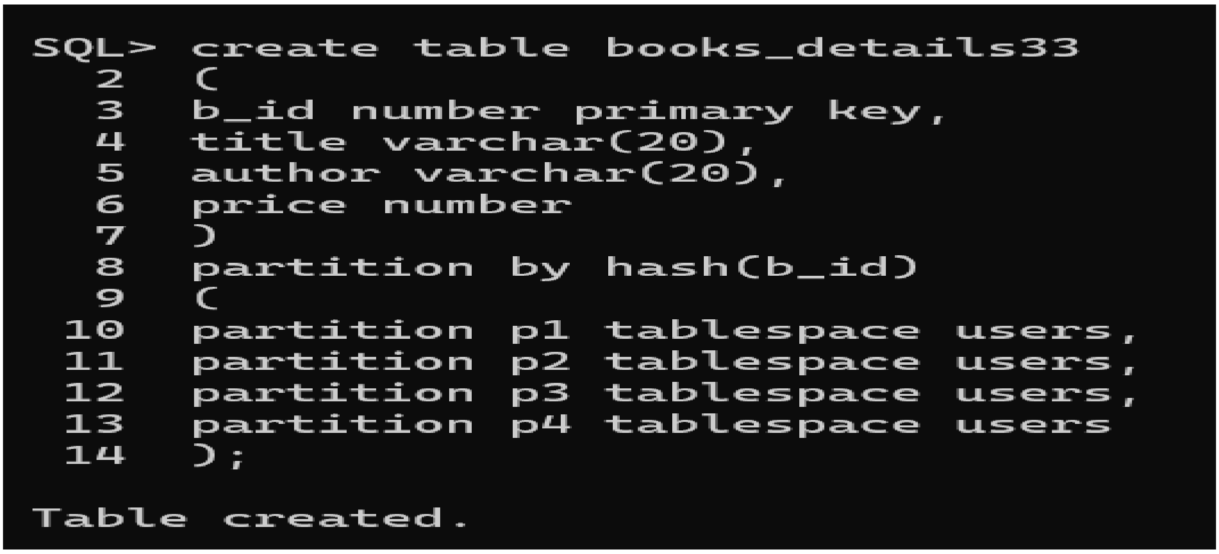
## Hash Partition

**Q.1 create table book_details with attributes b_id,title, author, price. Partition this table into 4 partitions using hash partitioning method.**



```
SQL> create table books_details33
     (
     b_id number primary key,
     title varchar(20),
     author varchar(20),
     price number
     )
     partition by hash(b_id)
     (
     partition p1 tablespace users,
     partition p2 tablespace users,
     partition p3 tablespace users,
     partition p4 tablespace users
     );

Table created.
```

1) **Display the contents of the table.**

select * from books_details33;

2) **Display the contents of each partition.**

select * from books_details33 partition(p1);

select * from books_details33 partition(p2);

select * from books_details33 partition(p3);

select * from books_details33 partition(p4);

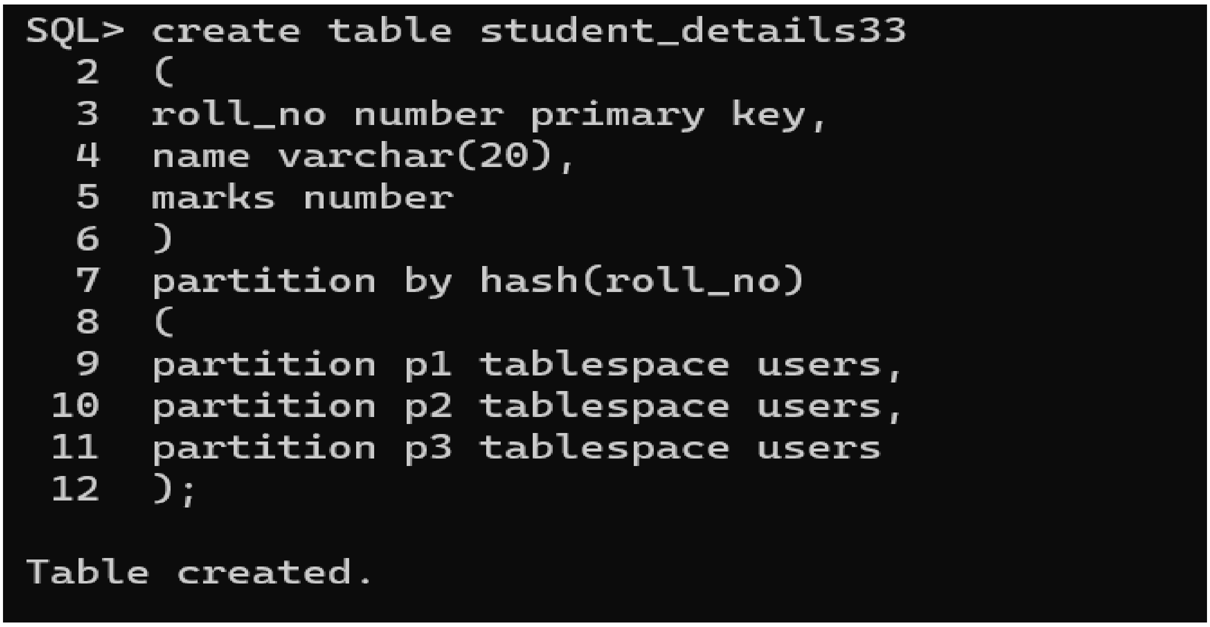
3) **Rename the partition p1 to part1**

alter table books_details33 rename partition p1 to part1;

4) **Display the partition names of the table books_details**

select partition_name from user_tab_partitions where table_name = 'BOOKS_DETAILS33';

**Q.2 create a table student_details with the attributes roll_no, names, marks using hash partitioning with 3 partitions**



```
SQL> create table student_details33
  2  (
  3  roll_no number primary key,
  4  name varchar(20),
  5  marks number
  6  )
  7  partition by hash(roll_no)
  8  (
  9  partition p1 tablespace users,
 10  partition p2 tablespace users,
 11  partition p3 tablespace users
 12  );

Table created.
```

1) **Display the contents of table.**

Select * from student_details33;

2) **Display the contents of the partitions.**

select * from student_details33 partition(p1);

select * from student_details33 partition(p2);

select * from student_details33 partition(p3);

3) **Delete on partition**

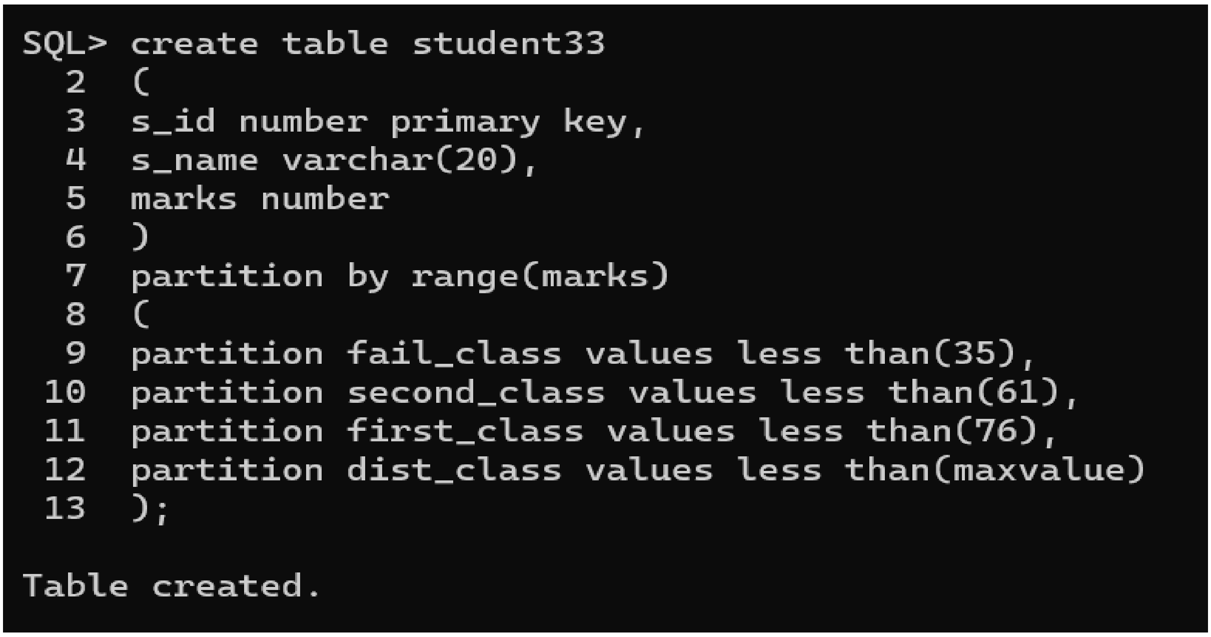
alter table student_details33 coalesce partition;

4) **Display the names of existing partitions**

select partition_name from user_tab_partitions where table_name = 'STUDENT_DETAILS33';

## Range Partition

Q.1 Create table student with attributes stud_id, name, marks with range partitioning and the partitioning attribute is marks.



```
SQL> create table student33
  2  (
  3  s_id number primary key,
  4  s_name varchar(20),
  5  marks number
  6  )
  7  partition by range(marks)
  8  (
  9  partition fail_class values less than(35),
 10  partition second_class values less than(61),
 11  partition first_class values less than(76),
 12  partition dist_class values less than(maxvalue)
 13  );

Table created.
```

1) **Display the content of the table.**

select * from student33;

2) **Display the details of students of failed class.**

select * from student33 partition(fail_class);

3) **Display the details of students of second class.**

select * from student33 partition(second_class);

4) **Display the details of students of first class.**

select * from student33 partition(first_class);

5) **Display the names of partitions.**

select partition_name from user_tab_partitions where table_name = 'STUDENT33';

**6) Display the details of students who passed with distinction.**

```
select * from student33 partition(dist_class);
```

**7) Display the number of students who failed.**

```
select count(s_id) from student33 partition(fail_class);
```

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

```
select * from student33 where marks = (select max(marks) from student33);
```

**9) Split the partition fail to f1 with marks less than 30 and f2 to marks less than 35.**

```
alter table student33 split partition fail_class at(30) into (partition faill, partition fail2);
```

**10) Merge fail1 and fail2 into new partition pp1**

```
alter table student33 merge partitions faill, fail2 into partition pp1;
```

```
select * from student33 partition(pp1);
```

**11) Drop the partition dist_class.**

```
select partition_name from user_tab_partitions where table_name = 'STUDENT33';
```

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

```
alter table student33 add partition p_new values less than(100);
```

**Q.2 Create a table purchase with attributes p_id,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

```
create table purchase33 (
   p_id number primary key,
   p_name varchar(20),
   p_amt number
)
partition by range(p_amt)
(
   partition p1 values less than(1000),
   partition p2 values less than(2000),
   partition p3 values less than(3000),
   partition p4 values less than(4000),
   partition p5 values less than(5000),
   partition p6 values less than(10000)
);
```

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

select * from purchase33 where p_amt=(select max(p_amt) from purchase33 partition(p3));

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

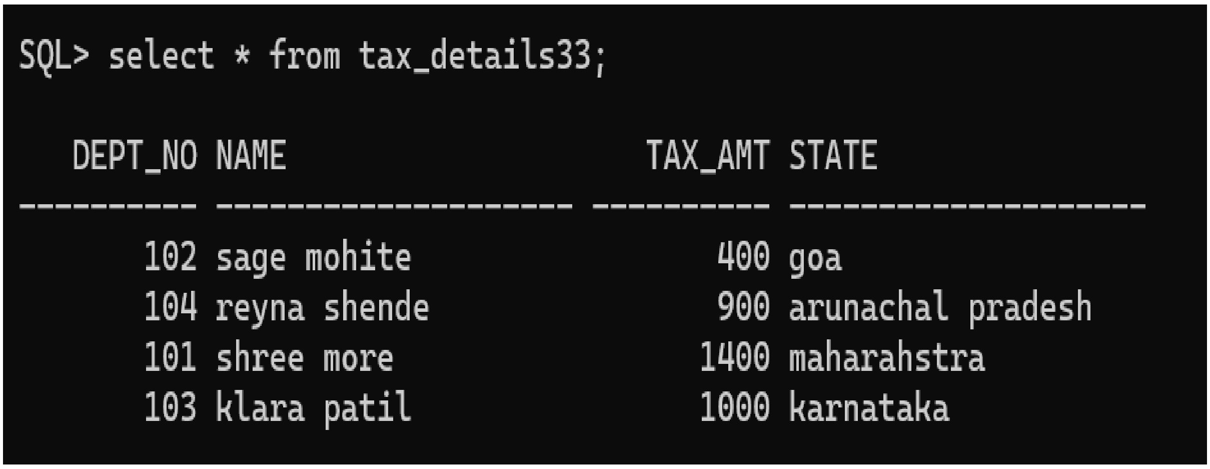
alter table purchase33 split partition p1 at(500) into (partition pp1, partition pp2);

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

alter table purchase33 merge partitions pp1, pp2 into partition newpp;

Q.3 Create a table tax_details with the attributes dept_no , name , tax_amt, state with three partitions p1,p2,p3 using the partition attribute tax_amt(range partition).

```
create table tax_details33 (
   dept_no number primary key,
   name varchar(20),
   tax_amt number,
   state varchar(20)
)
partition by range(tax_amt)
(
   partition p1 values less than(500),
   partition p2 values less than(1000),
   partition p3 values less than(1500)
);
```



```
SQL> select * from tax_details33;

   DEPT_NO NAME                    TAX_AMT STATE
---------- -------------------- ---------- --------------------
       102 sage mohite                 400 goa
       104 reyna shende                900 arunachal pradesh
       101 shree more                 1400 maharahstra
       103 klara patil                1000 karnataka
```

## List Partition

Q.1 Create a table to store customer details cust_id, c_name, state with 4 different partitions for different regions north, south, east and west using the list partition.

```
create table customer33 (
   cust_id number primary key,
   c_name varchar(20),
   state varchar(20)
)
partition by list(state)
(
   partition north values('Punjab', 'Himachal Pradesh', 'Uttarakhand'),
   partition east values('Nagaland', 'Tripura', 'Arunachal Pradesh', 'Meghalaya'),
   partition west values('Rajasthan', 'Gujarat', 'Maharashtra', 'Madhya Pradesh'),
   partition south values('Kerala', 'Karnataka', 'Tamil Nadu', 'Andhra Pradesh')
);
```

1) **Display data from all partitions**

```
selcect * from customer33;
```

2) **Split the partition south into s1 with kerala and Tamil Nadu and s2 with the remaining data.**

```
ALTER TABLE customer33 SPLIT PARTITION south INTO (PARTITION s1 VALUES ('Kerala', 'Tamil Nadu'), PARTITION s2 VALUES ('Karnataka', 'Andhra Pradesh'));
```

3) **Display the contents of new partitions.**

```
ALTER TABLE customer33 SPLIT PARTITION south INTO (PARTITION s1 VALUES ('Kerala', 'Tamil Nadu'), PARTITION s2);
```

4) **Merge the partition back**

```
ALTER TABLE customer33 MERGE PARTITIONS s1, s2 INTO PARTITION new_south;
```

```
select partition_name from user_tab_partitions where table_name =
'CUSTOMER33';
```

**5) Modify an existing partition east to add assam and manipur.**

```
alter table customer33 modify partition east add values('Assam', 'Manipur');
```

**6) Add new partition central.**

```
alter table customer33 add partition central values('Chhatisgarh');
```

```
select partition_name from user_tab_partitions where table_name =
'CUSTOMER33';
```

**7) Truncate the partition west.**

```
alter table customer33 truncate partition west;
```

# Abstract Data Type

Q.1

1. **Create a table customer with attributes c_id , c_name, address and price.**

```
create table customer33 (
   c_id number,
   name name_type33,
   address address_type33,
   price number
);
```

2. **Create an abstract data type name_type with attribute names fname and lname.**

```
create type name_type33 as object (
   fname varchar(10),
   lname varchar(10)
);
```

3. **Create an abstract data type address_type with attributes street, city and pincode.**

```
create type address_type33 as object (
   street varchar(10),
   city varchar(10),
   pincode number
);
```

4. **Display the first name of all the customers.**

```
select c.name.fname from customer33 c;
```

5. **Display the name of all the customers.**

```
select c.name.fname || ' ' || c.name.lname from customer33 c;
```

6. **Display all the details of customers whose first name starts with ‘P’.**

```
select c.c_id, c.name.fname, c.name.lname, c.address.street, c.address.city,
c.address.pincode, price from customer33 c where c.name.fname like 'P%';
```

7. **Disply th details of customer where city is ‘Mumbai’.**

```
select c.c_id, c.name.fname, c.name.lname, c.address.street, c.address.city,
c.address.pincode, price from customer33 c where c.address.city = 'Mumbai';
```

**Q.2 Create a table with following details using abstract datatype.**

a) **Name_type**
   - **Fname**
   - **lname**

```
create type name_type33 as object (
   fname varchar(10),
   lname varchar(10)
);
```

b) **Address_type**
   - **Street**
   - **City**
   - **Pincode**

```
create type address_type33 as object (
   street varchar(10),
   city varchar(10),
   pincode number
);
```

c) **Author_type**
   - **Name**
   - **Address**

```
create type author_type33 as object (
   name name_type33,
   address address_type33
);
```

d) **Publisher_type**
   - **Name**
   - **Address**

```
create type publisher_type33 as object (
   name name_type33,
   address address_type33
);
```

**e) Create a table books with following attributes.**

- **Book_id**
- **Book_title**
- **Price**
- **Author**
- **Publisher**

```
create table books33 (
   book_id number,
   book_title varchar(20),
   price number,
   author author_type33,
   publisher publisher_type33
);
```

```
insert into books33 values(1, 'core java', 399,
author_type33(name_type33('Rahul', 'Nemani'), address_type33('wadala',
'Mumbai', 400601)), publisher_type33(name_type33('TataMg', 'Hill'),
address_type33('belapur', 'NewMumbai', 400708)));
```

```
insert into books33 values(2, 'c++', 260, author_type33(name_type33('Shrijay',
'more'), address_type33('chaubeRoad', 'Delhi', 300643)),
publisher_type33(name_type33('SPD', 'books'), address_type33('MG road',
'Delhi', 300643)));
```

1) **Display all the books published by 'TataMg'.**

select b.book_title from books33 b where b.publisher.name.fname = 'TataMg';

2) **Display the first name of all publishers.**

select b.publisher.name.fname from books33 b;

3) **Display the first name of all authors.**

select b.author.name.fname from books33 b;

4) **Display all book details written by author with fname Rahul.**

select b.book_title from books33 b where b.author.name.fname = 'Rahul';

5) **Display all the information from books table where price is between 250 and 400 where author is from Mumbai and Delhi.**

select * from books33 b where b.price between 250 and 400 and (b.author.address.city = 'Mumbai' or b.author.address.city = 'Delhi');

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

select b.author.name.fname, count(book_title) from books33 b group by b.author.name.fname;

7) **Display the author who wrote only one book**

select count(book_title) as num_of_books, b.author.name.fname as authors from books33 b group by b.author.name.fname having count(book_title) = 1;

**Q.3 Create the following employee table with e_no ,e_name , hiredate ,salary, comm, dept. Create the abstract data type eno_type, ename_type, hire_type and dept_type.**

**Employee :**

```
create table employee_a33 (
   eno eno_type33,
   ename name_type33,
   hiredate hire_type33,
   salary number(10),
   comm number(10),
   dept dept_type33
);
```

**Ename_type :**

```
create type ename_type33 as object (
   name varchar(20),
   address varchar(20)
);
```

**Eno_type :**

```
create type eno_type33 as object (
   id number,
   designation varchar(20)
);
```

**Hire_type :**

```
create type hire_type33 as object (
   day number,
   month varchar(20),
   year number(20)
);
```

**Dept_type :**

```
create type dept_type33 as object (
   d_no number,
   d_name varchar(20),
   loc varchar(20)
);
```

1) **Display all the employees with designation as java dev and hired after march 2023.**

```
select e.eno.id, e.eno.designation from employee_a33 e where e.eno.designation = 'Java Dev' and e.hiredate.year = 2023;
```

2) **Display all the employees working for IT department situated in belapur.**

```
select e.eno.id, e.dept.d_name from employee_a33 e where e.dept.d_name = 'IT' and e.dept.loc = 'Belapur';
```

## Inheritance

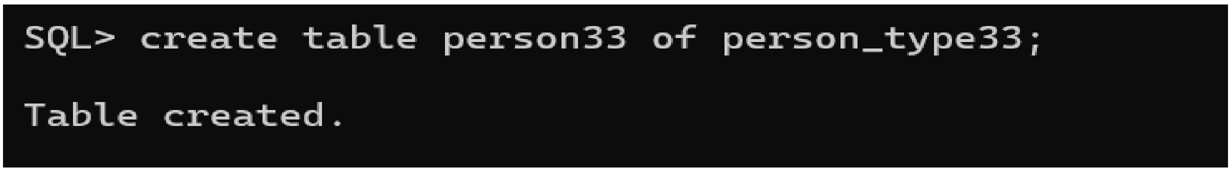
**Q.1 Create a type person_type with attributes person_id, p_name, p_address. Create a type student under person_type with the attributes dept_name and major subjects. Create a type emp_type under person_type with attributes emp_id and manager_name. Create a type part_time_student _type under student with attributes no. of hours. Create a table person as object table of person_type**.

```
create type person_type33 as object
(
   p_id number,
   p_name varchar(20),
   p_address varchar(20)
) not final;


create type student_type33 under person_type33
(
   dept_name varchar(20),
   major_subject varchar(20)
) not final;


create type emp_type33 under person_type33
(
   emp_id number,
   manager_name varchar(20)
);
create type part_time_student33 under student_type33
(
   no_of_hrs number
```

);



```
SQL> create table person33 of person_type33;

Table created.
```

1) **Display all the details of the table.**

select * from person33;

2) **Display details of the students.**

select * from person33 e where value(e) IS OF (student_type33);

3) **Display all major subjects of students.**

select TREAT(VALUE(E) AS student_type33).major_subject from person33 e where value(e) IS OF (student_type33);

4) **Display the name of the manager of employee with p_id = 3.**

select TREAT(VALUE(E) AS emp_type33).manager_name from person33 e where e.p_id = 3 and value(e) IS OF (emp_type33);