Practical 10

- ## Implementation of Data partitioning through Range and List partitioning

Range Partitioning:

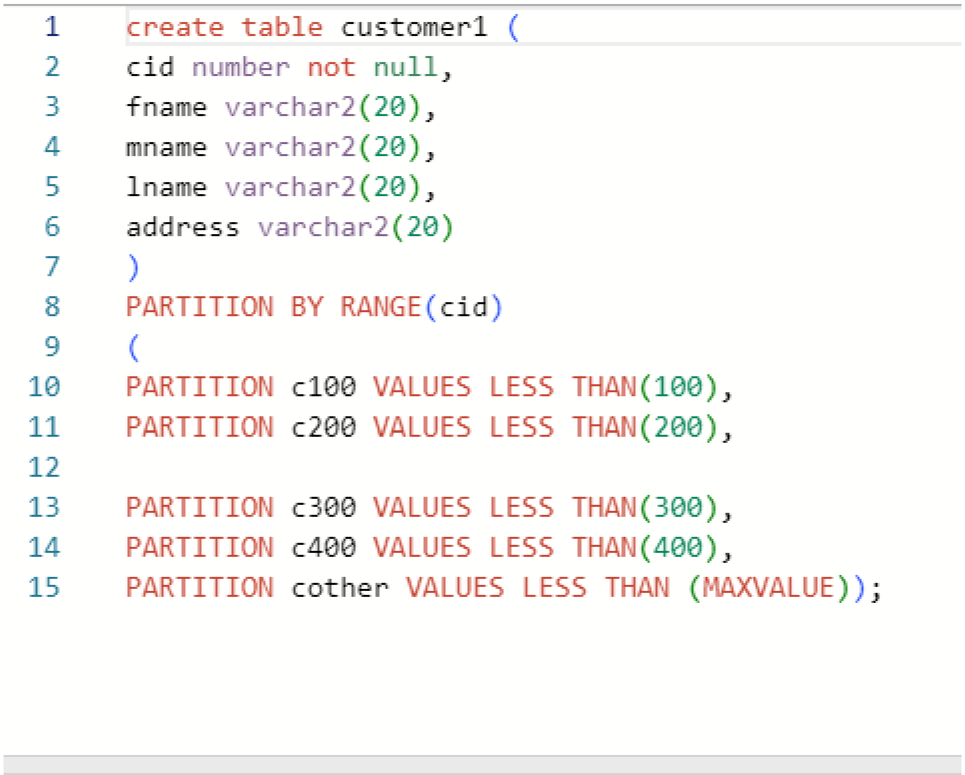
1. Create a table Customer with following schema &amp; partition the cid column by using range

Partitioning:

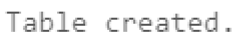
cid number not null
fname varchar2(20)
mname varchar2(20)
lname varchar2(20)
address varchar2(20)

1. Make 3 partitions which will contain the values from 1 to 99, 100 to 199, and 200

Onwards.



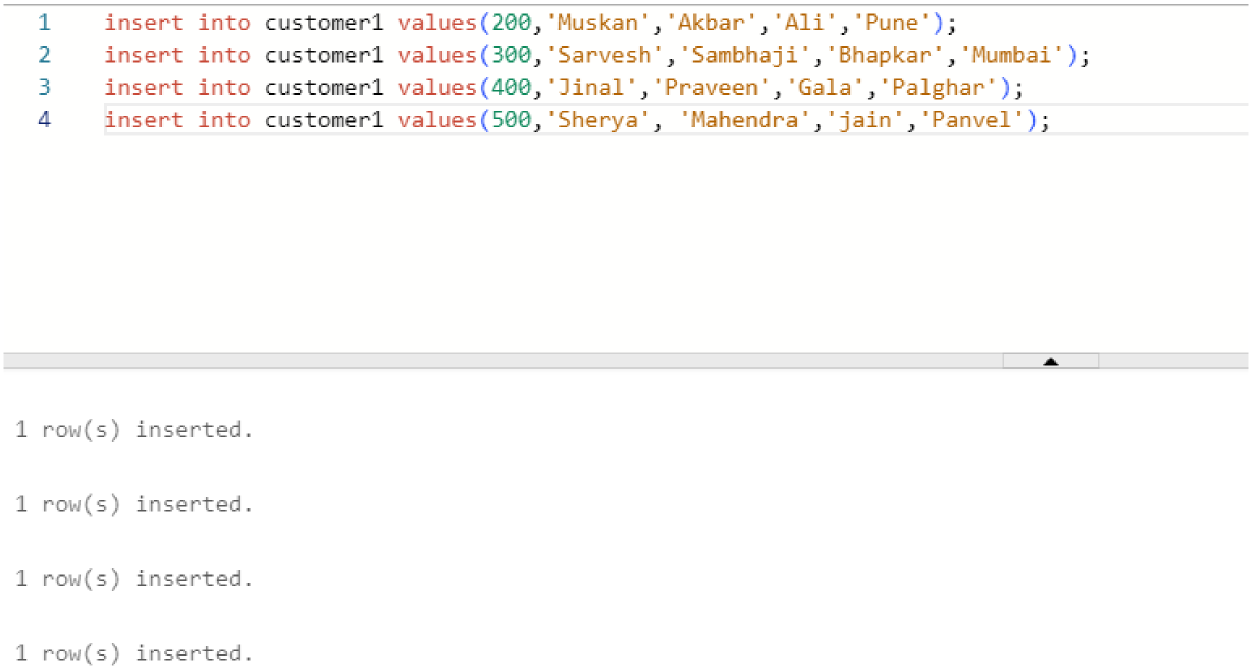
```
create table customer1 (
cid number not null,
fname varchar2(20),
mname varchar2(20),
lname varchar2(20),
address varchar2(20)
)
PARTITION BY RANGE(cid)
(
PARTITION c100 VALUES LESS THAN(100),
PARTITION c200 VALUES LESS THAN(200),

PARTITION c300 VALUES LESS THAN(300),
PARTITION c400 VALUES LESS THAN(400),
PARTITION cother VALUES LESS THAN (MAXVALUE));
```



```
Table created.
```

2 Insert the appropriate records into the table (Also insert some ids with value more than

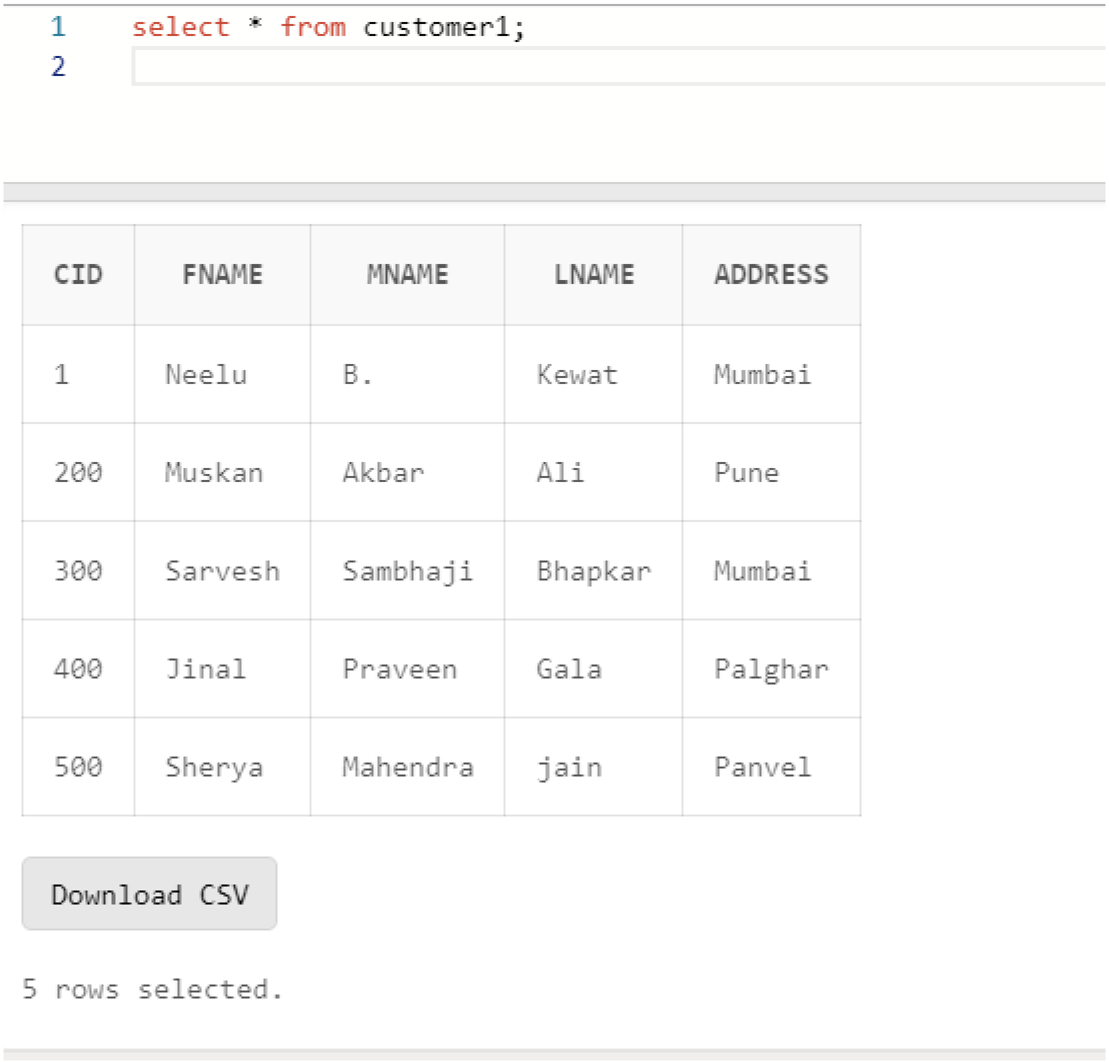
400)



```sql
insert into customer1 values(200,'Muskan','Akbar','Ali','Pune');
insert into customer1 values(300,'Sarvesh','Sambhaji','Bhapkar','Mumbai');
insert into customer1 values(400,'Jinal','Praveen','Gala','Palghar');
insert into customer1 values(500,'Sherya', 'Mahendra','jain','Panvel');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

3 Retrieve the values from the table



```
select * from customer1;
```

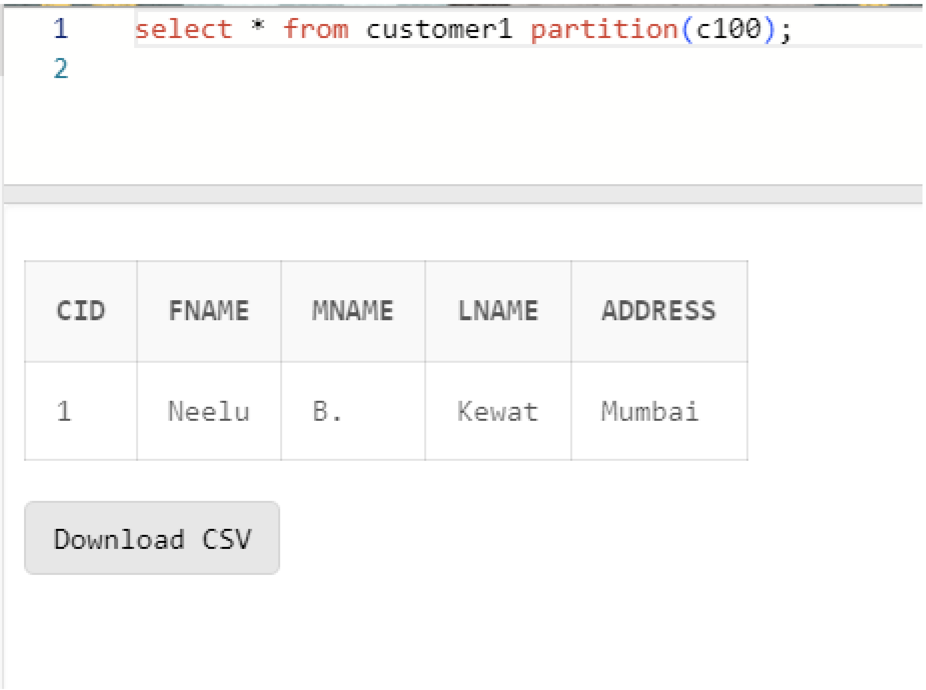
CID	FNAME	MNAME	LNAME	ADDRESS
1	Neelu	B.	Kewat	Mumbai
200	Muskan	Akbar	Ali	Pune
300	Sarvesh	Sambhaji	Bhapkar	Mumbai
400	Jinal	Praveen	Gala	Palghar
500	Sherya	Mahendra	jain	Panvel

```
5 rows selected.
```

-

4 Retrieve the values from individual partitions

Output:

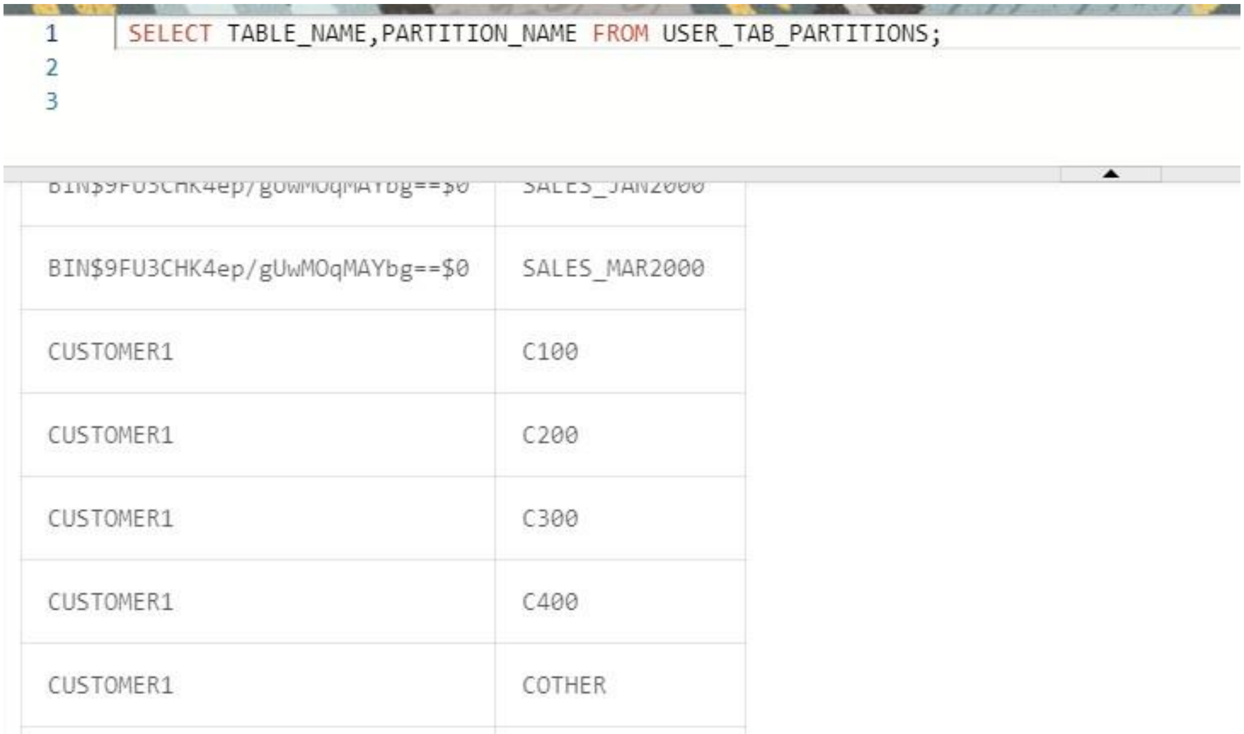


```
select * from customer1 partition(c100);
```

CID	FNAME	MNAME	LNAME	ADDRESS
1	Neelu	B.	Kewat	Mumbai

5 Retrieve the partition details from system table.

Output:



```
SELECT TABLE_NAME,PARTITION_NAME FROM USER_TAB_PARTITIONS;
```

BIN$9FU3CHK4ep/gUwMOqMAYbg==$0	SALES_JAN2000
BIN$9FU3CHK4ep/gUwMOqMAYbg==$0	SALES_MAR2000
CUSTOMER1	C100
CUSTOMER1	C200
CUSTOMER1	C300
CUSTOMER1	C400
CUSTOMER1	COTHER

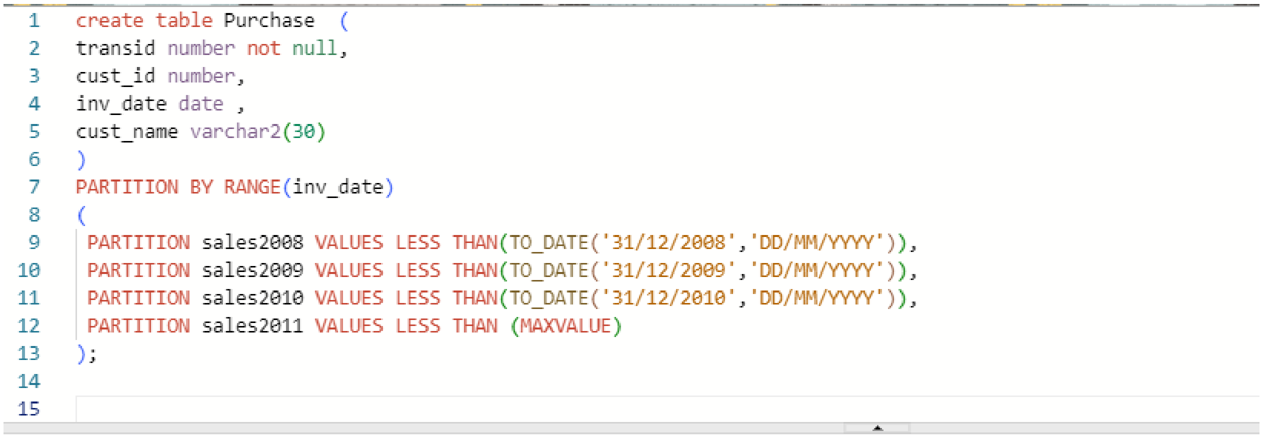
2. Create a table with following schema

   Table name: Purchase
   transid number not null
   cust_id number
   inv_date date
   cust_name varchar2(30)

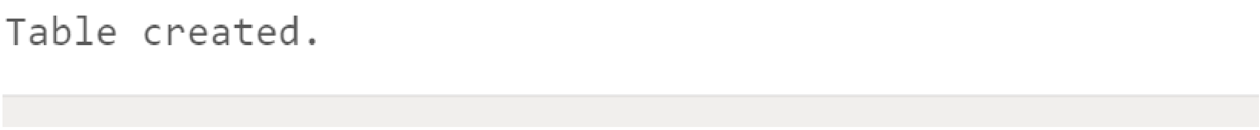
- Partition the table according to inv_date such that it has 4 partitions having:

  Data of 2008 & previous years,
  Data of 2009
  Data of 2010
  Data of 2011 & onwards.

Output:



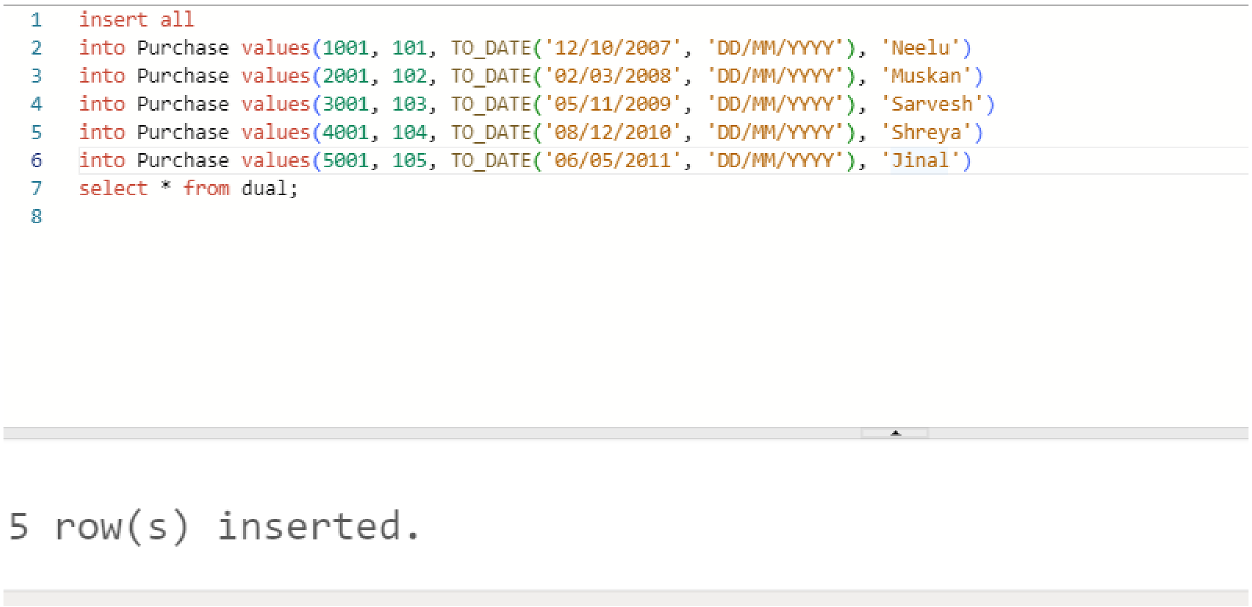
```sql
create table Purchase  (
transid number not null,
cust_id number,
inv_date date ,
cust_name varchar2(30)
)
PARTITION BY RANGE(inv_date)
(
 PARTITION sales2008 VALUES LESS THAN(TO_DATE('31/12/2008','DD/MM/YYYY')),
 PARTITION sales2009 VALUES LESS THAN(TO_DATE('31/12/2009','DD/MM/YYYY')),
 PARTITION sales2010 VALUES LESS THAN(TO_DATE('31/12/2010','DD/MM/YYYY')),
 PARTITION sales2011 VALUES LESS THAN (MAXVALUE)
);
```



```
Table created.
```

- Insert the appropriate records into the table

Output:

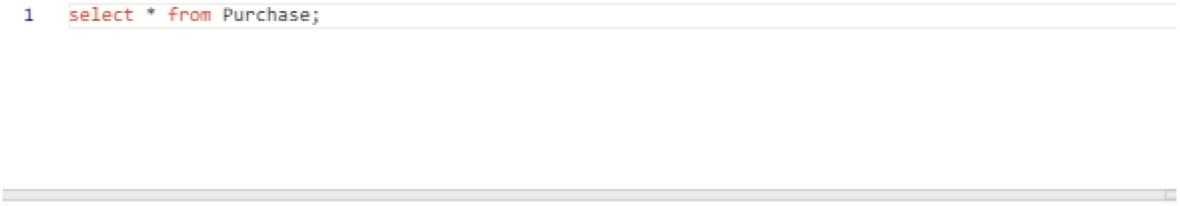


```
insert all
into Purchase values(1001, 101, TO_DATE('12/10/2007', 'DD/MM/YYYY'), 'Neelu')
into Purchase values(2001, 102, TO_DATE('02/03/2008', 'DD/MM/YYYY'), 'Muskan')
into Purchase values(3001, 103, TO_DATE('05/11/2009', 'DD/MM/YYYY'), 'Sarvesh')
into Purchase values(4001, 104, TO_DATE('08/12/2010', 'DD/MM/YYYY'), 'Shreya')
into Purchase values(5001, 105, TO_DATE('06/05/2011', 'DD/MM/YYYY'), 'Jinal')
select * from dual;
```

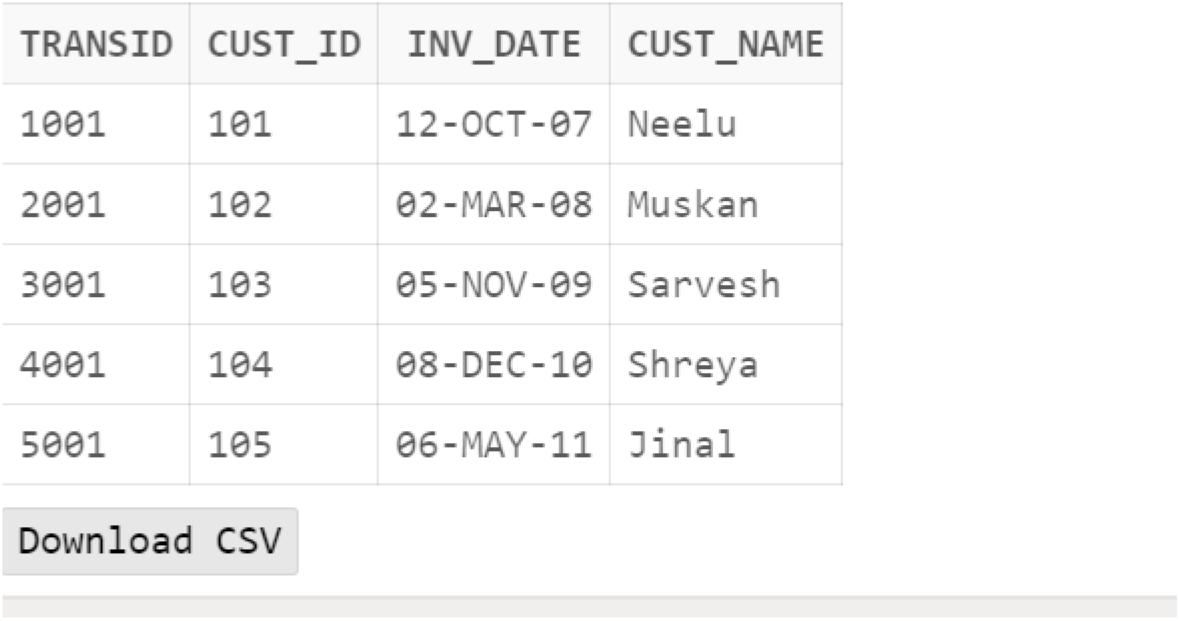
```
5 row(s) inserted.
```

- Retrieve the values from the table

Output:



```
select * from Purchase;
```

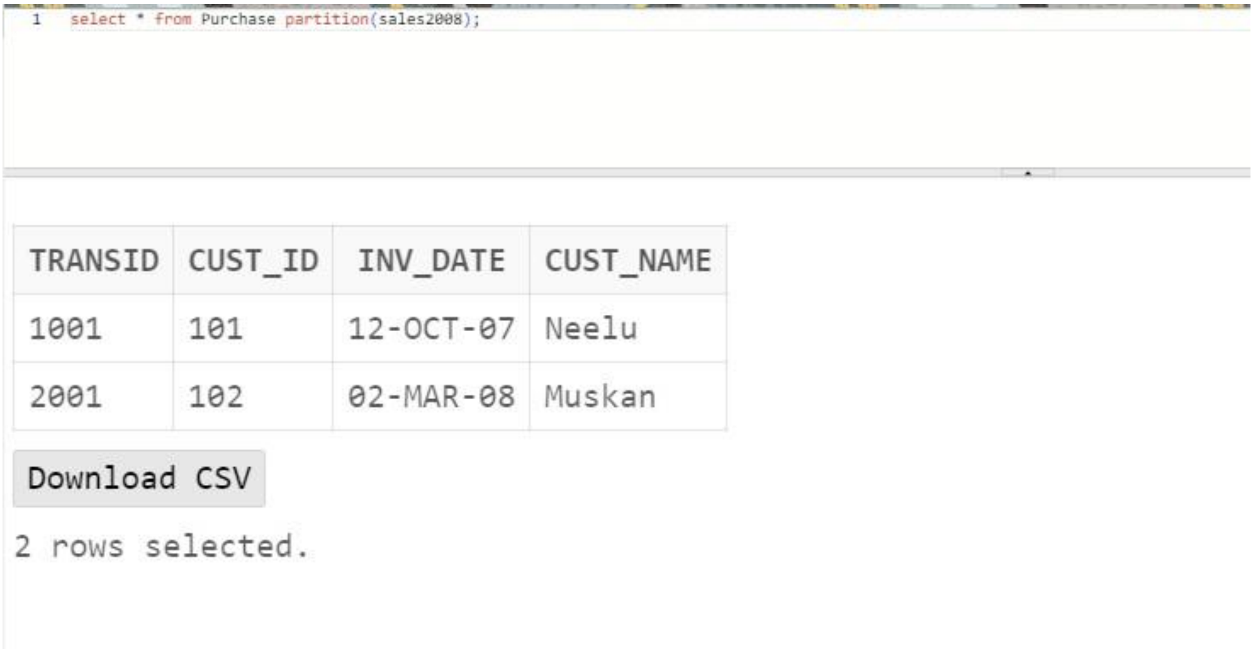


TRANSID	CUST_ID	INV_DATE	CUST_NAME
1001	101	12-OCT-07	Neelu
2001	102	02-MAR-08	Muskan
3001	103	05-NOV-09	Sarvesh
4001	104	08-DEC-10	Shreya
5001	105	06-MAY-11	Jinal

- Split the last partition so that we have a separate partition for 2011; check the entries of system table.

- Retrieve the values from individual partitions

Output:



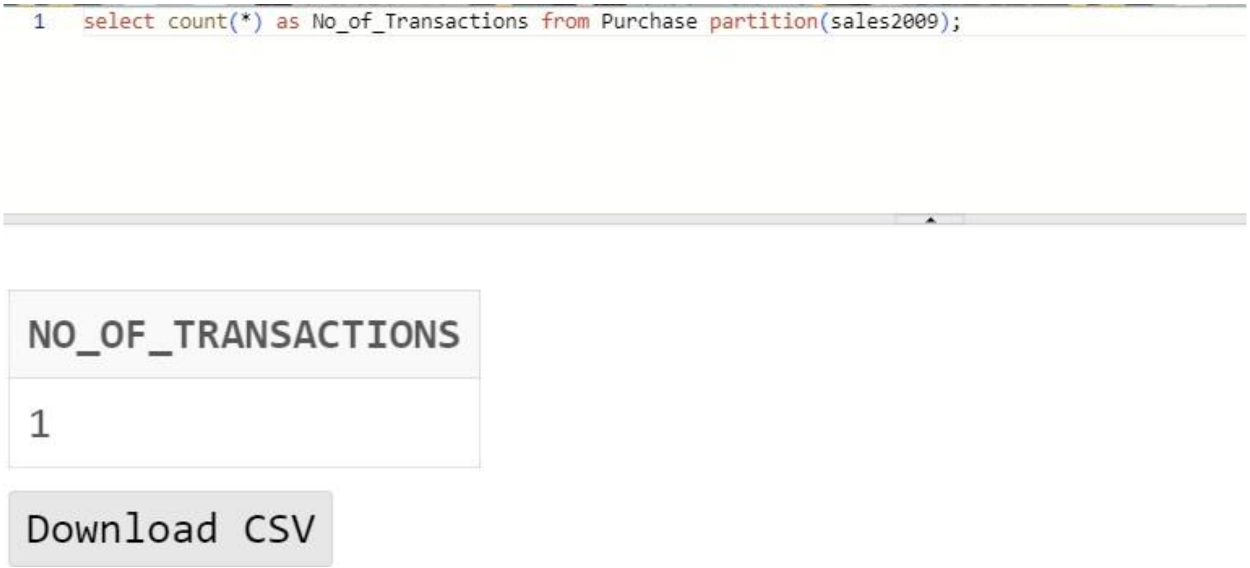
```
select * from Purchase partition(sales2008);
```

TRANSID	CUST_ID	INV_DATE	CUST_NAME
1001	101	12-OCT-07	Neelu
2001	102	02-MAR-08	Muskan

```
2 rows selected.
```

- Find the number of transactions done in the year 2009

Output:



```
select count(*) as No_of_Transactions from Purchase partition(sales2009);
```

NO_OF_TRANSACTIONS
1

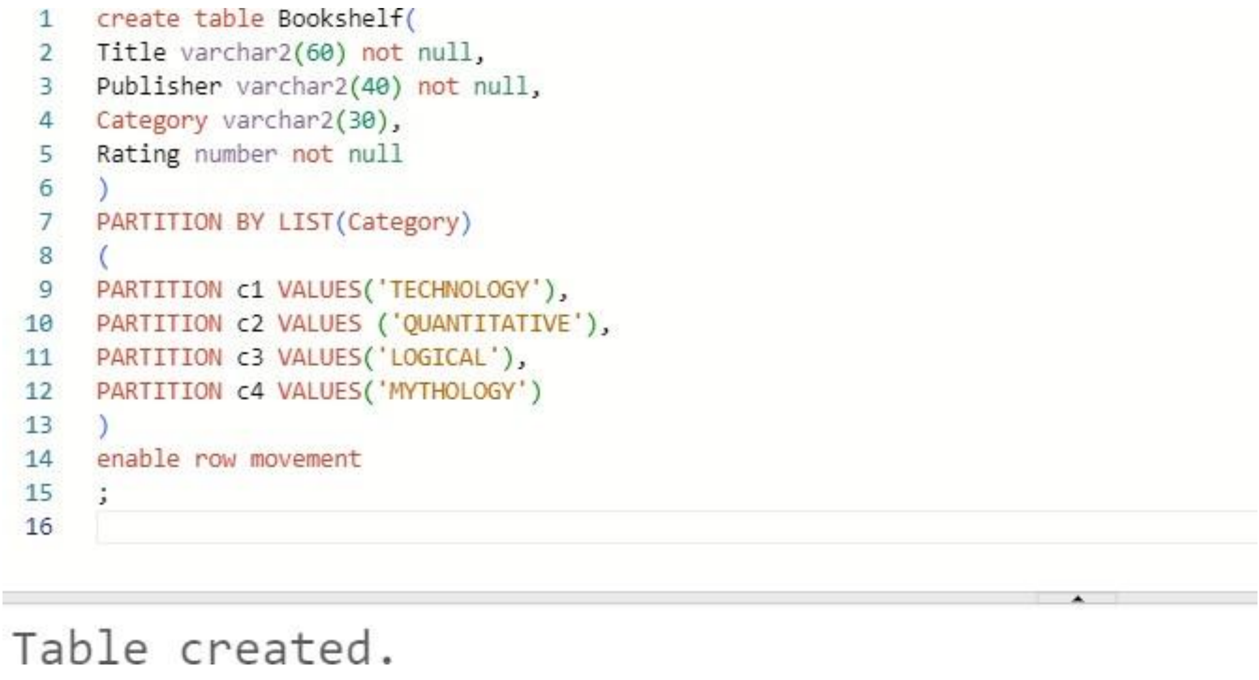
List:

1. Create a table Bookshelf with following schema & partition the Category column by using list partitioning:

Table name: Bookshelf
Title varchar2(60) not null
Publisher varchar2(40) not null
Category varchar2(30)
Rating number not null;

- Divide the data into 4 partitions using list partitioning on column Category with values 'TECHNOLOGY','QUANTITATIVE’, ‘LOGICAL’, ‘MYTHOLOGY’.

Output:

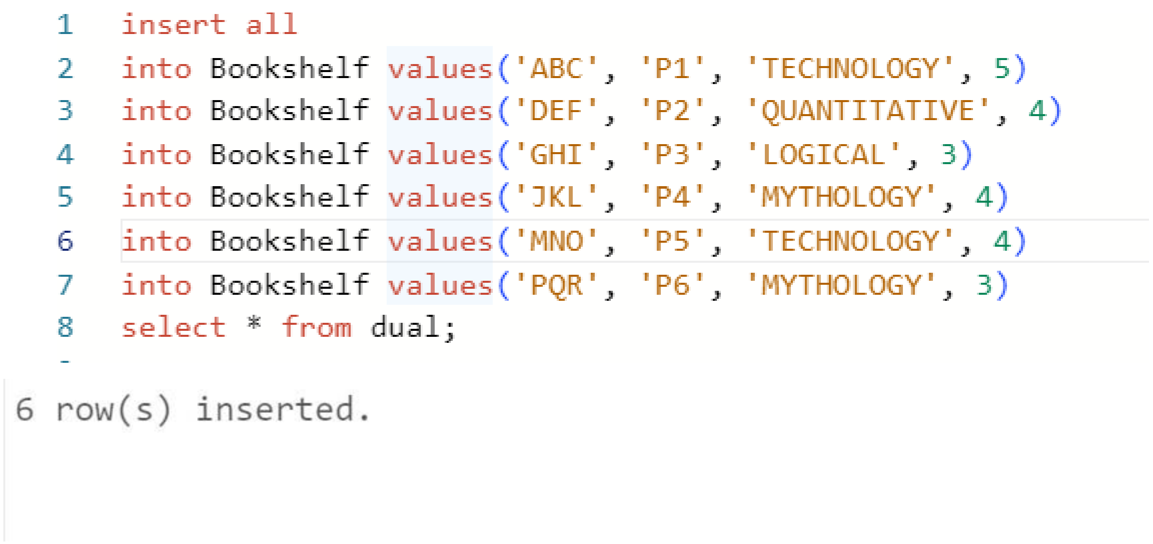


```
create table Bookshelf(
Title varchar2(60) not null,
Publisher varchar2(40) not null,
Category varchar2(30),
Rating number not null
)
PARTITION BY LIST(Category)
(
PARTITION c1 VALUES('TECHNOLOGY'),
PARTITION c2 VALUES ('QUANTITATIVE'),
PARTITION c3 VALUES('LOGICAL'),
PARTITION c4 VALUES('MYTHOLOGY')
)
enable row movement
;
```

```
Table created.
```

- Insert the appropriate records into the table

Output:

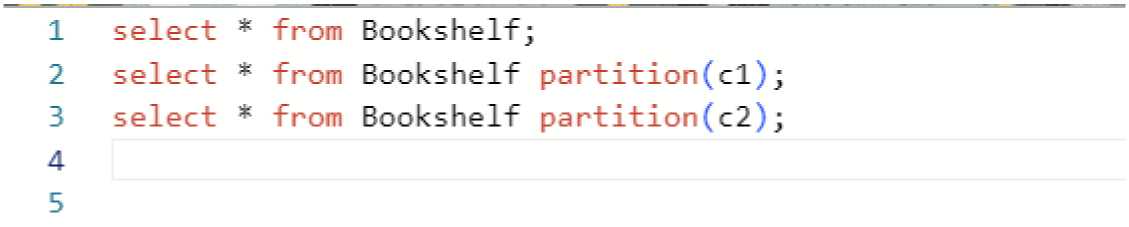


```
insert all
into Bookshelf values('ABC', 'P1', 'TECHNOLOGY', 5)
into Bookshelf values('DEF', 'P2', 'QUANTITATIVE', 4)
into Bookshelf values('GHI', 'P3', 'LOGICAL', 3)
into Bookshelf values('JKL', 'P4', 'MYTHOLOGY', 4)
into Bookshelf values('MNO', 'P5', 'TECHNOLOGY', 4)
into Bookshelf values('PQR', 'P6', 'MYTHOLOGY', 3)
select * from dual;
```

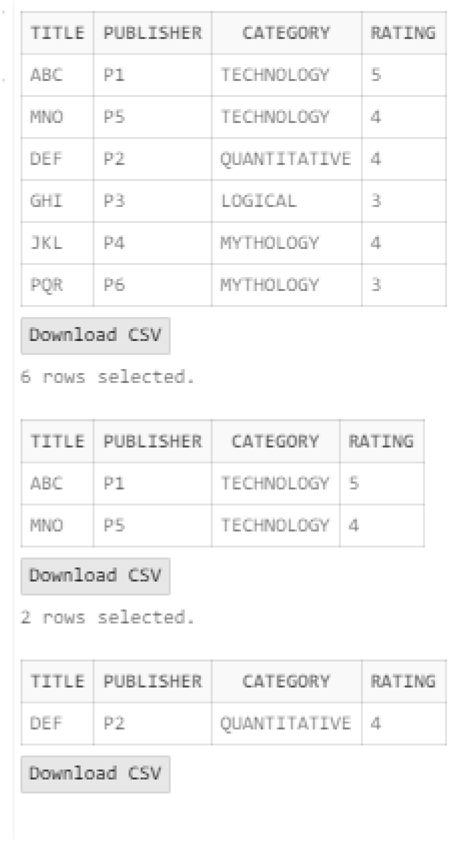
```
6 row(s) inserted.
```

- Retrieve the values from the table and from individual partitions

Output:



```
select * from Bookshelf;
select * from Bookshelf partition(c1);
select * from Bookshelf partition(c2);
```



TITLE	PUBLISHER	CATEGORY	RATING
ABC	P1	TECHNOLOGY	5
MNO	P5	TECHNOLOGY	4
DEF	P2	QUANTITATIVE	4
GHI	P3	LOGICAL	3
JKL	P4	MYTHOLOGY	4
PQR	P6	MYTHOLOGY	3

6 rows selected.

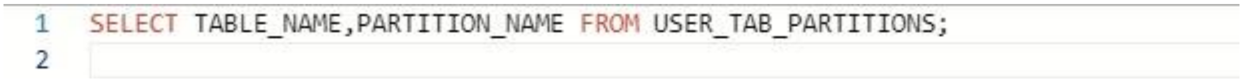
TITLE	PUBLISHER	CATEGORY	RATING
ABC	P1	TECHNOLOGY	5
MNO	P5	TECHNOLOGY	4

2 rows selected.

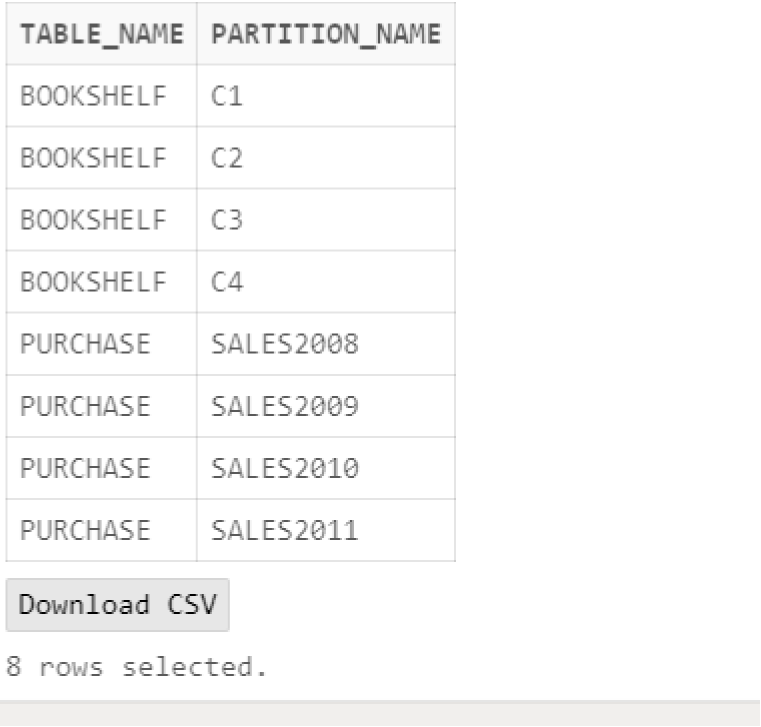
TITLE	PUBLISHER	CATEGORY	RATING
DEF	P2	QUANTITATIVE	4

- Retrieve the partition details from system table

Output:



```
SELECT TABLE_NAME,PARTITION_NAME FROM USER_TAB_PARTITIONS;

```



TABLE_NAME	PARTITION_NAME
BOOKSHELF	C1
BOOKSHELF	C2
BOOKSHELF	C3
BOOKSHELF	C4
PURCHASE	SALES2008
PURCHASE	SALES2009
PURCHASE	SALES2010
PURCHASE	SALES2011

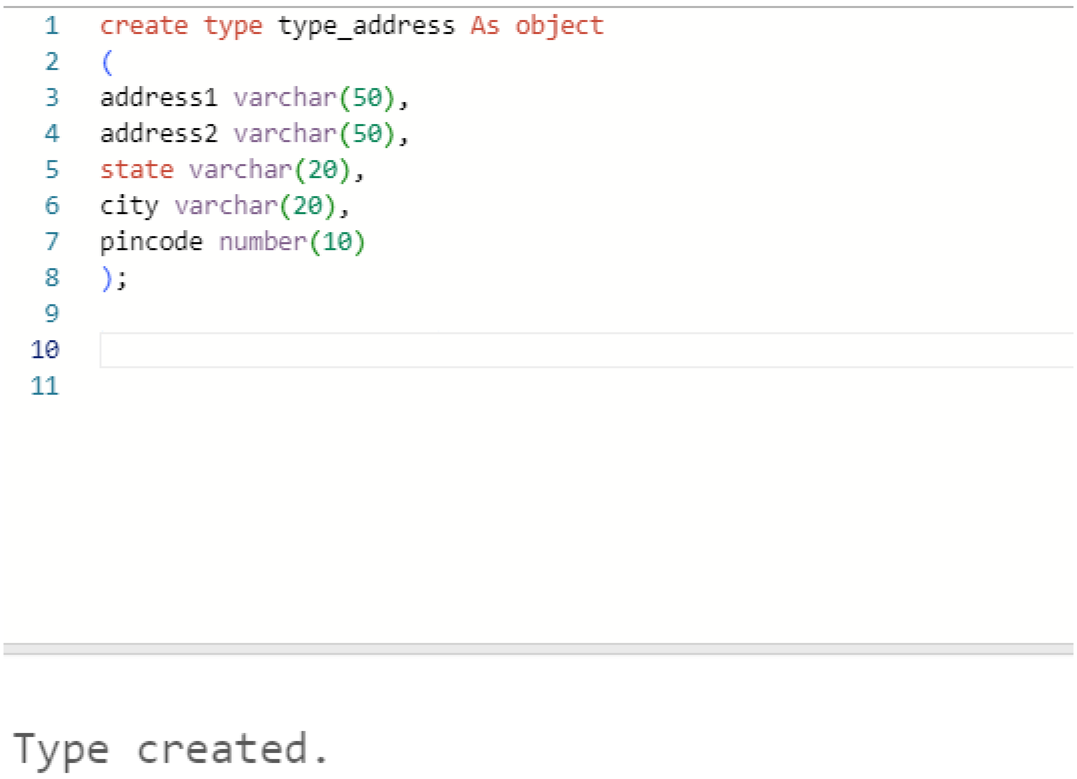
```
8 rows selected.
```

❖ Implementation of ORDBMS using ADT (Abstract Data Types)

1. Create type **Address** having the specified columns (address1, address2, state, city, pincode).

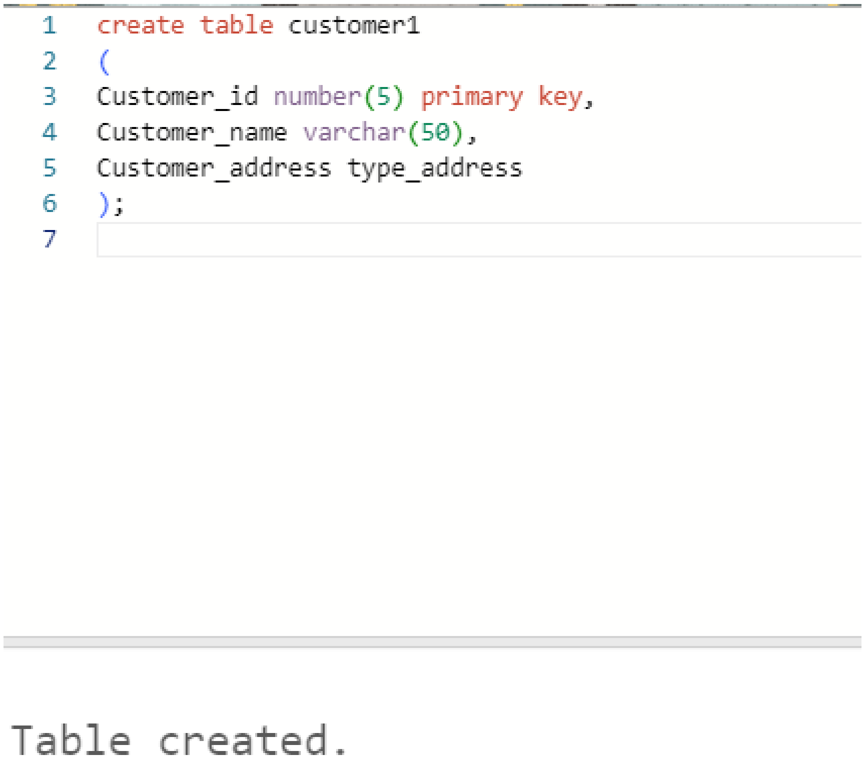
   Create **Customer** table having the specified columns (Customer_id, Customer_name and Address type).

Output:



```sql
create type type_address As object
(
address1 varchar(50),
address2 varchar(50),
state varchar(20),
city varchar(20),
pincode number(10)
);


```

```
Type created.
```

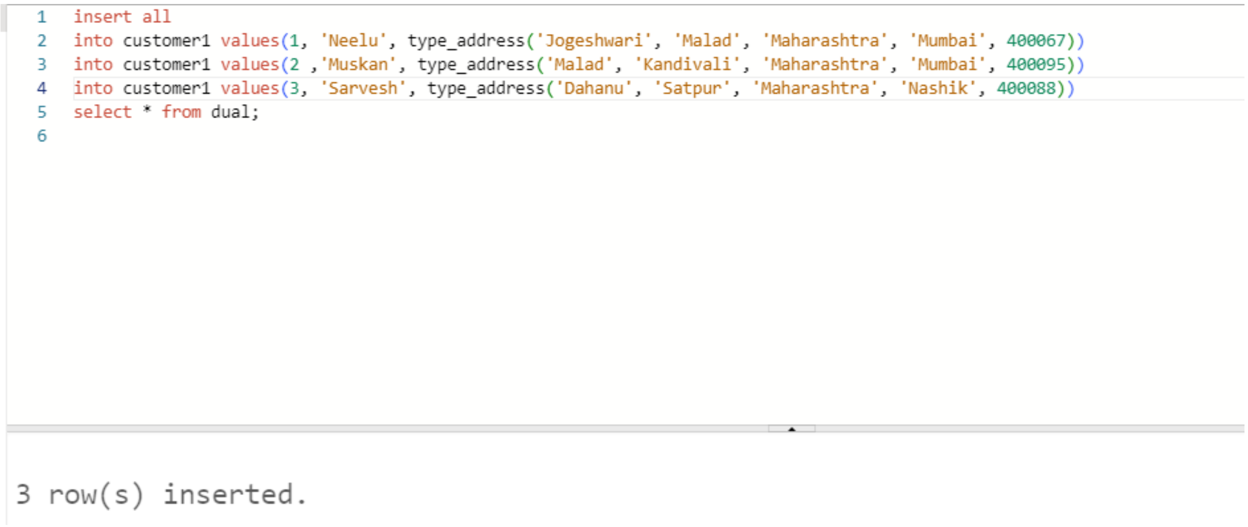


```sql
create table customer1
(
Customer_id number(5) primary key,
Customer_name varchar(50),
Customer_address type_address
);

```

```
Table created.
```

- Insert records into customer table.

Output:

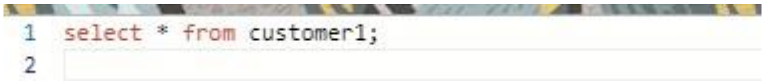


```
insert all
into customer1 values(1, 'Neelu', type_address('Jogeshwari', 'Malad', 'Maharashtra', 'Mumbai', 400067))
into customer1 values(2 ,'Muskan', type_address('Malad', 'Kandivali', 'Maharashtra', 'Mumbai', 400095))
into customer1 values(3, 'Sarvesh', type_address('Dahanu', 'Satpur', 'Maharashtra', 'Nashik', 400088))
select * from dual;

```

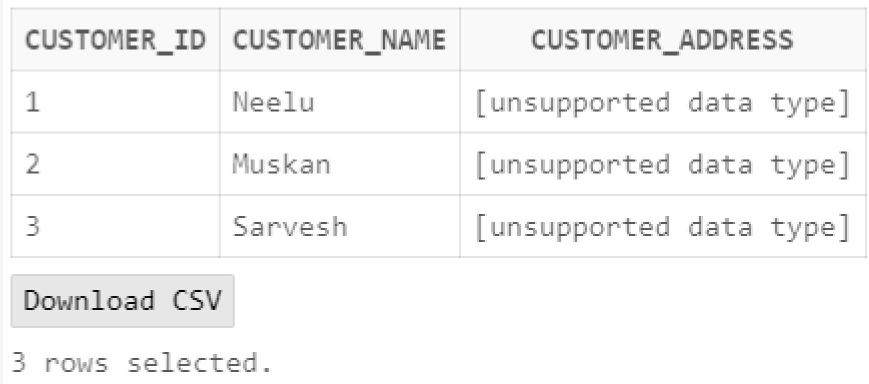
```
3 row(s) inserted.
```

- Display the details customer.

Output:



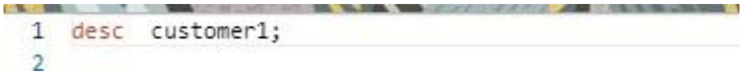
```
select * from customer1;

```



CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS
1	Neelu	[unsupported data type]
2	Muskan	[unsupported data type]
3	Sarvesh	[unsupported data type]

```
3 rows selected.
```

- Display the description/structure of the customer table



```
desc  customer1;

```

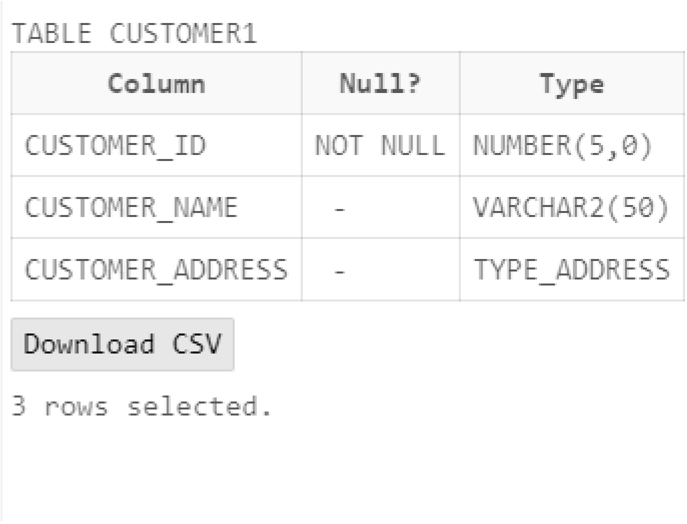


TABLE CUSTOMER1

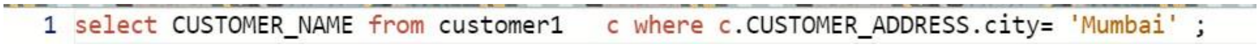
Column	Null?	Type
CUSTOMER_ID	NOT NULL	NUMBER(5,0)
CUSTOMER_NAME	-	VARCHAR2(50)
CUSTOMER_ADDRESS	-	TYPE_ADDRESS

3 rows selected.

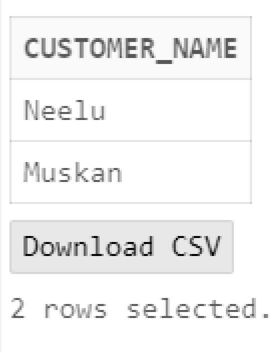
Output:

- List the customers from Mumbai

Output:



```
select CUSTOMER_NAME from customer1   c where c.CUSTOMER_ADDRESS.city= 'Mumbai' ;
```

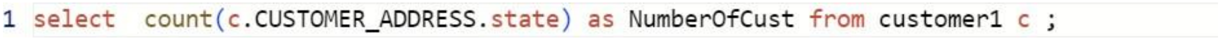


CUSTOMER_NAME
Neelu
Muskan

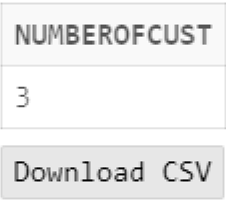
2 rows selected.

- Count the number of customers’ state wise

Output:



```
select  count(c.CUSTOMER_ADDRESS.state) as NumberOfCust from customer1 c ;
```



NUMBEROFCUST
3