 WHERE owner='HR'
4 AND segment_name='EMPLOYEES';

EXTENT_ID FILE_ID BLOCK_ID BLOCKS
-----
0 4 2 5
1 4 27 5
2 4 32 10
3 4 42 15
4 4 57 20
5 rows selected.
```

查询数据字典（续）

DBA_FREE_SPACE 视图:

使用 DBA_FREE_SPACE 视图以检查给定段的区。

```
SQLPLUS> SELECT tablespace_name, count(*),
2> max(blocks), sum(blocks)
3> FROM dba_free_space
4> GROUP BY tablespace_name;
```

TABLESPACE_NAME	COUNT (*)	MAX(BLOCKS)	SUM(BLOCKS)
DATA01	2	1284	1533
RBS	3	2329	2419
SORT	1	1023	1023
SYSTEM	1	5626	5626
TEMP	1	2431	2431

5 rows selected.

小结

在这一课中，您应该能够掌握：

- 使用表空间执行以下操作：
 - 将段分开以便于管理
 - 控制用户的空间分配
- 按段中存储的信息类型对段进行分类
- 使用存储子句确定区大小
- 控制块空间的使用

ORACLE®