分区提供以下优点：

       （1）由于将数据分散到各个分区中，减少了数据损坏的可能性；

       （2）可以对单独的分区进行备份和恢复；

       （3）可以将分区映射到不同的物理磁盘上，来分散IO；

       （4）提高可管理性、可用性和性能。

Oracle 10g提供了以下几种分区类型：

       （1）范围分区（range）；

       （2）哈希分区（hash）；

       （3）列表分区（list）；

       （4）范围－哈希复合分区（range-hash）；

       （5）范围－列表复合分区（range-list）。

Range分区是应用范围比较广的表分区方式，它是以列的值的范围来做为分区的划分条件，将记录存放到列值所在的range分区中。

       如按照时间划分，2010年1月的数据放到a分区，2月的数据放到b分区，在创建的时候，需要指定基于的列，以及分区的范围值。

       在按时间分区时，如果某些记录暂无法预测范围，可以创建maxvalue分区，所有不在指定范围内的记录都会被存储到maxvalue所在分区中。

如：

create table pdba (id number, time date) partition by range (time)

(

partition p1 values less than (to\_date('2010-10-1', 'yyyy-mm-dd')),

partition p2 values less than (to\_date('2010-11-1', 'yyyy-mm-dd')),

partition p3 values less than (to\_date('2010-12-1', 'yyyy-mm-dd')),

partition p4 values less than (maxvalue)

)

Hash分区：

　　对于那些无法有效划分范围的表，可以使用hash分区，这样对于提高性能还是会有一定的帮助。hash分区会将表中的数据平均分配到你指定的几个分区中，列所在分区是依据分区列的hash值自动分配，因此你并不能控制也不知道哪条记录会被放到哪个分区中，hash分区也可以支持多个依赖列。

如：

create table test

(

transaction\_id number primary key,

item\_id number(8) not null

)

partition by hash(transaction\_id)

(

partition part\_01 tablespace tablespace01,

partition part\_02 tablespace tablespace02,

partition part\_03 tablespace tablespace03

);

在这里，我们指定了每个分区的表空间。

List分区：

　　List分区也需要指定列的值，其分区值必须明确指定，该分区列只能有一个，不能像range或者hash分区那样同时指定多个列做为分区依赖列，但它的单个分区对应值可以是多个。

　　在分区时必须确定分区列可能存在的值，一旦插入的列值不在分区范围内，则插入/更新就会失败，因此通常建议使用list分区时，要创建一个default分区存储那些不在指定范围内的记录，类似range分区中的maxvalue分区。

在根据某字段，如城市代码分区时，可以指定default，把非分区规则的数据，全部放到这个default分区。

如：

create table custaddr  
 (

  id                     varchar2(15 byte)   not null,

  areacode   varchar2(4 byte)  
  )

partition by list (areacode)  
 ( partition t\_list025 values ('025'),    
  partition t\_list372 values ('372') ,   
  partition t\_list510 values ('510'),

partition p\_other values (default)

)

组合分区：

       如果某表按照某列分区之后，仍然较大，或者是一些其它的需求，还可以通过分区内再建子分区的方式将分区再分区，即组合分区的方式。

　　组合分区呢在10g中有两种：range-hash，range-list。注意顺序，根分区只能是range分区，子分区可以是hash分区或list分区。

在Oracle 11g中，组合分区功能这块有所增强，又增加了range-range,list-range,

list-list,list-hash，并且 11g里面还支持Interval分区和虚拟列分区。

==============================================================

分区表的索引

       分区索引分为本地(local index)索引和全局索引(global index)。局部索引比全局索引容易管理, 而全局索引比较快。

与索引有关的表：

       dba\_part\_indexes 分区索引的概要统计信息，可以得知每个表上有哪些分区索引，分区索引的类型(local/global)

       dba\_ind\_partitions  每个分区索引的分区级统计信息

       dba\_indexes/dba\_part\_indexes 可以得到每个表上有哪些非分区索引

       Local索引肯定是分区索引，Global索引可以选择是否分区，如果分区，只能是有前缀的分区索引。

注意：

1 对分区操作（truncate，drop，split）都会导致全局索引失效，需要rebuild或者UPDATE GLOBAL INDEXES

2 主键和唯一键索引不能创建为分区索引， 首先根据应用尽量将索引修改为分区索引（个别索引可能不能修改），删除主键也修改为唯一分区索引。

3  主键如果不是你选择的分区字段的话， 在分区表上创建的唯一索引必须包含分区字段，否则会提示错误(ORA-14039)，这一点也需要特别注意。

create table t\_part (id number,name varchar2(22),created date)

partition by range (id)(

partition t\_part\_01 values less than (10),

partition t\_part\_02 values less than (20),

partition t\_part\_03 values less than (30)

);

alter table t\_part add constraint t\_part\_pk primary key (id);

select owner,index\_name,index\_type,table\_owner,table\_name,table\_type,status from dba\_indexes where table\_owner='TEST' and table\_name='T\_PART';

VALID

insert into t\_part values (1,'kkk','03-OCT-16');

insert into t\_part values (10,'ttt','03-OCT-16');

insert into t\_part values (20,'aaa','03-OCT-16');

commit;

======================================================

1

alter table t\_part truncate partition t\_part\_02;

select owner,index\_name,index\_type,table\_owner,table\_name,table\_type,status from dba\_indexes where table\_owner='TEST' and table\_name='T\_PART';

UNUSABLE

alter index t\_part\_pk rebuild online;

2

alter table t\_part add partition t\_part\_04 values less than (40);

select owner,index\_name,index\_type,table\_owner,table\_name,table\_type,status from dba\_indexes where table\_owner='TEST' and table\_name='T\_PART';

VALID

3

alter table t\_part add partition t\_part\_05 values less than (maxvalue);

select owner,object\_name,subobject\_name,object\_type,to\_char(created,'yyyy-mm-dd hh24:mi:ss') as created from dba\_objects where owner='TEST' AND object\_name='T\_PART' order by created desc;

Select table\_name,partition\_name,high\_value,tablespace\_name from user\_tab\_partitions where table\_name='T\_PART' order by partition\_position;

 insert into t\_part values (61,'no valied',null);

 insert into t\_part values (51,'no valied',null);

--拆分分区

-- spilt partition 分区名 at(这里是一个临界区，比如：50000就是说小于50000的放在part\_01，而大于50000的放在part\_02中)

alter table Partition\_HB split Partition part\_02 at (50000) into (Partition part\_01 tablespace dinya\_space01, Partition part\_02 tablespace dinya\_space02);

alter table t\_part split partition t\_part\_max at (60) into (partition t\_part\_05,partition t\_part\_max);

select owner,index\_name,index\_type,table\_owner,table\_name,table\_type,status from dba\_indexes where table\_owner='TEST' and table\_name='T\_PART';

UNUSABLE