

Practical 5

AIM: Distributed databases- Horizontal Fragmentation

Perform Horizontal Fragmentation

- Create a global conceptual schema Emp (Eno; Ename; Address; Email; Salary) and insert 10 records.
- Divide Emp into Horizontal fragments using the condition that Emp1 contains the tuples with salary $\leq 10,000$ and Emp2 with $10,000 < \text{salary} \leq 20,000$ on two different nodes.

What is Distributed Database?

- A distributed database is basically a database that is not limited to one system, it is spread over different sites, i.e, on multiple computers or over a network of computers.
- A distributed database system is located on various sites that don't share physical components.
- This may be required when a particular database needs to be accessed by various users globally.
- It needs to be managed such that for the users it looks like one single database.

Distributed Data Storage:

There are 2 ways in which data can be stored on different sites. These are:

1. Replication
2. Fragmentation

Fragmentation of relations can be done in two ways:

- ✓ Horizontal fragmentation - Splitting by rows
- ✓ Vertical fragmentation - Splitting by columns

What is Horizontal fragmentation?

- Horizontal fragmentation refers to the process of dividing a table horizontally by assigning each row or (a group of rows) of relation to one or more fragments.
- These fragments are then be assigned to different sides in the distributed system.
- Some of the rows or tuples of the table are placed in one system and the rest are placed in other systems.
- The rows that belong to the horizontal fragments are specified by a condition on one or more attributes of the relation.

In relational algebra horizontal fragmentation on table T, can be represented as follows:

$\sigma p(T)$

where, σ is relational algebra operator for selection
 p is the condition satisfied by a horizontal fragment

Fire the following queries:

1. Find the salary of all employees.
2. Find the Email of all employees where salary=12,000.
3. Find the employee name and Email where employee number is known.
4. Find the employee's name where salary>20,000.

Step 1: Open the sql with system/root user.

Check global database using the following command:

SQL> select * from global_name;

```
SQL> spool 'D:\SADIQ\MSc\SEM 1\ADT\PRAC\HF\sys.txt'
SQL>
SQL> select * from global_name;

GLOBAL_NAME
-----
XE
```

So our global database name is XE.

Step 2: Connect to Global database using the following command:

SQL> Connect system@xe

```
SQL> Connect system@xe
Enter password:
Connected.
```

Step 3: To get access to create user we have to alter session.

Alter session using the following command:

SQL> alter session set "_ORACLE_SCRIPT"=true;

```
SQL> alter session set "_ORACLE_SCRIPT"=true;
Session altered.
```

Step 4: Now we have access to create user and give them privilege access.

Using the following command create user and give them privilege

```
SQL> create USER user1 IDENTIFIED by u1;
```

User created.

```
SQL> Grant create session, create table,create sequence,create view,create procedure,connect,resource, create database link, unlimited tablespace to user1;
```

Grant succeeded.

```
SQL> create USER user2 IDENTIFIED by u2;
```

User created.

```
SQL> Grant create session, create table,create sequence,create view,create procedure,connect,resource, create database link, unlimited tablespace to user2;
```

Grant succeeded.

```
SQL> create USER user3 IDENTIFIED by u3;
```

User created.

```
SQL> Grant create session,create table,create sequence,create view,create procedure,connect,resource, create database link, unlimited tablespace to user3;
```

Grant succeeded.

```
SQL> create USER user1 IDENTIFIED by u1;
User created.

SQL> Grant create session, create table,create sequence,create view,create procedure,connect,resource, create database link, unlimited tablespace to user1;
Grant succeeded.

SQL> create USER user2 IDENTIFIED by u2;
User created.

SQL> Grant create session, create table,create sequence,create view,create procedure,connect,resource, create database link, unlimited tablespace to user2;
Grant succeeded.

SQL> create USER user3 IDENTIFIED by u3;
User created.

SQL> Grant create session,create table,create sequence,create view,create procedure,connect,resource, create database link, unlimited tablespace to user3;
Grant succeeded.
```

Step 5: Create table in system:

```
SQL> create table emp(eno number(3) primary key,ename varchar2(20),address varchar2(30),email varchar2(30),sal number(6));
```

```
SQL> create table emp(eno number(3) primary key,ename varchar2(20),address varchar2(30),email varchar2(30),sal number(6));
Table created.
```

Step 6: Insert 10 values in the table created:

SQL> insert into emp values(101,'sadiq','bandra','sadiq@gmail.com',21000);

1 row created.

SQL> insert into emp values(102,'sova','cst', 'sova@gmail.com',9000);

1 row created.

SQL> insert into emp values(103,'reyna','dadar','reyna@gmail.com',12000);

1 row created.

SQL> insert into emp values(104,'sage', 'andheri','sage@gmail.com',8600);

1 row created.

SQL> insert into emp values(105,'viper','marine lines', 'viper@gmail.com',
18000);

1 row created.

SQL> insert into emp values(106,'harbor','chruchgate', 'harbor@gmail.com',
24000);

1 row created.

SQL> insert into emp values(107,'brim', 'mahim','brim@gmail.com',9900);

1 row created.

SQL> insert into emp values(108,'omen','sandhrust road','omen@gmail.com',
16000);

1 row created.

SQL> insert into emp values(109,'pheonix','cotton green','pheonix@gmail.com',
9999);

1 row created.

SQL> insert into emp values(110,'fade','khar','fade@gmail.com',10001);

1 row created.

Step 7: Perform commit for system

SQL> commit;

```
SQL> commit;  
Commit complete.
```

Step 8: Now open another sqlplus and login with user1 id and password

SQL> connect user1@xe

```
SQL> spool 'D:\SADIQ\MSc\SEM 1\ADT\PRAC\HF\u1.txt'
SQL> connect user1@xe
Enter password:
Connected.
```

Step 9: Create database link to global database in order to access the data of global database table.

SQL> create database link l1 connect to system identified by rdnc using 'xe';

```
SQL> create database link l1 connect to system identified by rdnc using 'xe';
Database link created.
```

Step 10: Fire select query to check if the link to our global database is functioning well.

SQL> select * from emp@l1;

```
SQL> select * from emp@l1;

      ENO ENAME          ADDRESS
-----
EMAIL          SAL
-----
      101 sadiq          bandra
sadiq@gmail.com 21000

      102 sova          cst
sova@gmail.com  9000

      103 reyna         dadar
reyna@gmail.com 12000

      ENO ENAME          ADDRESS
-----
EMAIL          SAL
-----
      104 sage          andheri
sage@gmail.com  8600

      105 viper         marine lines
viper@gmail.com 18000
```

```

      106 harbor          chruchgate
harbor@gmail.com        24000

      ENO  ENAME          ADDRESS
-----
EMAIL          SAL
-----
      107 brim           mahim
brim@gmail.com    9900

      108 omen           sandhrust road
omen@gmail.com    16000

      109 pheonix        cotton green
pheonix@gmail.com 9999

      ENO  ENAME          ADDRESS
-----
EMAIL          SAL
-----
      110 fade           khar
fade@gmail.com    10001

10 rows selected.

```

Step 11: Our link is working perfectly fine. Now create table and insert the data from global database table where all sal<=10000.

SQL> create table emp1 as select * from emp@l1 where sal<=10000;

```

SQL> create table emp1 as select * from emp@l1 where sal<=10000;

Table created.

```

Step 12: We have inserted data from the global database into our emp 1 table. Now view that table:

SQL> select * from emp1;

```

SQL> select * from emp1;

      ENO  ENAME          ADDRESS
-----
EMAIL          SAL
-----
      102 sova           cst
sova@gmail.com  9000

      104 sage           andheri
sage@gmail.com  8600

      107 brim           mahim
brim@gmail.com  9900

      ENO  ENAME          ADDRESS
-----
EMAIL          SAL
-----
      109 pheonix        cotton green
pheonix@gmail.com 9999

```

We can see all the Employee with salary below 10000 from global database is presented in our table.

Step 13: Perform commit for User1

SQL> commit;

```
SQL> commit;
Commit complete.
SQL> spool end;
SQL> _
```

Step 14: Now open another sqlplus and login with user2 id and password

SQL> connect user2@xe

```
SQL> spool 'D:\SADIQ\MSc\SEM 1\ADT\PRAC\HF\u2.txt'
SQL> connect user2@xe
Enter password:
Connected.
```

Step 15: Create database link to global database in order to access the data of global database table.

SQL> create database link l2 connect to system identified by rdnc using 'xe';

```
SQL> create database link l2 connect to system identified by rdnc using 'xe';
Database link created.
```

Step 16: Our link is created. Now create table and insert the data from global database table where all 10000 < sal <= 20000.

SQL> create table emp2 as select * from emp@l2 where sal>10000 and sal<=20000;

```
SQL> create table emp2 as select * from emp@l2 where sal>10000 and sal<=20000;
Table created.
```

Step 17: We have inserted data from the global database into our emp 2 table. Now view that table:

SQL> select * from emp2;

```
SQL> select * from emp2;
```

| ENO | ENAME | ADDRESS |
|-----------------|-------|----------------|
| 103 | reyna | dadar |
| reyna@gmail.com | | 12000 |
| 105 | viper | marine lines |
| viper@gmail.com | | 18000 |
| 108 | omen | sandhrust road |
| omen@gmail.com | | 16000 |

| ENO | ENAME | ADDRESS |
|----------------|-------|---------|
| 110 | fade | khar |
| fade@gmail.com | | 10001 |

We can see all the Employee with salary between 10000 and 20000 from global database is presented in our table.

Step 13: Perform commit for User2

SQL> commit;

```
SQL> commit;
Commit complete.
SQL> spool end;
SQL> _
```

Step 14: To maintain consistency of the data in all the distributed tables, we will create a trigger in global database. Before that we will create link for user 1 and user 2 in global database.

SQL> create database link l2 connect to user2 identified by u2 using 'xe';

Database link created.

SQL> create database link l1 connect to user1 identified by u1 using 'xe';

Database link created.


```
SQL>
SQL> create database link l2 connect to user2 identified by u2 using 'xe';
Database link created.

SQL> create database link l1 connect to user1 identified by u1 using 'xe';
Database link created.
```

Step 15: Now we will create a inserting trigger in global database. So if we insert any new data in global database it will be reflected in distributed table.

I will be inserting a new data with salary between 10000 and 20000. So it will be reflected in our user2's emp2 table with the help of trigger.

```
SQL> create or replace trigger sadiq
2  after insert on emp
3  for each row
4  begin
5  if :new.sal>10000 and :new.sal<=20000 then
6  insert into emp2@l2 values
7  (:new.eno,:new.ename,:new.address,:new.email,:new.sal);
8  else
9  insert into emp1@l1 values
10 (:new.eno,:new.ename, :new.address,:new.email,:new.sal);
11 end if;
12 end;
13 /
```

```
SQL> create or replace trigger sadiq
2  after insert on emp
3  for each row
4  begin
5  if :new.sal>10000 and :new.sal<=20000 then
6  insert into emp2@l2 values
7  (:new.eno,:new.ename,:new.address,:new.email,:new.sal);
8  else
9  insert into emp1@l1 values
10 (:new.eno,:new.ename, :new.address,:new.email,:new.sal);
11 end if;
12 end;
13 /

Trigger created.
```

Step 16: The trigger is created. Let's insert one row to the global table and check whether it is being available in the distributed table or not.

```
SQL> insert into emp values(111,'swordx','pearl','swordx@gmail.com',14500);
```

1 row created.

Step 17: Perform commit for System

```
SQL> commit;
```

```
SQL> commit;
Commit complete.
SQL> spool end;
SQL> _
```

Step 18: We have inserted a new data in global table. Now check in table emp2 of user 2 if this data is being added or not, since the sal>14500 it should be added in emp2.

```
SQL> select * from emp2;
```

```
SQL> select * from emp2;

      ENO ENAME      ADDRESS
-----
EMAIL      SAL
-----
      103 reyna      dadar
reyna@gmail.com      12000

      105 viper      marine lines
viper@gmail.com      18000

      108 omen      sandhrust road
omen@gmail.com      16000

      ENO ENAME      ADDRESS
-----
EMAIL      SAL
-----
      110 fade      khar
fade@gmail.com      10001

      111 swordx      pearl
swordx@gmail.com      14500
```

We can see the new data is automatically reflected. So we can conclude our tables are consistent.

Step 17: Perform commit for User2

```
SQL> commit;
```

```
SQL> commit;
Commit complete.
SQL> spool end;
SQL> _
```

Step 18: Now we will fire our queries in the third user. So open another sqlplus and login with user3 id and password.

SQL> connect user3@xe

```
SQL> spool 'D:\SADIQ\MSc\SEM 1\ADT\PRAC\HF\u3.txt'
SQL>
SQL> connect user3@xe;
Enter password:
Connected.
```

Step 19: Create database link for all the three tables here in order to access various types of data based on given queries:

SQL> create database link l0 connect to system identified by rdnc using 'xe';

Database link created.

SQL> create database link l2 connect to user2 identified by u2 using 'xe';

Database link created.

SQL> create database link l1 connect to user1 identified by u1 using 'xe';

Database link created.

```
SQL> create database link l0 connect to system identified by rdnc using 'xe';
Database link created.

SQL> create database link l2 connect to user2 identified by u2 using 'xe';
Database link created.

SQL>
SQL> create database link l1 connect to user1 identified by u1 using 'xe';
Database link created.
```

Step 20: Fire the Queries:

1. Find the salary of all employees.

SQL> select sal from emp@l0;

```
SQL> select sal from emp@l0;

      SAL
-----
    21000
     9000
    12000
     8600
    18000
    24000
     9900
    16000
     9999
    10001
    14500

11 rows selected.
```

To find the salary of all the employee we will use the global table because it contains all the employee details.

2. Find the Email of all employees where salary=12,000.

SQL> select email from emp2@l2 where sal=12000;

```
SQL> select email from emp2@l2 where sal=12000;

EMAIL
-----
reyna@gmail.com
```

As 12000 is between 10000 and 20000 we will use emp2 of user2.

3. Find the employee Name and Email where employee number is known.

SQL> select ename,email from emp@l0 where eno=109;

```
SQL> select ename,email from emp@l0 where eno=109;

ENAME          EMAIL
-----
pheonix        pheonix@gmail.com
```

To find the employee Name and Email where employee number is known we will use the global table because it contains all the employee details.

1. Find the employee's name where salary>20,000.

SQL> select ename from emp@l0 where sal>20000;

```
SQL> select ename from emp@l0 where sal>20000;

ENAME
-----
sadiq
harbor

SQL> commit;

Commit complete.

SQL> spool end;
SQL>
```

Global Table have the employees name whose salary is greater than 20000. So we will use global table.

We have successfully performed Horizontal Fragmentation.