

# 大对象技术

单世民



大连理工大学

# 概述

- Oracle Database 10g系统提供了4种大对象 (large object, lob)类型。这4种大对象类型的特点分别如下：
  - ✧ **blob**类型：二进制lob类型，主要用于存储二进制数据。
  - ✧ **clob**类型：字符lob类型，可以用来存储各种字符数据。
  - ✧ **nclob**类型：国际语言字符lob类型，使用多字节存储各种语言的字符，主要用于存储非英文字符。
  - ✧ **bfile**类型：二进制file类型，用于存储指向数据库系统外的文件系统中的文件的指针。这些外部文件可以储存在硬盘上、DVD或CD等介质上。

# 概述

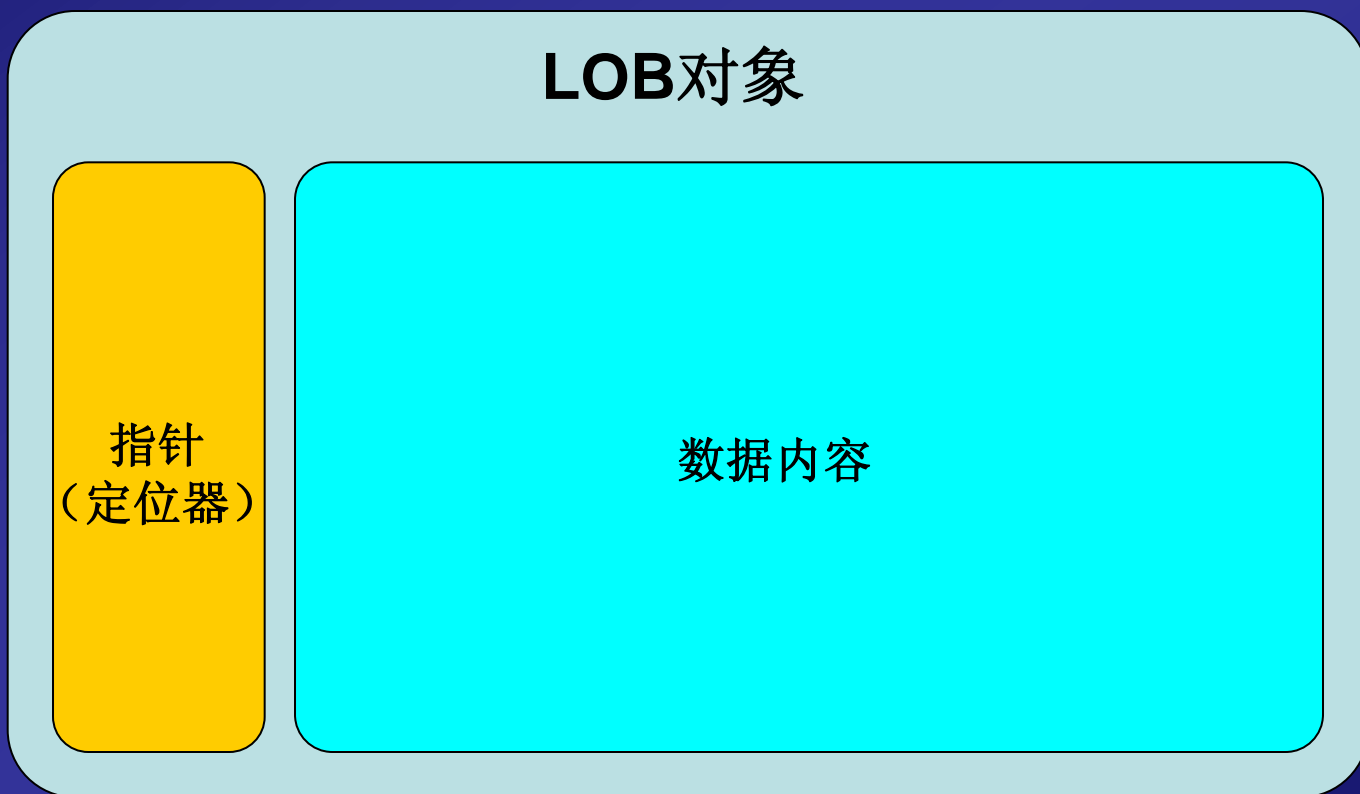
- 在Oracle Database 10g系统中，用户可以使用blob、clob、nclob、bfile等lob数据类型代替long和long raw来存储长数据。虽然Oracle系统允许创建long和long raw列，但是建议使用lob数据类型来创建新列，并且建议将现有的long和long raw列分别转换为clob或blob列。如果使用这些数据类型中的某一种来存储大对象，就可以利用Oracle Database 10g系统提供的新功能来浏览和操作数据，并且可以使用Oracle Text对clob数据执行文本搜索。

# 概述

- 与以前的long和long raw类型相比，lob类型具有以下优点：
  - ✧ 可以存储128TB级的数据，而long类型数据最多只能存储2GB的数据，其中raw类型只能存储4KB的数据
  - ✧ 一个表中可以有多个lob列，但最多只能有一个long或long raw列
  - ✧ 可以按照随机顺序访问lob数据，但是只能顺序访问long和long raw数据

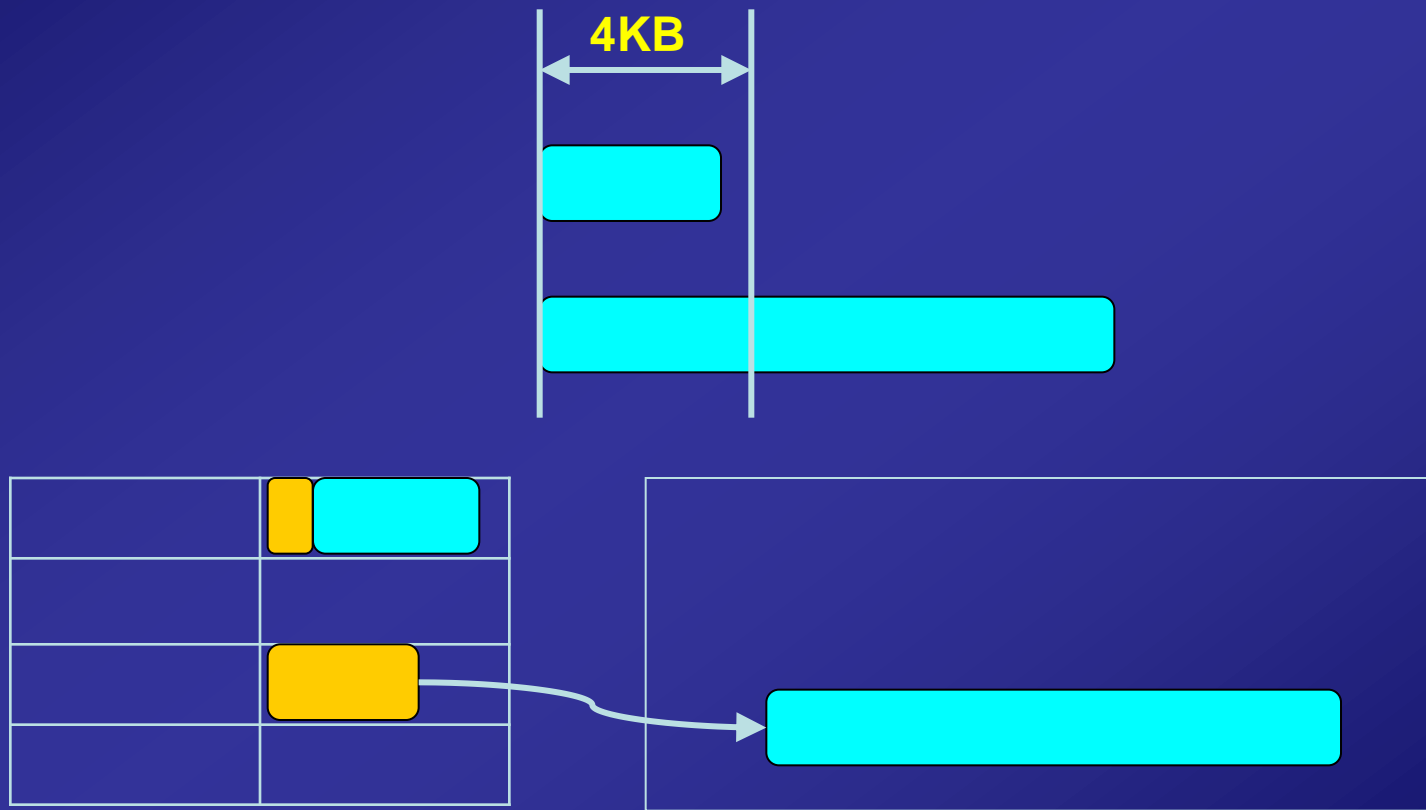
# 概述

- 每一个lob对象都由两部分组成：指针（定位器）和数据内容。



# 概述

- 系统根据lob内容大小的不同，采取不同的存储方式：



# lob数据的存储参数

- 在创建没有lob数据的表时，在create table语句中可能会有一个storage子句。在创建包含lob列的表时，可以为lob数据使用的数据区域指定存储参数，这时，需要在create table语句中再附加一个lob子句。

# lob数据的存储参数

```
CREATE TABLE proposal(  
    proposalID number(10),  
    recipientName varchar2(20),  
    proposalName varchar2(20),  
    description varchar2(200),  
    proposalText clob,  
    budget blob,  
    cover bfile,  
    constraint proposal_pk primary key(proposalID)  
)  
tablespace proposals  
lob(proposalText, budget) store as  
    (tablespace proposals_lobs  
    storage(initial 100K next 100K pctincrease 0)  
    chunk 16k pctversion 10 nocache nologging);
```



# LOB数据的处理

- 在Oracle Database 10g系统中，可以使用多种方法来处理lob值。可以使用原来用于varchar2数据的字符串函数来处理lob数据。对于比较大的lob值，例如大于100KB，或对于那些复杂的数据操作，应该使用dbms\_lob程序包来处理lob数据。除此之外，还可以使用OCI等程序。

# LOB数据的处理

- 对于每一个lob列，Oracle系统将保存一个定位器值，以便找到该记录存储的外部数据。
- 在向包含了lob值的表中插入记录时，可以用函数来告诉Oracle为内部存储的lob列创建一个空定位器。
- 注意，**空定位器值与null值不同**。如果一个内部存储的lob列值为null，那么在更新为非null值之前，必须将其设置为空定位器。——初始化lob值

# LOB数据的处理

- 在Oracle系统中，可以采用以下方式设置Lob数据的空定位器：
  - ✧ blob: empty\_blob()
  - ✧ clob: empty\_clob()
  - ✧ bfile: bfilename()

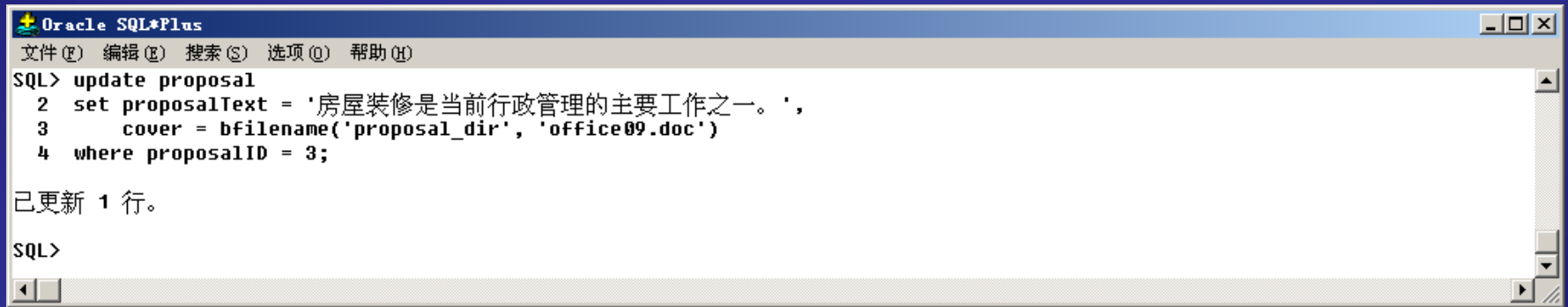


可以使用**bfilename**函数指向一个目录和文件名的组合值，但必须先创建目录对象。

**注意，在插入数据时，bfile数据所指向的文件不一定必须存在**

# 更新LOB数据

- 对于bfile类型的LOB数据，可以采用类似以下示例的方法进行更新。



```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> update proposal
2  set proposalText = '房屋装修是当前行政管理的主要工作之一。',
3    cover = bfilename('proposal_dir', 'office09.doc')
4  where proposalID = 3;

已更新 1 行。

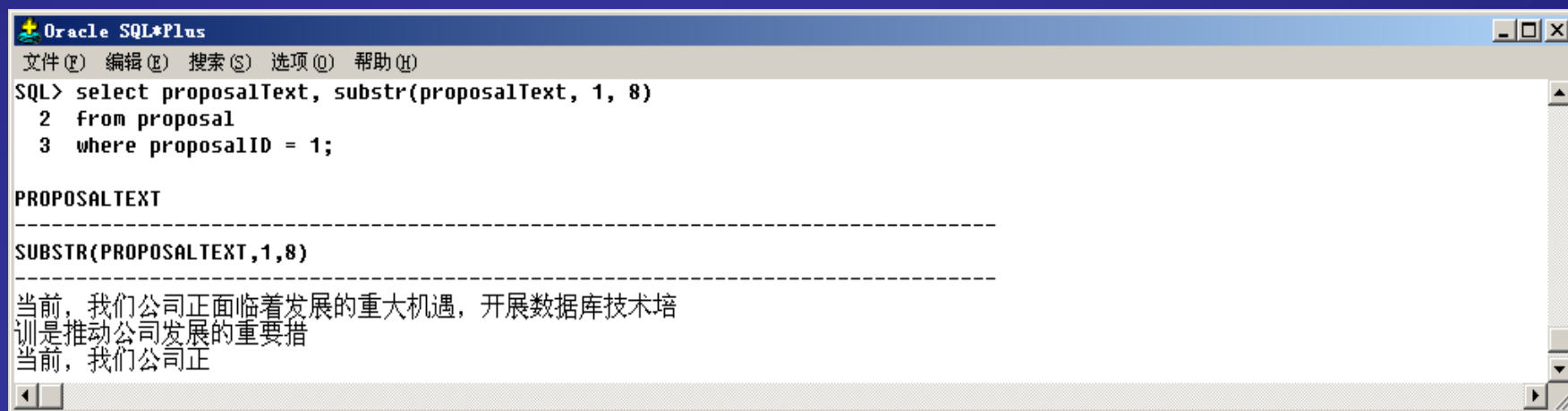
SQL>
```



注意，若要更新lob型数据，必须以将其初始化为空指针为基础

# 使用字符串处理函数处理LOB数据

- 在Oracle系统中，可以使用字符串函数处理clob值。这些字符串函数包括substr、instr、ltrim等函数。
- 例如，可以使用select语句检索proposal表中的proposalText列数据，并且对该列数据使用了substr()函数。从图中可以看到，在使用substr函数检索数据时，一个中文就是一个字符。



```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> select proposalText, substr(proposalText, 1, 8)
2   from proposal
3  where proposalID = 1;

PROPOSALTEXT
-----
SUBSTR(PROPOSALTEXT,1,8)
-----
当前，我们公司正面临着发展的重大机遇，开展数据库技术培
训是推动公司发展的重要措
当前，我们公司正
```

# 处理LOB数据

- 另外，可以使用dbms\_lob程序包对lob数据进行复杂的处理。dbms\_lob程序包中可以使用的方法如教材表18-1所示。
- 以read函数为例：

```
read(lob, amount, offset, buffer)
```

- 小问题：buffer是什么类型的参数？

# 提高大对象的使用效率

- 使用大对象数据有许多好处，方便地处理各种容量的数据。但是，当数据库中的数据太多时，也会给数据库管理员、应用开发人员和终端用户带来许多麻烦。对数据库管理员来说，大对象数据可以带来维护上和管理上的一系列问题。对于应用开发人员和终端用户来说，可以带来查询性能和数据可用性的问题。为了解决这种大对象数据应用带来的问题，Oracle Database 10g系统提供了许多提高使用大对象效率的方法。这些方法主要包括：
  - ❑ 数据分区技术
  - ❑ 压缩数据技术
  - ❑ 并行处理技术
  - ❑ 物化视图技术
  - ❑ RAC技术



# 数据分区技术

- 数据分区就是把业务数据集分解成多个规模比较小的子数据集单元，通过对这些子数据集单元独立地管理和访问，可以提高整个业务数据集的使用效率。
- 例如，北京TOM超市集团有15家分店，每一家分店的销售业务量都很大。如果将这些销售业务数据都放在一个表中，那么可能会引起相当大的性能问题。现在，采取这样的一种策略：创建物理上独立、结构上相同的表，每一个表都包含了每一家分店的销售数据，这些销售数据是集团销售数据的一个子集。如果希望访问整个数据，那么可以采取视图技术将这些数据合并起来。



# 数据分区技术

- 在Oracle Database 10g系统中，可以通过采用分区技术，将大量的业务数据分区成比较小的数据。但是，这种分区技术对于终端用户来说是不透明的。在Oracle Database 10g系统中，从终端用户的角度来看，这里只有一个表。
- 在Oracle Database 10g系统中，可以采取4种不同的方式来分区数据。这4种分区方式分别是范围分区、列表分区、散列分区和组合分区。

# 压缩数据技术

- 如果业务数据的增长速度超过了系统设计的能力，那么会由于数据量过大带来一系列查询性能上的问题。这时，可以使用**Oracle database 10g**系统提供的压缩数据功能来压缩业务数据，从而降低数据量。
- 在压缩数据时，数据库中的冗余数据被删除了，只留下了对这些冗余数据的引用关系，并且这些引用关系都放在数据块的开头位置。通过压缩数据可以大大降低物理磁盘需求的空间。压缩数据最适合用于那些数据很少修改或几乎不修改的环境中。

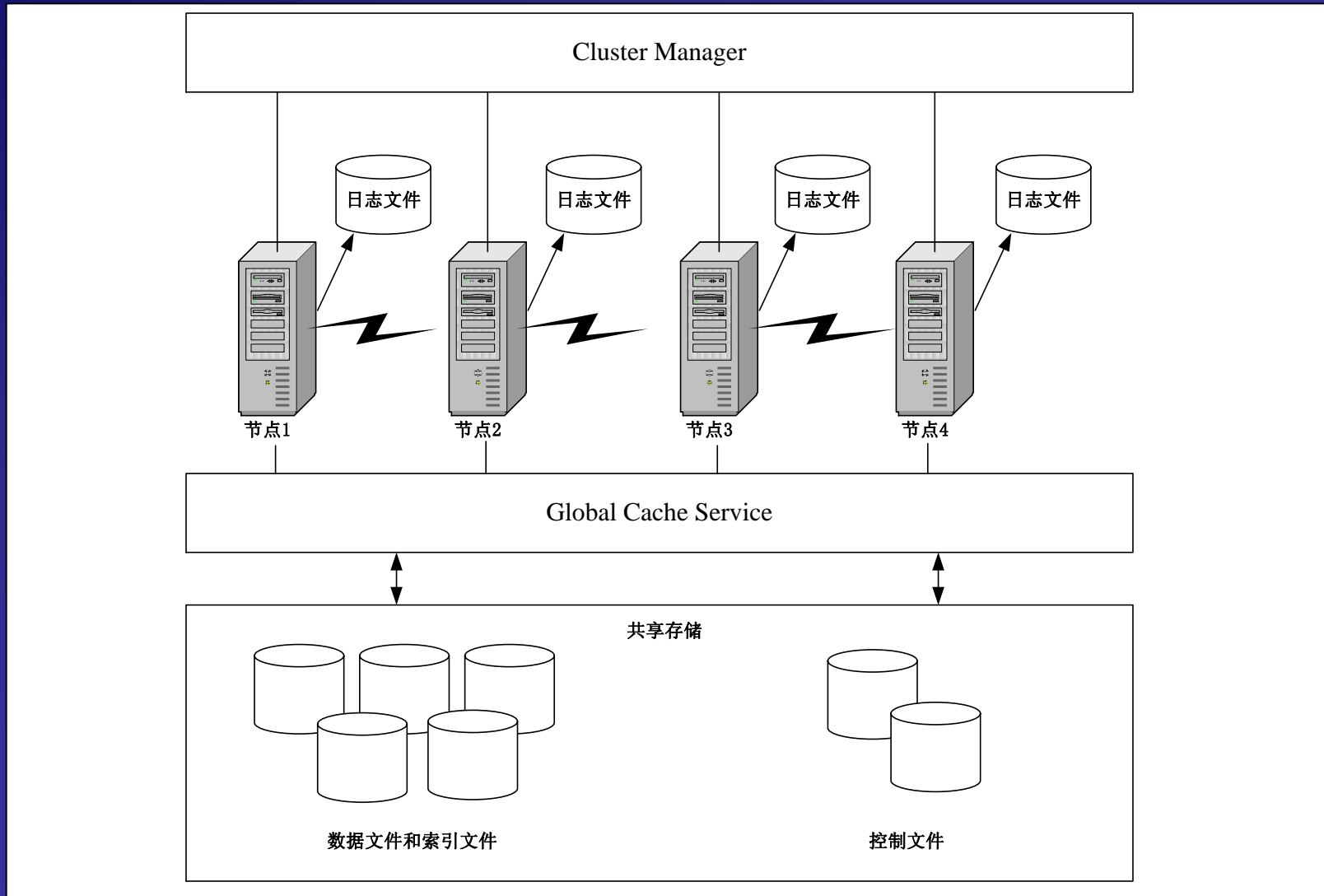
# 并行技术

- 在Oracle Database 10g系统中，还可以通过并行处理的方式来提高查询的性能。并行处理适用于大量数据需要处理的情况，例如，扫描包含大量数据的表、连接查询多个表、创建大表的索引和扫描分区索引等。
- Oracle Database 10g系统提供的并行处理组件包括并行执行协调器和并行执行服务器。并行执行协调器负责把请求分解到多个处理过程中，每一个处理过程都发送到一个并行执行服务器。最后，并行执行协调器把来自每一个服务器的结果组合在一起，把最终完整的结果提交给请求者。

# 物化视图技术

- 物化视图具有视图的特征，但是与视图也不完全相同。可以基于**select**语句创建物化视图，但是物化视图可以物理地保留和存储数据。使用物化视图，可以执行各种查询操作、可以被分区、可以被压缩、可以被并行处理，甚至可以在其上面创建索引。
- 使用物化视图可以增强系统的性能，例如，可以执行数据汇总运算；可以执行表的预连接操作；可以执行耗时且占用大量**CPU**运算时间的计算；以及可以复制和分布数据等。

# RAC技术



# 小结

- 概述
- 存储参数
- 处理方法
- 提高大对象的处理效率

*The End*