

HASH PARTITION

Q1. Create table Book details with the attribute b_id, title, author, price. Partition this table into 4 partitions using hash partitioning method.

```
SQL> create table bookdetails04(b_id number,b_title varchar(20),author varchar(10),price number)
  2 partition by hash(b_id)
  3 (
  4 partition p1 tablespace users,
  5 partition p2 tablespace users,
  6 partition p3 tablespace users,
  7 partition p4 tablespace users
  8 );
```

Table created.

1. Display the contents of the table.

```
SQL> select * from bookdetails04;
```

B_ID	B_TITLE	AUTHOR	PRICE
2	DS	MCgref	400
5	SPM	Red	790
1	C++	Sid	200
3	JAVA	MAGO	670
4	PHP	Hollow	380

2. Display the contents of each partition

```
SQL> select * from bookdetails04 partition(p1);
```

no rows selected

```
SQL> select * from bookdetails04 partition(p2);
```

no rows selected

```
SQL> select * from bookdetails04 partition(p3);
```

B_ID	B_TITLE	AUTHOR	PRICE
2	DS	MCgref	400
5	SPM	Red	790

```
SQL> select * from bookdetails04 partition(p4);
```

B_ID	B_TITLE	AUTHOR	PRICE
1	C++	Sid	200
3	JAVA	MAGO	670
4	PHP	Hollow	380

3. Rename the partition p1 to part1.

```
SQL> alter table bookdetails04 rename partition p1 to part1;  
Table altered.
```

4. Display the partition names of table book_details.

```
SQL> select partition_name from user_tab_partitions where table_name='BOOKDETAILS04';  
  
PARTITION_NAME  
-----  
P2  
P3  
P4  
PART1
```

Q2. Create a table student_details with the attributes Roll_no, names, marks using hash partitioning with 3 partitions.

```
SQL> create table Studentdetails04(Roll_no number primary key,names varchar(10),marks number)
2  partition by hash(Roll_no)
3  (
4  partition p1 tablespace users,
5  partition p2 tablespace users,
6  partition p3 tablespace users
7  );
Table created.
```

1. Display the content of the partitions.

```
SQL> select * from Studentdetails04 partition(p1);
no rows selected

SQL> select * from Studentdetails04 partition(p2);

  ROLL_NO NAMES      MARKS
-----
      1 SID          99
      3 Light        80
      4 Asced        20

SQL> select * from Studentdetails04 partition(p3);

  ROLL_NO NAMES      MARKS
-----
      2 DARK         40
      5 Bellow       56
```

2. Delete one partition.

```
SQL> alter table Studentdetails04 coalesce partition;
Table altered.
```

3. Display the name of existing partitions.

```
SQL> select partition_name from user_tab_partitions where table_name='STUDENTDETAILS04';

PARTITION_NAME
-----
P1
P2
```

RANGE PARTITION

Q1. Create table student with attributes stud_id, name, marks with range partitioning and the partitioning attribute is marks.

```
SQL> Create table stud_A4
  2  (st_id number primary key,
  3  name varchar(15),
  4  marks number)
  5  Partition by range(marks)
  6  (partition fail_class values less than(45),
  7  partition second_class values less than(59),
  8  partition first_class values less than(75),
  9  partition Dist_class values less than(maxvalue)
 10 );
```

Table created.

1. Display contents of the table.

```
SQL> select * from stud_A4;
```

ST_ID	NAME	MARKS
4	Rupesh	44
1	Rahul	30
3	Tuhar	55
2	Siddaiah	74
5	parshu	100

2. Display the details of the students who failed.

```
SQL> select * from stud_A4 partition(Fail_class);
```

ST_ID	NAME	MARKS
4	Rupesh	44
1	Rahul	30

3. Display the details of the students of "second class".

```
SQL> select * from stud_A4 partition(second_class);
```

ST_ID	NAME	MARKS
3	Tuhar	55

4. Display the details of the students of "First class".

```
SQL> select * from stud_A4 partition(First_class);
```

ST_ID	NAME	MARKS
2	Siddaiah	74

5. Display the name of partitions.

```
SQL> select partition_name from user_tab_partitions where table_name='STUD_A4';
```

PARTITION_NAME
FAIL_CLASS
SECOND_CLASS
FIRST_CLASS
DIST_CLASS

6. Display the details of students who passed with distinctions.

```
SQL> select * from stud_A4 partition(dist_class);
```

ST_ID	NAME	MARKS
5	parshu	100

7. Display the number of students who failed.

```
SQL> select count(marks) from stud_A4 partition(fail_class);
```

COUNT(MARKS)
2

8. Display the details of the student who scored highest marks

```
SQL> select * from stud_A4 where marks=(select max(marks) from stud_A4);
```

ST_ID	NAME	MARKS
5	parshu	100

9. Split the partition fail to f1 with marks less than 30 and f2 to marks less than 45.

```
SQL> alter table stud_A4 split partition fail_class at (30) into (partition f1, partition f2);
Table altered.

SQL> select * from stud_A4 partiton(f1);
select * from stud_A4 partiton(f1)
*
```

ERROR at line 1:
ORA-00933: SQL command not properly ended

```
SQL> select * from stud_A4 partition(f1);
no rows selected

SQL> select * from stud_A4 partition(f2);
```

ST_ID	NAME	MARKS
4	Rupesh	44
1	Rahul	30

10. Merge f1, f2 into a new partition pp1;

```
SQL> alter table stud_A4 merge partitions f1,f2 into partition fail_new;
Table altered.

SQL> select * from stud_A4 partition(fail_new);
```

ST_ID	NAME	MARKS
4	Rupesh	44
1	Rahul	30

11. Drop the partition dist_class.

```
SQL> alter table stud_A4 drop partition dist_class;  
Table altered.
```

12. Add a partition p_new for storing the marks less than 100.

```
SQL> alter table stud_A4 add partition New_class values less than(100);  
Table altered.
```

Q2. Create a table purchase with attributes p_id, p_name and p_amt using range partitioning
create the following six partitions -

P1- amount less than 1000,

P2- amount less than 2000,

P3- amount less than 3000,

P4- amount less than 4000,

P5- amount less than 5000,

P6- amount less than 10000

```
SQL> Create table PurchaseA04(
  2  P_id number primary key,
  3  P_name varchar(20),
  4  P_amt number
  5  )
  6  partition by range(P_amt)
  7  (
  8  partition P1 values less than(1000),
  9  partition P2 values less than(2000),
 10  partition P3 values less than(3000),
 11  partition P4 values less than(4000),
 12  partition P5 values less than(5000),
 13  partition P6 values less than(10000)
 14 );
```

Table created.

1. Display the purchase details having the maximum purchase amount in partition p3.

```
SQL> select max(P_amt) from PurchaseA04 partition(p3)
  2  ;

MAX(P_AMT)
-----
      2200
```


2. Split the partition p1 into pp1 and pp2 with the amount less than 500 and pp2 greater than 500 to pp2.

```
SQL> alter table purchaseA04 split partition p1 at (500) into (partition pp1, partition pp2);
Table altered.
```

```
SQL> select * from PurchaseA04 partition(pp1);
```

P_ID	P_NAME	P_AMT
8	RED	440

```
SQL> select * from PurchaseA04 partition(pp2);
```

P_ID	P_NAME	P_AMT
3	Apple	990

3. Merge the partition pp1 and pp2 into a new partition.

```
SQL> alter table PurchaseA04 merge partitions pp1,pp2 into partition part1;
Table altered.
```

```
SQL> select * from PurchaseA04 partition(part1);
```

P_ID	P_NAME	P_AMT
8	RED	440
3	Apple	990

Q3. Create a table tax details with attributes dept_no, name, tax_amt, state with three partitions p1, p2 and p3 using the partition attribute tax_amt(range partition) partition p1 for tax < 5000, partition p2 for tax < 10000, p3 for tax < 20000.

```
SQL> create table Tax_detail(dept_no number primary key,name varchar(10),tax_amt number,state varchar(30))
2 partition by range(tax_amt)
3 (
4 partition p1 values less than(5000),
5 partition p2 values less than(10000),
6 partition p3 values less than(20000)
7 );
```

Table created.

1. Display the partition wise data.

```
SQL> select * from Tax_detail partition(p1);
```

DEPT_NO	NAME	TAX_AMT	STATE
6	Yellow	4000	Haryana

```
SQL> select * from Tax_detail partition(p2);
```

DEPT_NO	NAME	TAX_AMT	STATE
2	Apples	7000	Kashmir
3	DESCC	8500	Hyderabad

```
SQL> select * from Tax_detail partition(p3);
```

DEPT_NO	NAME	TAX_AMT	STATE
4	Code	12000	Kerala
5	Red	15000	Assam
1	Sid	18000	Maharashtra

2. Display the details if the tax amount is greater than 1000

```
SQL> select * from Tax_detail where tax_amt>1000;
```

DEPT_NO	NAME	TAX_AMT	STATE
6	Yellow	4000	Haryana
2	Apples	7000	Kashmir
3	DESCC	8500	Hyderabad
4	Code	12000	Kerala
5	Red	15000	Assam
1	Sid	18000	Maharashtra

6 rows selected.

3. Display the department having maximum tax amount

```
SQL> select * from Tax_detail where tax_amt=(select max(tax_amt)from tax_detail);
```

DEPT_NO	NAME	TAX_AMT	STATE
1	Sid	18000	Maharashtra

4. Display the state and department having minimum tax amount

```
SQL> select * from Tax_detail where tax_amt=(select min(tax_amt)from tax_detail);
```

DEPT_NO	NAME	TAX_AMT	STATE
6	Yellow	4000	Haryana

5. Drop existing partition p3

```
SQL> alter table Tax_detail drop partition p3;
```

Table altered.

6. Create a new partition p4 to store all the values greater than 10000

```
SQL> alter table Tax_details_A4
  2 Add partition p4 values less Than(MAXVALUE);
```

Table altered.

7. Split the partition p2 to s1 and s2 at 8000

```
SQL> alter table Tax_detail split partition p2 at (8000) into (partition s1, partition s2);
```

Table altered.

8. Merge the partitions p1 and s1 into p11.

```
SQL> alter table Tax_detail merge partitions p1,s1 into partition p11;  
Table altered.
```

9. Rename the partition p11 to p1_new.

```
SQL> alter table Tax_detail rename partition p11 to p1_new;  
Table altered.
```

LIST PARTITIONS

Q1. Create a table to store customer details custid, cname, state with 4 different partitions for 4 different regions north, south, east and west using the list partition.

```
SQL> create table Customer04(custid number primary key, cname varchar(10), state varchar(10))
  2 partition by list(state)
  3 (
  4 partition north values('Delhi','UP','Haryana'),
  5 partition south values('Kerala','Tamilnadu','Andra'),
  6 partition east values('WestBengal','Orissa','Arunachal'),
  7 partition west values('Maharashtr','Gujurat','Goa'));

Table created.
```

1. Display data from all the partitions.

```
SQL> select * from Customer04 partition(north);

  CUSTID CNAME      STATE
-----
      3 Ryuk      Delhi

SQL> select * from Customer04 partition(east);

  CUSTID CNAME      STATE
-----
      4 Red      WestBengal

SQL> select * from Customer04 partition(south);

  CUSTID CNAME      STATE
-----
      1 Siddaiah  Andra
      5 Yellow    Kerala

SQL> select * from Customer04 partition(west);

  CUSTID CNAME      STATE
-----
      2 Light    Maharashtr
```

2. Split the partition south into s1 with Kerala and tamilnadu and s2 with the remaining data.

```
SQL> alter table Customer04 split partition south values('Tamilnadu','Kerala') into (partition s1,partition s2);

Table altered.
```

3. Display the contents of new partition.

```
SQL> select * from Customer04 partition(s1);
```

CUSTID	CNAME	STATE
5	Yellow	Kerala
6	Black	Tamilnadu

```
SQL> select * from Customer04 partition(s2);
```

CUSTID	CNAME	STATE
1	Siddaiah	Andra

4. Merge the partition back.

```
SQL> alter table Customer04 merge partitions s1,s2 into partition south;
Table altered.
```

5. Modify an existing partition east to add Assam and Manipur.

```
SQL> alter table Customer04 modify partition east add values('Assam','Manipur');
Table altered.
```

6. Add new partition Central.

```
SQL> alter table Customer04 add partition central values('Chattisgar','MP');
Table altered.
SQL>
```

7. Truncate the partition west.

```
SQL> alter table Customer04 truncate partition(west);
Table truncated.
```

Abstract Data Type

Q1) Create a table customer with attributes cid, name, address and price.

```
SQL> create table Customer_A4(cid number primary key,name name_type_A4,address address_type_A4,price number);
Table created.
```

Create an abstract data type Name_Type for the attribute name with fname, lname.

```
SQL> create or replace type name_type_A4 as object(fname varchar(10),lname varchar(10));
2 /
Type created.
```

Create an ADT Address_Type for the attribute address with street, city, pincode

```
SQL> create or replace type address_type_A4 as object(street varchar(10),city varchar(10),pincode number);
2 /
Type created.
```

1. Display the first name of all the customers.

```
SQL> select C.name.fname from Customer_A4 C;

NAME.FNAME
-----
Light
pigeon
Red
Ash
```

2. Display the name of all the customers.

```
SQL> select C.name.fname || C.name.lname from Customer_A4 C;

C.NAME.FNAME||C.NAME
-----
LightDark
pigeonwhite
RedYellow
AshKetchum
```

3. Display all the details of customer whose first name starts with 'p'.

```
SQL> select C.name.fname from Customer_A4 C where c.name.fname like 'p%';

NAME.FNAME
-----
pigeon
```

4. Display the details of customers where city is 'Mumbai'

```
SQL> select*from Customer_A4 C where c.address.city like 'Mumbai';
```

```

      CID
-----
NAME(FNAME, LNAME)
-----
ADDRESS(STREET, CITY, PINCODE)
-----
      PRICE
-----
      1
NAME_TYPE_A4('Light', 'Dark')
ADDRESS_TYPE_A4('abc', 'Mumbai', 400095)
      10000

```


Q2. Create a table with following details using Abstract datatype:

name_type

☐ Fname

☐ Lname

```
SQL> create or replace type name_type_A4 as object(fname varchar(10),lname varchar(10));
2 /
Type created.
```

address_type

☐ Street

☐ City

☐ Pin code

```
SQL> create or replace type address_type_A4 as object(street varchar(10),city varchar(10),pincode number);
2 /
Type created.
```

Author_type

☐ Name

☐ Address

```
SQL> create or replace type Author_type as object(name name_type_A4,address address_type_A4);
2 /
Type created.
```

publisher_type

☐ Name

☐ Address

```
SQL> create or replace type publisher_type as object(name name_type_A4,address address_type_A4);
2 /
Type created.
```

Create the table BOOK with following attributes

- ☐ Book id
- ☐ Book title
- ☐ Price
- ☐ Author
- ☐ Publisher

```
SQL> create table Book_A4(Book_id number primary key, Book_title varchar
(10), Price number, Author Author_type, Publisher publisher_type);

Table created.
```

1. Display all the books published by “TMH” .

```
SQL> select b.Book_id,b.Book_title from Book_A4 b where b.publisher.name.fname = 'TMH';

   BOOK_ID BOOK_TITLE
-----
          1 Nature
          2 Peace
```

2. Display the first name of all publishers.

```
SQL> select b.publisher.name.fname from Book_A4 b;

PUBLISHER.
-----
TMH
TMH
Arihant
Shonen
```

3. Display first name of all authors.

```
SQL> select b.author.name.fname from Book_A4 b;

AUTHOR.NAM
-----
Rahul
Rahul
Max
uzumaki
```

4. Display all books details written by author with fname 'Rahul'

```
SQL> select b.Book_id,b.Book_title from Book_A4 b where b.author.name.fname='Rahul';
```

BOOK_ID	BOOK_TITLE
1	Nature
2	Peace

5. Display all the information from BOOK table where price in between 250 and 400 where the Author is from 'Mumbai' and 'Delhi'.

```
SQL> select b.Book_id,b.Book_title from Book_A4 b where b.price between 250 and 450 and b.author.address.city='Mumbai' or b.author.address.city='Delhi';
```

BOOK_ID	BOOK_TITLE
2	Peace
3	Village

6. Display the number of books published by each author.

```
SQL> select b.author.name.fname, count(*) from Book_A4 b
2 Group by b.author.name.fname;
```

AUTHOR.NAM	COUNT(*)
Rahul	2
Max	1
uzumaki	1

7. Display the name of author who wrote only one book.

```
SQL> select b.author.name.fname from Book_A4 b
2 Group by b.author.name.fname
3 Having count(*)=1;
```

AUTHOR.NAM
Max
uzumaki

Q3) Create the following Employee table with eno, ename, hiredate, salary, comm and dept.
Create

the abstract data type ename_type, hire_type and dept_type.

Employee

Eno no_type

Ename name_type

Hiredate hire_type

salary

comm

dept dept_type

no_type

id

designation

name_type

name

address

hire_type

day

month

year

dept_type

dno

dname

loc

```
SQL> create or replace type no_type as object(id number, designation varchar(10));
2 /
Type created.

SQL> create or replace type name_type as object(name varchar(10), address varchar(10));
2 /
Type created.

SQL> create or replace type hire_type as object(day varchar(10), month varchar(10), year number);
2 /
Type created.

SQL> create or replace type dept_type as object(dno number, dname varchar(10), loc varchar(10));
2 /
Type created.
```

```
SQL> create table Employee_A4(Eno no_type,ENAME name_type,Hiredate hire_type,salary number,comm number,dept dept_type);
Table created.
```

1. Display all employees with designation as Analyst and hired after June 2020.

```
SQL> SELECT e.eno.id AS emp_id,
2      e.ename.fname AS first_name,
3      e.hiredate.day AS hire_day,
4      e.hiredate.month AS hire_month,
5      e.hiredate.year AS hire_year,
6      e.salary
7 FROM Employee_A25 e
8 WHERE e.eno.designation = 'Analyst'
9      AND (e.hiredate.year > 2020 OR
10      (e.hiredate.year = 2020 AND e.hiredate.month > 6));
```

EMP_ID	FIRST_NAME	HIRE_DAY	HIRE_MONTH	HIRE_YEAR	SALARY
1	Rahul Sharma	15	7	2021	60000

2. Display all employees working for sales department situated in Navi Mumbai

```
SQL> SELECT e.eno.id AS emp_id,
2      e.ename.fname AS first_name,
3      e.ename.lname AS last_name,
4      e.dept.dname AS dept_name,
5      e.dept.loc AS location,
6      e.salary
7 FROM Employee_A25 e
8 WHERE e.dept.dname = 'Sales'
9      AND e.dept.loc = 'Navi Mumbai';
```

EMP_ID	FIRST_NAME	LAST_NAME	DEPT_NAME	LOCATION	SALARY
1	Rahul Sharma	Andheri, Mumbai	Sales	Navi Mumbai	60000

INHERITANCE

Q1. Create a type person_type with attributes person_id, p_name, p_address. Create a type student under person_type with the attributes dept_name and major subjects. Create a type emp_type under person_type with attributes emp_id and manager_name. Create a type part_time_student_type under student with attributes no. of hours. Create a table person as object table of person_type.

```
SQL> create type person_type_A4 as object(
  2  person_id number,
  3  p_name varchar(10),
  4  p_address varchar(10))not final;
  5  /

Type created.

SQL> create type student_type_A4 under person_type_A4(
  2  dept_name varchar(10),
  3  marjor_subjects varchar2(10))not final;
  4  /

Type created.

SQL> create type emp_type_A4 under person_type_A4(
  2  emp_id number,
  3  manager_name varchar(10))not final;
  4  /

Type created.
```

```
SQL> create type part_time_student_type under student_type_A4(
  2  no_of_hours number)not final;
  3  /

Type created.

SQL> create table person_A4 of person_type_A4;

Table created.

SQL>
```

1. Display all the details of the table.

```
SQL> select * from person_A4;
```

PERSON_ID	P_NAME	P_ADDRESS
1	Siddaiah	Chembur
2	Red	Thane
3	Blue	Bandra
4	Yellow	Dadar
5	Orange	CST
6	Grey	Vashi

```
6 rows selected.
```

2. Display the details of the students.

```
SQL> select * from person_A4 p where value(p) is of (student_type_A4);
```

PERSON_ID	P_NAME	P_ADDRESS
1	Siddaiah	Chembur
3	Blue	Bandra
4	Yellow	Dadar
6	Grey	Vashi

3. Display all the major subjects of student.

```
SQL> select treat(value(p) as student_type_A4).marjor_subjects as marjor_subjects
2   from person_A4 p
3   where value(p) is of (student_type_A4);
```

MARJOR_SUB
Ai,Ml
Eng,Maths
Eng,Maths
C++,Python

4. Display the name of the manager of employee with p_id=2.

```
SQL> select treat(value(p) as emp_type_A4).manager_name as manager_name
2   from person_A4 p
3   where p.person_id = 2;
```

MANAGER_NA
ABC