

Range	Dates / Numbers / Characters
List	List of values
Hash	No idea
Composite	Combination
Interval	New values
Reference	Child table / Parent Table
System	Application
Virtual	Virtual column

1. Add partition
2. Drop partition
3. Merge partition
4. Split partition
5. Exchange partition
6. Move partition
7. Truncate partition

Range Allows partitioning based on ranges of dates, numbers, or characters.

List useful when the partitions fit nicely into a list of values, like state or region codes.

Hash Allows even distribution of rows when there is no obvious partitioning key.

composite Allows combinations of other partitioning strategies.

interval Extends range partitioning by automatically allocating new partitions when new partition key values exceed the existing high range.

Reference Useful for partitioning a child table based on a parent table column.

Virtual Allows partitioning on a virtual column.

System Allows the application inserting the data to determine which partition

should be used.

Labs

Select name from v\$datafile

Create 3 tablespaces with size 50 m each . (Use the above query for the path)

Hash Partitioning

```
CREATE TABLE emp_hash  
( empno int, ename varchar2(20))  
PARTITION BY HASH (empno)  
( partition part_1 tablespace p1,  
  partition part_2 tablespace p2) ;
```

insert into emp select empno, ename from scott.emp ;

```
select * from emp partition(part_1);  
select * from emp partition(part_2);
```

Range Partitioning

```
create table emp_range  
(EMPNO          NUMBER(4) NOT NULL,  
  ENAME          VARCHAR2(10),  
  JOB            VARCHAR2(9),  
  HIREDATE       DATE,  
  SAL            NUMBER(7,2),  
  DEPTNO         NUMBER(2) NOT NULL,  
  LOC            VARCHAR2(13) NOT NULL)  
partition by range(loc)  
(partition p1 values less than('C') tablespace p1,
```

partition p2 values less than('D') tablespace p2,
partition p3 values less than('N') tablespace p3,
partition p4 values less than('Z') tablespace p4) ;

insert into emp
select e.*, d.loc from scott.emp e, scott.dept d
where e.deptno = d.deptno ;

select 'p1' pname, empno, job, loc from emp partition(p1)
union all
select 'p2' pname, empno, job, loc from emp partition(p2)
union all
select 'p3' pname, empno, job, loc from emp partition(p3)
union all
select 'p4' pname, empno, job, loc from emp partition(p4) ;

select empno, job, loc from emp where empno = 7782;
select empno, job, loc from emp where job = 'CLERK';
select count(*) from emp where job = 'CLERK';

Range Partitioning with date

```
CREATE TABLE range_example  
( range_key_column date ,  
data varchar2(20));  
PARTITION BY RANGE (range_key_column)  
( PARTITION part_1 VALUES LESS THAN  
  (to_date('01/01/2005','dd/mm/yyyy')),  
  PARTITION part_2 VALUES LESS THAN  
  (to_date('01/01/2006','dd/mm/yyyy')) );
```

insert into range_example(range_key_column, data)values

```
( to_date( '15/12/2007 00:00:00', 'dd/mm/yyyy hh24:mi:ss' ),  
  'application data...' );
```

```
select * from range_example partition(part_1);
```

```
update range_example  
  set range_key_column = trunc(range_key_column)  
where range_key_column =  
  to_date( '31-dec-2004 23:59:59', 'dd-mon-yyyy hh24:mi:ss' );
```

```
update range_example  
  set range_key_column = to_date('02-jan-2005','dd-mon-yyyy')  
where range_key_column = to_date('31-dec-2004','dd-mon-yyyy');
```

```
select rowid from range_example  
where range_key_column = to_date('31-dec-2004','dd-mon-yyyy');
```

```
alter table range_example enable row movement;
```

```
update range_example  
  set range_key_column = to_date('02-jan-2005','dd-mon-yyyy')  
where range_key_column = to_date('31-dec-2004','dd-mon-yyyy');
```

List partitioning

```
create table list_example  
( state_cd varchar2(2), data varchar2(20))  
partition by list(state_cd)  
( partition part_1 values ( 'ME', 'NH', 'VT', 'MA' ),  
  partition part_2 values ( 'CT', 'RI', 'NY' ) );
```

```
insert into list_example values ( 'VA', 'data' );
```

```
alter table list_example add partition part_3 values ( DEFAULT );
```

```
insert into list_example values ( 'VA', 'data' );
alter table list_example add partition part_4 values( 'CA', 'NM' );
```

Composite partitioning

```
CREATE TABLE composite_range_list_example
( range_key_column date, code_key_column int,
  data varchar2(20) )
PARTITION BY RANGE (range_key_column)
subpartition by list(code_key_column)
(PARTITION part_1
VALUES LESS THAN(to_date('01/01/2005','dd/mm/yyyy'))
  (subpartition part_1_sub_1 values( 1, 3, 5, 7 ),
   subpartition part_1_sub_2 values( 2, 4, 6, 8 )),
PARTITION part_2
VALUES LESS THAN(to_date('01/01/2006','dd/mm/yyyy'))
  (subpartition part_2_sub_1 values ( 1, 3 ),
   subpartition part_2_sub_2 values ( 5, 7 ),
   subpartition part_2_sub_3 values ( 2, 4, 6, 8 ))) ;
-----
select table_name ,partitioned from user_tables
where table_name='%EMP%';
```

System Partitioning

```
create table apps (app_id number ,app_amnt number)
partition by system
(partition p1 ,partition p2 ,partition p3);
```

Interval Partitioning

```
CREATE TABLE sales (order_date DATE, ...)
PARTITION BY RANGE (order_date)
```

*INTERVAL(NUMTOYMINTERVAL(1,'month'))
(PARTITION p_first VALUES LESS THAN ('01-FEB-2006'));*

*INSERT INTO sales (order_date DATE) VALUES ('04-MAR-2006');
INSERT INTO sales (order_date DATE) VALUES ('17-OCT-2009');*

Virtual Partitioning

*CREATE TABLE employees (
 id NUMBER,
 first_name VARCHAR2(10),
 last_name VARCHAR2(10),
 salary NUMBER(9,2),
 comm1 NUMBER(3),
 comm2 NUMBER(3),
 salary1 AS (ROUND(salary*(1+comm1/100),2)),
 salary2 NUMBER GENERATED ALWAYS AS
 (ROUND(salary*(1+comm2/100),2)) VIRTUAL,
 CONSTRAINT employees_pk PRIMARY KEY (id));*

*INSERT INTO employees (id, first_name, last_name, salary, comm1, comm2)
VALUES (1, 'JOHN', 'DOE', 100, 5, 10);*

*INSERT INTO employees (id, first_name, last_name, salary, comm1, comm2)
VALUES (2, 'JAYNE', 'DOE', 200, 10, 20);*

COMMIT;

Reference Partitioning

*CREATE TABLE Orders
(PONo NUMBER(5),
Custno NUMBER(3),
OrderDate DATE,
ShipDate DATE,
ToStreet VARCHAR2(20),
ToCity VARCHAR2(20),
ToState CHAR(2),
ToZip VARCHAR2(10),
CONSTRAINT Orders_PK PRIMARY KEY (PONo),*

```

CONSTRAINT Orders_FK1 FOREIGN KEY (CustNo) REFERENCES
Customers
)
PARTITION BY RANGE (OrderDate)
(PARTITION olddata VALUES LESS THAN (TO_DATE('01-JAN-2008','DD-
MON-YYYY')),
PARTITION jan2008 VALUES LESS THAN (TO_DATE('01-FEB-2008','DD-
MON-YYYY')),
PARTITION feb2008 VALUES LESS THAN (TO_DATE('01-MAR-2008','DD-
MON-YYYY')),
PARTITION mar2008 VALUES LESS THAN (TO_DATE('01-APR-2008','DD-
MON-YYYY')),
PARTITION apr2008 VALUES LESS THAN (TO_DATE('01-MAY-2008','DD-
MON-YYYY')),
PARTITION may2008 VALUES LESS THAN (TO_DATE('01-JUN-2008','DD-
MON-YYYY')),
PARTITION jun2008 VALUES LESS THAN (TO_DATE('01-JUL-2008','DD-
MON-YYYY')),
PARTITION jul2008 VALUES LESS THAN (TO_DATE('01-AUG-2008','DD-
MON-YYYY')),
PARTITION aug2008 VALUES LESS THAN (TO_DATE('01-SEP-2008','DD-
MON-YYYY')),
PARTITION sep2008 VALUES LESS THAN (TO_DATE('01-OCT-2008','DD-
MON-YYYY')),
PARTITION oct2008 VALUES LESS THAN (TO_DATE('01-NOV-2008','DD-
MON-YYYY')),
PARTITION nov2008 VALUES LESS THAN (TO_DATE('01-DEC-2008','DD-
MON-YYYY')),
PARTITION dec2008 VALUES LESS THAN (TO_DATE('01-JAN-2009','DD-
MON-YYYY'))
);
CREATE TABLE LineItems
(LineNo NUMBER(2),
  POno NUMBER(5) NOT NULL,
  StockNo NUMBER(4),
  Quantity NUMBER(2),
  Discount NUMBER(4,2),
  CONSTRAINT LineItems_PK PRIMARY KEY (LineNo, POno),
  CONSTRAINT LineItems_FK1 FOREIGN KEY (POno) REFERENCES
Orders,

```

```
CONSTRAINT LinItems_FK2 FOREIGN KEY (StockNo) REFERENCES  
StockItems  
)  
PARTITION BY REFERENCE (LinItems_FK1);
```

Indexes

```
CREATE TABLE partitioned_table  
( a int, b int, data char(20))  
PARTITION BY RANGE (a)  
(PARTITION part_1 VALUES LESS THAN(2) tablespace TS1,  
PARTITION part_2 VALUES LESS THAN(3) tablespace TS2) ;
```

```
create index local_prefixed on partitioned_table (a,b) local;  
create index local_nonprefixed on partitioned_table (b) local;
```

```
CREATE TABLE partitioned  
( timestamp date, id int )  
PARTITION BY RANGE (timestamp)  
(PARTITION part_1 VALUES LESS THAN  
( to_date('01-jan-2000','dd-mon-yyyy') ) ,  
PARTITION part_2 VALUES LESS THAN  
( to_date('01-jan-2001','dd-mon-yyyy') ) ) ;
```

```
create index partitioned_idx on partitioned(id) local ;
```

Exchange Partition

```
CREATE TABLE partitioned  
( timestamp date, id int ) )  
PARTITION BY RANGE (timestamp)
```



```
(PARTITION fy_2004 VALUES LESS THAN  
( to_date('01-jan-2005','dd-mon-yyyy') ) ,  
PARTITION fy_2005 VALUES LESS THAN  
( to_date('01-jan-2006','dd-mon-yyyy') )) ;
```

```
insert into partitioned partition(fy_2004)
```

```
select to_date('31-dec-2004','dd-mon-yyyy')-mod(rownum,360),  
object_id from all_objects ;
```

```
insert into partitioned partition(fy_2005)
```

```
select to_date('31-dec-2005','dd-mon-yyyy')-mod(rownum,360),  
object_id from all_objects ;
```

```
create index partitioned_idx_local on partitioned(id) local ;
```

```
create index partitioned_idx_global on partitioned(timestamp) global ;
```

```
create table fy_2004 ( timestamp date, id int );  
create index fy_2004_idx on fy_2004(id) ;
```

```
create table fy_2006 ( timestamp date, id int );
```

```
insert into fy_2006  
select to_date('31-dec-2006','dd-mon-yyyy')-mod(rownum,360),  
object_id from all_objects ;
```

```
create index fy_2006_idx on fy_2006(id) ;
```

```
alter table partitionedexchange partition fy_2004  
with table fy_2004 including indexes without validation ;
```

```
alter table partitioned drop partition fy_2004 ;
```

*alter table partitioned add partition fy_2006
values less than (to_date('01-jan-2007','dd-mon-yyyy')) ;*

*alter table partitioned exchange partition fy_2006
with table fy_2006 including indexes
without validation ;*
