**ADBMS PRACTICAL**

**Practical-1,2,3**

**//For a given a global conceptual schema, divide the schema into vertical fragments and place them on different nodes. Execute queries on these fragments that will demonstrate distributed databases environment.**

**// For a given a global conceptual schema, divide the schema into horizontal fragments and place them on different nodes. Execute queries on these fragments that will demonstrate distributed databases environment.**

**// Place the replication of global conceptual schema on different nodes and execute queries that will demonstrate distributed databases environment.**

**System**

SQL\*Plus: Release 11.2.0.2.0 Production on Tue Oct 30 08:19:52 2018

Copyright (c) 1982, 2010, Oracle. All rights reserved.

SQL>

Enter user-name: system

Enter password:

Connected.

SQL> CREATE USER user1 IDENTIFIED BY user1;

User created.

SQL> CREATE USER user2 IDENTIFIED BY user2;

User created.

SQL> GRANT CONNECT,RESOURCE TO user1;

Grant succeeded.

SQL> GRANT CONNECT,RESOURCE TO user2;

Grant succeeded.

SQL> CREATE PUBLIC DATABASE LINK linksys CONNECT TO system IDENTIFIED BY server USING 'XE';

Database link created.

SQL> CREATE PUBLIC DATABASE LINK linku1 CONNECT TO user1 IDENTIFIED BY user1 USING 'XE';

Database link created.

SQL> CREATE PUBLIC DATABASE LINK linku2 CONNECT TO user2 IDENTIFIED BY user2 USING 'XE';

Database link created.

SQL> CREATE TABLE Emp(Eno NUMBER PRIMARY KEY,Ename VARCHAR2(10),Address VARCHAR2(10),Email VARCHAR2(18),Salary NUMBER);

Table created.

SQL> INSERT INTO Emp VALUES(&Eno,'&Ename','&Address','&Email',&Salary);

Enter value for eno: 01

Enter value for ename: Rizwan

Enter value for address: Kurla

Enter value for email: rizwan@gmail.com

Enter value for salary: 6500

old 1: INSERT INTO Emp VALUES(&Eno,'&Ename','&Address','&Email',&Salary)

new 1: INSERT INTO Emp VALUES(01,'Rizwan','Kurla','rizwan@gmail.com',6500)

1 row created.

SQL> /

Enter value for eno: 02

Enter value for ename: Pintu

Enter value for address: Naupada

Enter value for email: pintu@yahoo.com

Enter value for salary: 6000

old 1: INSERT INTO Emp VALUES(&Eno,'&Ename','&Address','&Email',&Salary)

new 1: INSERT INTO Emp VALUES(02,'Pintu','Naupada','pintu@yahoo.com',6000)

1 row created.

SQL> /

Enter value for eno: 03

Enter value for ename: John

Enter value for address: Dadar

Enter value for email: john@outlook.com

Enter value for salary: 5000

old 1: INSERT INTO Emp VALUES(&Eno,'&Ename','&Address','&Email',&Salary)

new 1: INSERT INTO Emp VALUES(03,'John','Dadar','john@outlook.com',5000)

1 row created.

SQL> /

Enter value for eno: 04

Enter value for ename: Abhishek

Enter value for address: Ghatkopar

Enter value for email: abhi@live.com

Enter value for salary: 4800

old 1: INSERT INTO Emp VALUES(&Eno,'&Ename','&Address','&Email',&Salary)

new 1: INSERT INTO Emp VALUES(04,'Abhishek','Ghatkopar','abhi@live.com',4800)

1 row created.

SQL> /

Enter value for eno: 05

Enter value for ename: Vikas

Enter value for address: Bandra

Enter value for email: vikas@gmail.com

Enter value for salary: 4700

old 1: INSERT INTO Emp VALUES(&Eno,'&Ename','&Address','&Email',&Salary)

new 1: INSERT INTO Emp VALUES(05,'Vikas','Bandra','vikas@gmail.com',4700)

1 row created.

SQL> SELECT \* FROM Emp;

ENO ENAME ADDRESS EMAIL SALARY

---------- ---------- ---------- ------------------ ----------

1 Rizwan Kurla rizwan@gmail.com 6500

2 PintuNaupada pintu@yahoo.com 6000

3 John Dadar john@outlook.com 5000

4 Abhishek Ghatkopar abhi@live.com 4800

5 Vikas Bandra vikas@gmail.com 4700

SQL> COMMIT;

Commit complete.

**// For a given a global conceptual schema, divide the schema into horizontal fragments and place them on different nodes. Execute queries on these fragments that will demonstrate distributed databases environment.**

**User1**

SQL> CREATE TABLE Emp1 AS SELECT \* FROM Emp@linksys WHERE Eno <=3;

Table created.

SQL> SELECT \* FROM Emp1;

ENO ENAME ADDRESS EMAIL SALARY

---------- ---------- ---------- ------------------ ----------

1 Rizwan Kurla rizwan@gmail.com 6500

2 PintuNaupada pintu@yahoo.com 6000

3 John Dadar john@outlook.com 5000

**User2**

SQL> CREATE TABLE Emp2 AS SELECT \* FROM Emp@linksys WHERE Eno BETWEEN 4 AND 5;

Table created.

SQL> SELECT \* FROM Emp2;

ENO ENAME ADDRESS EMAIL SALARY

---------- ---------- ---------- ------------------ ----------

4 Abhishek Ghatkopar abhi@live.com 4800

5 Vikas Bandra vikas@gmail.com 4700

**Q. Find the salary of all the employees.**

SQL> SELECT Salary FROM Emp1 UNION SELECT Salary FROM Emp2;

SALARY

----------

4700

4800

5000

6000

6500

**Q. Find the email of all the employees where salary is greater than 5000.**

SQL> SELECT Email FROM Emp1 WHERE Salary > 5000 UNION SELECT Email FROM Emp2 WHERE Salary > 5000;

EMAIL

------------------

pintu@yahoo.com

[rizwan@gmail.com](mailto:rizwan@gmail.com)

**Q. Find the employee name and email where employee number is known.**

SQL> SELECT Ename, Email FROM Emp1 WHERE Eno = 4 UNION SELECT Ename, Email FROM Emp2 WHERE Eno = 04;

ENAME EMAIL

---------- ------------------

Abhishek abhi@live.com

**Q. Find the employee name and address where employee number is known.**

SQL> SELECT Ename,Address FROM Emp1 WHERE Eno = 01 UNION SELECT Ename,Address FROM Emp2 WHERE Eno = 05;

ENAME ADDRESS

---------- ----------

Rizwan Kurla

**//For a given a global conceptual schema, divide the schema into vertical fragments and place them on different nodes. Execute queries on these fragments that will demonstrate distributed databases environment.**

**User1**

SQL> CREATE TABLE Emp3 AS SELECT Eno,Ename,Address FROM Emp@linksys;

Table created.

SQL> SELECT \* FROM Emp3;

ENO ENAME ADDRESS

---------- ---------- ----------

1 Rizwan Kurla

2 PintuNaupada

3 John Dadar

4 Abhishek Ghatkopar

5 Vikas Bandra

**User2**

SQL> CREATE TABLE Emp4 AS SELECT Eno,Email,Salary FROM Emp@linksys;

Table created.

SQL> SELECT \* FROM Emp4;

ENO EMAIL SALARY

---------- ------------------ ----------

1 rizwan@gmail.com 6500

2 pintu@yahoo.com 6000

3 john@outlook.com 5000

4 abhi@live.com 4800

5 vikas@gmail.com 4700

SQL>

**// Place the replication of global conceptual schema on different nodes and execute queries that will demonstrate distributed databases environment.**

SQL> CREATE OR REPLACE TRIGGER temp

AFTER INSERT ON Emp FOR EACH ROW

BEGIN

INSERT INTO Emp3 VALUES(:new.Eno, :new.Ename, :new.Address);

INSERT INTO Emp4 VALUES(:new.Eno, :new.Email, :new.Salary);

END;

/

Trigger created.

SQL> COMMIT;

Commit complete.

**Practical No 4**

**Topic:** Object Oriented Databases

**Problem Statement:**

Using Object Oriented databases create the following types:

1. AddrType (Pincode: number, Street: char, City: char, state: char)
2. BranchType (address: AddrType, phone1: integer: integer)
3. AuthorType (name: char, addr AddrType)
4. PublisherType (name: char, addr: AddrType, branches: BranchTableType
5. AuthorListType as varray, which is a reference to AuthorType

\*\*A *varray* is used to store an ordered collection of data, however it is often better to think of an array as a collection of variables of the same type.

Next create the following tables:

f) BranchTableType of BranchType

g) authors of AuthorType

h) books(title: varchar, year : date, published\_by ref PublisherType,authors AuthorListType)

i) Publishers of PublisherType

Insert 5 records into the above tables and fire the following queries:

a) List all of the authors that have the same pin code as their publisher:

b) List all books that have 2 or more authors:

c) List the name of the publisher that has the most branches

d) Name of authors who have not published a book

e) Name of authors who have not published a book.

**Overview:** An object oriented database management system (OODBMS) is the coupling of database and object oriented concepts. OODBMS concepts have their origin in object oriented programming language that contains abstract data types, encapsulation, information hiding methods, class/type hierarchies, inheritance etc.

An object has state (value) and behaviour (operation).

**For Example**- An Employee can be an object with state as fields eno, ename, bdate, salary and behavior may be finding age operation to find the age of an employee using bdate.

# Step1: Create different Types and Tables

SQL> Create or replace type AddrType1 as object(

Pincode number(5),

Street char(20),

City varchar2(50),

state varchar2(40),

no number(4) );

/

Type created.

SQL> create or replace type BranchType as object(address AddrType1,phone1 integer,

phone2 integer );

/

Type created.

SQL> create or replace type BranchTableType as table of BranchType;

/

Type created.

SQL> create or replace type AuthorType as object( name varchar2(50), addr AddrType1 );

/

Type created.

SQL> create table authors of AuthorType;

Table created.

SQL> create or replace type AuthorListType as varray(10) of ref AuthorType;

/

Type created.

SQL> create or replace type PublisherType as object( name varchar2(50),

addr AddrType1, branches BranchTableType);

/

Type created.

SQL> create table books( title varchar2(50), year date, published\_by ref PublisherType, authors AuthorListType);

Table created.

SQL> create table Publishers of PublisherType NESTED TABLE branches STORE as branchtable;

Table created.

# Step 2: Insertion of different values in the tables

**Table Name:Authors**

**Format :** Authors(name, addr:<pincode,street,city,state,no>)

SQL> insert into Authors values('stallings', AddrType1(5002,'sstreet','pune','mha',04));

1 row created.

SQL>

SQL> insert into Authors values('stallings', AddrType1(7007,'sstreet','mumbai','mha',1007));

1 row created.

SQL> insert into Authors values('Navathe', AddrType1(7008,'nstreet','nasik','mha',08));

1 row created.

SQL> insert into Authors values('Dhumdhare', AddrType1(7003,'dstreet','mumbai','mha',1003));

1 row created.

**Table name : Publishers**

**Format :** Publishers(name, addr:<pincode,street,city,state,no>, branches:set of<address:<pincode,street,city,state,no>,phone1,phone2>)

/\*where addr is an object-valued attribute. Branches is a complex-valued attribute, in this case a nested table where each element in the table has 3 parts: an address and two phones\*/

SQL> insert into Publishers values('tata',AddrType1(4002,'rstreet','mumbai','mha',03),

BranchTableType(BranchType( AddrType1(5002,'fstreet','mumbai','mha',03),23406,69896)));

1 row created.

SQL> insert into Publishers values('seth',AddrType1(7007,'sstreet','mumbai','mha',1007),

BranchTableType(BranchType(AddrType1(7007,'sstreet','mumbai','mha',1007),4543545,8676775)));

1 row created.

SQL> insert into Publishers values('joshi',AddrType1(7008,'sstreet','mumbai','mha',1007),

BranchTableType(BranchType(AddrType1(1002,'sstreet','nasik','mha',1007),456767,7675757)));

1 row created.

SQL> select \* from Authors;

**Table Name3: books**

**Format :**  books(title,year,published\_by: ref<Publishers>, authors:list of ref Author)

SQL> insert into books select 'java','28-may-1983', ref(pub), AuthorListType(ref(aut)) from

Publishers pub, Authors aut where pub.name='joshi' and aut.name=' Dhumdhare ';

0 rows created.

SQL> insert into books select 'java','28-may-1983',ref(pub),AuthorListType(ref(aut)) from

Publishers pub,Authors aut where pub.name='joshi' and aut.name='Dhumdhare';

1 row created.

SQL> insert into books select 'java','28-may-1983',ref(pub),AuthorListType(ref(aut)) from

Publishers pub,Authors aut where pub.name='seth' and aut.name='Navathe';

1 row created.

SQL> insert into books select 'adv java','28-may-1983',ref(pub), AuthorListType(ref(aut)) from Publishers pub, Authors aut where pub.name='seth' and aut.name='stallings';

2 rows created.

SQL> insert into books

select 'DBMS','28-may-1983',ref(pub),AuthorListType(ref(aut)) from

Publishers pub,Authors aut where pub.name='tata' and aut.name='Navathe';

1 row created.

SQL> select \* from books;

**Step 3: Firing queries on the created tables**

**a) List all of the authors that have the same address as their publisher:**

SQL> select a.name from authors a, publishers p where a.addr = p.addr;

NAME

--------------------------------------------------

stallings

**b) List all of the authors that have the same pin code as their publisher:**

SQL> select a.name from authors a, publishers p where a.addr.pincode = p.addr.pincode;

NAME

--------------------------------------------------

stallings

Navathe

**c) select List all books that have 2 or more authors:**

SQL> select \* from books b where 1 <( select count(\*) from table(b.authors));

no rows selected

**d) List the title of the book that has the most authors:**

SQL> Select title from books b, table(b.authors) group by title having count(\*) =

(select max(count(\*)) from books b, table(b.authors) group by title);

TITLE

--------------------------------------------------

adv java

java

1. **Name of authors who have not published a book:**

SQL> select a.name from authors a where not exists( select b.title from books b, table(b.authors) where a.name = name);

no rows selected

**Practical 5**

**// Create different types that include attributes and methods. Define tables for these types by adding sufficient number of tuples. Demonstrate insert, update and delete operations on these tables. Execute queries on them.**

SQL\*Plus: Release 11.2.0.2.0 Production on Tue Oct 30 08:19:52 2018

Copyright (c) 1982, 2010, Oracle. All rights reserved.

SQL> connect

Enter user-name: system

Enter password:

Connected.

SQL> create or replace type ttype as object(tid number ,tname varchar2(20),tcost number);

2 /

Type created.

SQL> create or replace type ctype as object(cid number ,cname varchar2(20),members number);

2 /

Type created.

SQL> create or replace type bktype as object(tid number, cid number, bkdt date, frmdt date, todt date);

2 /

SQL> create table tour of ttype;

Table created.

SQL> create table customer of ctype;

Table created.

SQL> create table booking of bktype;

Table created.

SQL> insert into tour values (ttype(001,'kashmir',20000));

1 row created.

SQL> insert into tour values (ttype(002,'kanyakumari',25000));

1 row created.

SQL> insert into tour values (ttype(005,'assam',22000));

1 row created.

SQL> insert into tour values (ttype(007,'dombivli',23000));

1 row created.

SQL> insert into tour values (ttype(008,'ghatkopar',24000));

1 row created.

SQL> insert into tour values (ttype(010,'nalasopara',26000));

1 row created.

SQL> insert into customer values (ctype(001,'aditya chachlani',3));

1 row created.

SQL> insert into customer values (ctype(003,'manish birla',2));

1 row created.

SQL> insert into customer values (ctype(004,'neeta singh',4));

1 row created.

SQL> insert into customer values (ctype(006,'kajal rathod',1));

1 row created.

SQL> insert into customer values (ctype(008,'vibhuti mishra',6));

1 row created.

SQL> insert into customer values (ctype(012,'happu singh',12));

1 row created.

SQL> select \* from tour;

TID TNAME TCOST

---------- -------------------- ----------

1 kashmir 20000

2 kanyakumari 25000

5 assam 22000

7 dombivli 23000

8 ghatkopar 24000

10 nalasopara 26000

6 rows selected.

SQL> select \* from customer;

CID CNAME MEMBERS

---------- -------------------- ----------

1 adityachachlani 3

3 manishbirla 2

4 neetasingh 4

6 kajal rathod 1

8 vibhutimishra 6

12 happusingh 12

6 rows selected.

SQL> insert into booking values (bktype(001, 012, to\_date('25-07-2018 5:45','dd-mm-yyyy hh24:mi'), to\_date('28-07-2018 7:45','dd-mm-yyyy hh24:mi'), to\_date('30-07-2018 7:45','dd-mm-yyyy hh24:mi')));

1 row created.

SQL> insert into booking values (bktype(002,006,to\_date('26-07-2018 5:45','dd-mm-yyyy hh24:mi'),to\_date('1-08-2018 7:45','dd-mm-yyyy hh24:mi'),to\_date('3-08-2018 7:45','dd-mm-yyyy hh24:mi')));

1 row created.

SQL> insert into booking values (bktype(005,003,to\_date('25-07-2018 5:45','dd-mm-yyyy hh24:mi'),to\_date('2-08-2018 7:45','dd-mm-yyyy hh24:mi'),to\_date('5-08-2018 7:45','dd-mm-yyyy hh24:mi')));

1 row created.

SQL> insert into booking values (bktype(007,008,to\_date('1-08-2018 5:45','dd-mm-yyyy hh24:mi'),to\_date('6-08-2018 7:45','dd-mm-yyyy hh24:mi'),to\_date('12-08-2018 7:45','dd-mm-yyyy hh24:mi')));

1 row created.

SQL> insert into booking values (bktype(008,004,to\_date('8-08-2018 5:45','dd-mm-yyyy hh24:mi'),to\_date('10-08-2018 7:45','dd-mm-yyyy hh24:mi'),to\_date('15-08-2018 7:45','dd-mm-yyyy hh24:mi')));

1 row created.

SQL> insert into booking values (bktype(010,001,to\_date('5-08-2018 5:45','dd-mm-yyyy hh24:mi'),to\_date('12-08-2018 7:45','dd-mm-yyyy hh24:mi'),to\_date('16-08-2018 7:45','dd-mm-yyyy hh24:mi')));

1 row created.

SQL> select \* from booking;

TID CID BKDT FRMDT TODT

---------- ---------- --------- --------- ---------

1 12 25-JUL-18 28-JUL-18 30-JUL-18

2 6 26-JUL-18 01-AUG-18 03-AUG-18

5 3 25-JUL-18 02-AUG-18 05-AUG-18

7 8 01-AUG-18 06-AUG-18 12-AUG-18

8 4 08-AUG-18 10-AUG-18 15-AUG-18

10 1 05-AUG-18 12-AUG-18 16-AUG-18

6 rows selected.

**Q. Find name of customer whose tour name is Kashmir.**

SQL> select c.cname from customer c where c.cid=(select b.cid from booking b where b.tid=(select t.tid from tour t where tname='kashmir'));

CNAME

--------------------

Happusingh

**Q. Find total number of members belongs to customer Manish Birla.**

SQL> select c.members from customer c where cname='manish birla' ;

MEMBERS

----------

2

**Q. Find name of tour who have customer with maximum number of members.**

SQL> select \* from tour t where t.tid=(select b.tid from booking b where b.cid=(select c.cid from customer c where c.members=(select max(c.members) from customer c )));

TID TNAME TCOST

---------- -------------------- ----------

1 kashmir 20000

SQL> select \* from tour t where t.tid in (select b.tid from booking b where trunc(frmdt,'mm')=to\_date('aug','mm'));

TID TNAME TCOST

---------- -------------------- ----------

2 kanyakumari 25000

5 assam 22000

7 dombivli 23000

8 ghatkopar 24000

10 nalasopara 26000

**Practical 6**

**// Create a temporal database and issue queries on it.**

SQL\*Plus: Release 11.2.0.2.0 Production on Sat Aug 18 11:35:48 2018

Copyright (c) 1982, 2010, Oracle. All rights reserved.

SQL> connect

Enter user-name: system

Enter password:

Connected.

SQL> CREATE TABLE tb1Emp\_appnt(acc\_no NUMBER(3), name VARCHAR2(10), recruitment\_date DATE, retirement\_date DATE);

Table created.

SQL> INSERT INTO tb1Emp\_appnt VALUES(101,'Aditya',TO\_DATE('2-3-2001','DD-MM-YYYY'),TO\_DATE('13-04-2005','DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO tb1Emp\_appnt VALUES(102,'Medha',TO\_DATE('12-8-2002','DD-MM-YYYY'),TO\_DATE('12-07-2004','DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO tb1Emp\_appnt VALUES(103,'Rahul',TO\_DATE('11-3-2002','DD-MM-YYYY'),TO\_DATE('13-07-2005','DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO tb1Emp\_appnt VALUES(104,'Karan',TO\_DATE('2-3-2003','DD-MM-YYYY'),TO\_DATE('13-09-2005','DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO tb1Emp\_appnt VALUES(105,'Rizwan',TO\_DATE('8-7-2000','DD-MM-YYYY'),TO\_DATE('2-03-2001','DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO tb1Emp\_appnt VALUES(106,'Alok',TO\_DATE('17-9-1999','DD-MM-YYYY'),TO\_DATE('13-12-2004','DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO tb1Emp\_appnt VALUES(107,'Ram',TO\_DATE('12-11-2001','DD-MM-YYYY'),TO\_DATE('13-02-2005','DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO tb1Emp\_appnt VALUES(108,'Ganesh',TO\_DATE('16-1-2001','DD-MM-YYYY'),TO\_DATE('2-03-2001','DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO tb1Emp\_appnt VALUES(109,'Amey',TO\_DATE('21-4-2002','DD-MM-YYYY'),TO\_DATE('13-04-2005','DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO tb1Emp\_appnt VALUES(110,'Pritish',TO\_DATE('28-2-2003','DD-MM-YYYY'),TO\_DATE('13-04-2005','DD-MM-YYYY'));

1 row created.

SQL> SELECT \* FROM tb1Emp\_appnt;

ACC\_NO NAME RECRUITME RETIREMEN

---------- ---------- --------- ---------

101 Aditya 02-MAR-01 13-APR-05

102 Medha 12-AUG-02 12-JUL-04

103 Rahul 11-MAR-02 13-JUL-05

104 Karan 02-MAR-03 13-SEP-05

105 Rizwan 08-JUL-00 02-MAR-01

106 Alok 17-SEP-99 13-DEC-04

107 Ram 12-NOV-01 13-FEB-05

108 Ganesh 16-JAN-01 02-MAR-01

109 Amey 21-APR-02 13-APR-05

110 Pritish 28-FEB-03 13-APR-05

10 rows selected.

SQL> SELECT \* FROM tb1Emp\_appnt WHERE recruitment\_date = TO\_DATE('2-03-2001','dd-mm-yyyy');

ACC\_NO NAME RECRUITME RETIREMEN

---------- ---------- --------- ---------

101 Aditya 02-MAR-01 13-APR-05

SQL> SELECT \* FROM tb1Emp\_appnt WHERE retirement\_date = TO\_DATE('2-03-2001','dd-

mm-yyyy');

ACC\_NO NAME RECRUITME RETIREMEN

---------- ---------- --------- ---------

105 Rizwan 08-JUL-00 02-MAR-01

108 Ganesh 16-JAN-01 02-MAR-01

SQL> CREATE TABLE tbl\_shares(company\_name VARCHAR2(10),no\_shares NUMBER(5),price NUMBER(5),transaction\_time DATE);

Table created.

SQL> INSERT INTO tbl\_shares VALUES('Infosys',300,10,TO\_DATE('11:45','hh24:mi'));

1 row created.

SQL> INSERT INTO tbl\_shares VALUES('Wipro',200,20,TO\_DATE('06:45','hh24:mi'));

1 row created.

SQL> INSERT INTO tbl\_shares VALUES('Himalaya',100,15,TO\_DATE('17:45','hh24:mi'));

1 row created.

SQL> INSERT INTO tbl\_shares VALUES('MBT',100,20,TO\_DATE('18:45','hh24:mi'));

1 row created.

SQL> INSERT INTO tbl\_shares VALUES('Patni',500,10,TO\_DATE('11:45','hh24:mi'));

1 row created.

SQL> SELECT \* FROM tbl\_shares;

COMPANY\_NA NO\_SHARES PRICE TRANSACTI

---------- ---------- ---------- ---------

Infosys 300 10 01-AUG-18

Wipro 200 20 01-AUG-18

Himalaya 100 15 01-AUG-18

MBT 100 20 01-AUG-18

Patni 500 10 01-AUG-18

SQL> ALTER SESSION SET nls\_date\_format = 'HH24:MI';

Session altered.

(Use the ALTER SESSION statement to set or modify any of the conditions or parameters that affect your connection to the database. The statement stays in effect until you disconnect from the database.)

SQL> SELECT \* FROM tbl\_shares;

COMPANY\_NA NO\_SHARES PRICE TRANS

---------- ---------- ---------- -----

Infosys 300 10 11:45

Wipro 200 20 06:45

Himalaya 100 15 17:45

MBT 100 20 18:45

Patni 500 10 11:45

**Q: Find all the names of a company whose share price is more than Rs. 1000 at 11:45 AM.**

SQL> SELECT no\_shares\*price FROM tbl\_shares WHERE no\_shares\*price > 1000 AND transaction\_time = TO\_DATE('11:45','hh24:mi');

NO\_SHARES\*PRICE

---------------

3000

5000

**\*Q: Find the name of company which has highestshare price at 5:00 PM.**

SQL> SELECT MAX(no\_shares\*price) FROM tbl\_shares WHERE transaction\_time< TO\_DATE('17:00','hh24:mi');

MAX(NO\_SHARES\*PRICE)

--------------------

5000

SQL>

**Practical 7**

**// Formulate a database using active rules with row and statement level.**

Copyright (c) 1982, 2010, Oracle. All rights reserved.

SQL> connect

Enter user-name: system

Enter password:

Connected.

SQL> create table project(pno varchar2(2) primary key, pname varchar2(20),thrs number);

Table created.

SQL> create table emp(eno varchar2(2) primary key,ename varchar2(20),hrs number, pno varchar2(2) references project(pno));

Table created.

SQL> insert into project values('p1','java',0);

1 row created.

SQL> insert into project values('p2','xml',0);

1 row created.

SQL> insert into project values('p3','php',0);

1 row created.

**(Create following triggers in notepad. Save within directory D with extension .sql)**

**Trigger1**

create or replace trigger t1 after insert on emp for each row

begin

update project set thrs = thrs+ :new.hrs where pno = :new.pno;

end;

/

**Trigger2**

create or replace trigger t2 after update of hrs on emp for each row

begin

update project set thrs = (thrs- :old.hrs )+:new.hrs where pno = :old.pno;

end;

/

**Trigger3**

create or replace trigger t3 after update of pno on emp for each row

begin

update project set thrs = thrs+ :new.hrs where pno = :new.pno;

update project set thrs = thrs- :old.hrs where pno = :old.pno;

end;

/

**Trigger4**

create or replace trigger t4 after delete on emp for each row

begin

update project set thrs = thrs- :old.hrs where pno = :old.pno;

end;

/

SQL> @D:\msc\t1

Trigger created.

SQL> @D:\msc\t2

Trigger created.

SQL> @D:\msc\t3

Trigger created.

SQL> @D:\msc\t4

Trigger created.

SQL> insert into emp values ('&eno','&ename',&hrs,'&pno');

Enter value for eno: 1

Enter value for ename: vijay

Enter value for hrs: 10

Enter value for pno: p1

old 1: insert into emp values ('&eno','&ename',&hrs,'&pno')

new 1: insert into emp values ('1','vijay',10,'p1')

1 row created.

SQL> /

Enter value for eno: 2

Enter value for ename: akshay

Enter value for hrs: 15

Enter value for pno: p2

old 1: insert into emp values ('&eno','&ename',&hrs,'&pno')

new 1: insert into emp values ('2','akshay',15,'p2')

1 row created.

SQL> /

Enter value for eno: 3

Enter value for ename: tejas

Enter value for hrs: 20

Enter value for pno: p2

old 1: insert into emp values ('&eno','&ename',&hrs,'&pno')

new 1: insert into emp values ('3','tejas',20,'p2')

1 row created.

SQL> /

Enter value for eno: 4

Enter value for ename: komal

Enter value for hrs: 15

Enter value for pno: p3

old 1: insert into emp values ('&eno','&ename',&hrs,'&pno')

new 1: insert into emp values ('4','komal',15,'p3')

1 row created.

SQL> select \* from project;

PN PNAME THRS

-- -------------------- ----------

p1 java 10

p2 xml 35

p3 php 15

SQL> select \* from emp;

EN ENAME HRS PN

-- -------------------- ---------- --

1 vijay 10 p1

2 akshay 15 p2

3 tejas 20 p2

4 komal 15 p3

SQL> update emp set hrs= 15 where eno=1;

1 row updated.

SQL> select \* from project;

PN PNAME THRS

-- -------------------- ----------

p1 java 15

p2 xml 35

p3 php 15

SQL> update emp set pno='p1' where eno=3;

1 row updated.

SQL> select \* from project;

PN PNAME THRS

-- -------------------- ----------

p1 java 30

p2 xml 35

p3 php 0

SQL> select \* from project;

PN PNAME THRS

-- -------------------- ----------

p1 java 30

p2 xml 35

p3 php 0

SQL> delete from emp where eno=4;

1 row deleted.

SQL> select \* from project;

PN PNAME THRS

-- -------------------- ----------

p1 java 15

p2 xml 35

p3 php 0

SQL>

**Practical 8**

**// Formulate a database using active rules with row and statement level.**

**SQL> CREATE TABLE EMPLOYEE(NAME VARCHAR2(10), SSN NUMBER(5), SALARY NUMBER(5), DNO NUMBER(3) NULL, SUPERVISOR\_SSN NUMBER(3));**

SQL> SELECT \* FROM AEMPLOYEE;

NAME SSN SALARY DNO SUPERVISOR\_SSN

---------- ---------- ---------- ---------- --------------

anita 123 10000 213 345

sunita 124 12000 214 346

nilima 125 13000 215 347

vinita 126 14000 216 348

nita 127 14000 216 348

**SQL> CREATE TABLE ADEPARTMENT (DNAME VARCHAR2(10), DNO NUMBER(3), TOTAL\_SAL NUMBER(10), MANAGER\_SSN NUMBER(3));**

SQL> SELECT \* FROM ADEPARTMENT;

DNAME DNO TOTAL\_SAL MANAGER\_SSN

---------- ---------- ---------- -----------

sales 213 10000 421

manager 214 12000 422

director 215 13000 432

production 216 28000 433

**TRIGGER 1**

CREATE OR REPLACE TRIGGER TOTAL\_SAL1

AFTER INSERT ON AEMPLOYEE

FOR EACH ROW

WHEN(NEW.DNO IS NOT NULL)

BEGIN

UPDATE ADEPARTMENT

SET TOTAL\_SAL=TOTAL\_SAL+:NEW.SALARY

WHERE DNO=:NEW.DNO;

END;

/

(Insert data into employee table and check changes in Department table)

**TRIGGER2**

CREATE OR REPLACE TRIGGER TOTAL\_SAL2

AFTER UPDATE OF SALARY ON AEMPLOYEE

FOR EACH ROW

WHEN(NEW.DNO IS NOT NULL)

BEGIN

UPDATE ADEPARTMENT

SET TOTAL\_SAL=TOTAL\_SAL+:NEW.SALARY-:OLD.SALARY

WHERE DNO=:NEW.DNO;

END;

/

(Update salary into employee table and check changes in Department table)

**TRIGGER3**

CREATE OR REPLACE TRIGGER TOTAL\_SAL3

AFTER UPDATE OF DNO ON AEMPLOYEE

FOR EACH ROW

BEGIN

UPDATE ADEPARTMENT

SET TOTAL\_SAL=TOTAL\_SAL+:NEW.SALARY

WHERE DNO=:NEW.DNO;

UPDATE ADEPARTMENT

SET TOTAL\_SAL=TOTAL\_SAL-:OLD.SALARY

WHERE DNO=:OLD.DNO;

END;

/

(Update DNO into employee table and check changes in Department table)

**TRIGGER4**

CREATE OR REPLACE TRIGGER TOTAL\_SAL4

AFTER DELETE ON AEMPLOYEE

FOR EACH ROW

WHEN(OLD.DNO IS NOT NULL)

BEGIN

UPDATE ADEPARTMENT

SET TOTAL\_SAL=TOTAL\_SAL-:OLD.SALARY

WHERE DNO=:OLD.DNO;

END;

/

(Delete any employee from employee table and check changes in Department table)

**Practical 9**

**// Create a XML data base and demonstrate insert, update and delete operations on these tables. Issue queries on it.**

SQL\*Plus: Release 11.2.0.2.0 Production on Sat Oct 27 09:06:59 2018

Copyright (c) 1982, 2010, Oracle. All rights reserved.

SQL> connect

Enter user-name: system

Enter password:

Connected.

* **Creating table with XMLType**

SQL> CREATE TABLE employee(

dept\_id NUMBER(3),

employee\_spec XMLTYPE);

Table created.

* **Inserting values into employee table**

SQL> INSERT INTO employee VALUES

(100,XMLTYPE(

'<employees>

<emp id="1">

<name>Rizwan Ansari</name>

<email>rizwan@outlook.com</email>

<acc\_no>111</acc\_no>

<dateOfJoining>2015-11-16</dateOfJoining>

</emp>

</employees>')) ;

1 row created.

SQL> INSERT INTO employee VALUES

(200,XMLTYPE(

'<employees>

<emp id="2">

<name>Vijay Sangoi</name>

<email>vijay@gmail.com.com</email>

<acc\_no>222</acc\_no>

<dateOfJoining>2016-10-28</dateOfJoining>

</emp>

</employees>')) ;

1 row created.

SQL> INSERT INTO employee VALUES

(300,XMLTYPE(

'<employees>

<emp id="3">

<name>Abhishek Yadav</name>

<email>abhishek@yahoo.com</email>

<acc\_no>333</acc\_no>

<dateOfJoining>2010-4-14</dateOfJoining>

</emp>

</employees>')) ;

1 row created.

* **Retrieve all information from employee table**

SQL> SELECT \* FROM employee;

DEPT\_ID

----------

EMPLOYEE\_SPEC

--------------------------------------------------------------------------------

100

<employees>

<emp id="1">

<name>Rizwan Ansari</name>

<email>rizwan@outl

200

<employees>

<emp id="2">

DEPT\_ID

----------

EMPLOYEE\_SPEC

--------------------------------------------------------------------------------

<name>Vijay Sangoi</name>

<email>vijay@gmail.

300

<employees>

<emp id="3">

<name>Abhishek Yadav</name>

<email>abhishek@y

* **Retrieve the name of employee from employee table**

SQL> SELECT w.employee\_spec.extract

('/employees/emp/name/text()').getStringVal() "name" FROM employee w

;

name

--------------------------------------------------------------------------------

Rizwan Ansari

Vijay Sangoi

Abhishek Yadav

* **Retrieve the acc\_no of employee from employee table**

SQL> SELECT w.employee\_spec.extract

('/employees/emp/acc\_no/text()').getStringVal() "acc\_no" FROM employee w

;

acc\_no

--------------------------------------------------------------------------------

111

222

333

* **Retrieve the names, acc\_no, email of employees from employee table**

SQL> SELECT

w.employee\_spec.extract('/employees/emp/name/text()').getStringVal() "name",

w.employee\_spec.extract('/employees/emp/acc\_no/text()').getStringVal() "acc\_no",

w.employee\_spec.extract('/employees/emp/email/text()').getStringVal() "email"

FROM employee w ;

name

--------------------------------------------------------------------------------

acc\_no

--------------------------------------------------------------------------------

email

--------------------------------------------------------------------------------

Rizwan Ansari

111

rizwan@outlook.com

Vijay Sangoi

222

vijay@gmail.com.com

name

--------------------------------------------------------------------------------

acc\_no

--------------------------------------------------------------------------------

email

--------------------------------------------------------------------------------

Abhishek Yadav

333

[abhishek@yahoo.com](mailto:abhishek@yahoo.com)

* **Updating XMLType**

SQL> UPDATE employee w SET employee\_spec = XMLTYPE

('<employees>

<emp id="5">

<name>Raheem Khan</name>

</emp>

</employees>')

WHERE

w.employee\_spec.extract('//acc\_no/text()').getStringVal() = '333'

;

1 row updated.

SQL> SELECT

w.employee\_spec.extract('/employees/emp/name/text()').getStringVal() "name" FROM employee w

;

name

--------------------------------------------------------------------------------

Rizwan Ansari

Vijay Sangoi

Raheem Khan

* **Deleting an XMLType Column Row**

SQL> DELETE FROM employee w WHERE

w.employee\_spec.extract('/employees/emp/acc\_no/text()').getStringVal() = '111'

;

1 row deleted.

SQL> SELECT

w.employee\_spec.extract('/employees/emp/acc\_no/text()').getStringVal() "acc\_no" FROM employee w

;

acc\_no

--------------------------------------------------------------------------------

222

SQL>