

Problem statement - To implement relational database of banking enterprise and Implement following DDL & DML queries.

command 1 ==> CREATE DATABASE bank;

command 2 ==> USE bank;

```
command 3 ==> CREATE TABLE customer
(
    custid VARCHAR(6),
    fname VARCHAR(30),
    mname VARCHAR(30),
    ltname VARCHAR(30),
    city VARCHAR(15),
    mobileno VARCHAR(10),
    occupation VARCHAR(10),
    dob DATE,
    CONSTRAINT customer_custid_pk PRIMARY KEY(custid)
);
```

```
command 4 ==> CREATE TABLE branch
(
    bid VARCHAR(6),
    bname VARCHAR(30),
    bcity VARCHAR(30),
    CONSTRAINT branch_bid_pk PRIMARY KEY(bid)
);
```

```
command 5 ==> CREATE TABLE account
(
    acnumber VARCHAR2(6),          -- Use VARCHAR2 instead of VARCHAR
    custid VARCHAR2(6),
    bid VARCHAR2(6),
    opening_balance NUMBER(7),     -- Use NUMBER instead of INT
    aod DATE,
    atype VARCHAR2(10),           -- Use VARCHAR2 instead of VARCHAR
    astatus VARCHAR2(10),
    CONSTRAINT account_acnumber_pk PRIMARY KEY (acnumber),
    CONSTRAINT account_custid_fk FOREIGN KEY (custid) REFERENCES customer(custid),
    CONSTRAINT account_bid_fk FOREIGN KEY (bid) REFERENCES branch(bid)
);
```

```
command 6 ==>
INSERT INTO customer VALUES ('C00001', 'Ramesh', 'Chandra', 'Sharma', 'Delhi', '9543198345', 'Service', TO_DATE('1976-12-06', 'YYYY-MM-DD'));
INSERT INTO customer VALUES ('C00002', 'Avinash', 'Sunder', 'Minha', 'Delhi', '9876532109', 'Service', TO_DATE('1974-10-16', 'YYYY-MM-DD'));
INSERT INTO customer VALUES ('C00003', 'Rahul', NULL, 'Rastogi', 'Delhi', '9765178901', 'Student', TO_DATE('1981-09-26', 'YYYY-MM-DD'));
INSERT INTO customer VALUES ('C00004', 'Parul', NULL, 'Gandhi', 'Delhi', '9876532109', 'Housewife', TO_DATE('1976-11-03', 'YYYY-MM-DD'));
INSERT INTO customer VALUES ('C00005', 'Naveen', 'Chandra', 'Aedekar', 'Mumbai', '8976523190', 'Service', TO_DATE('1976-09-19', 'YYYY-MM-DD'));
INSERT INTO customer VALUES ('C00006', 'Chitresh', NULL, 'Barwe', 'Mumbai', '7651298321', 'Student', TO_DATE('1992-11-06', 'YYYY-MM-DD'));
INSERT INTO customer VALUES ('C00007', 'Amit', 'Kumar', 'Borkar', 'Mumbai', '9875189761', 'Student', TO_DATE('1981-09-06', 'YYYY-MM-DD'));
INSERT INTO customer VALUES ('C00008', 'Nisha', NULL, 'Damle', 'Mumbai', '7954198761', 'Service', TO_DATE('1975-12-03', 'YYYY-MM-DD'));
INSERT INTO customer VALUES ('C00009', 'Abhishek', NULL, 'Dutta', 'Kolkata', '9856198761', 'Service', TO_DATE('1973-05-22', 'YYYY-MM-DD'));
INSERT INTO customer VALUES ('C00010', 'Shankar', NULL, 'Nair', 'Chennai', '8765489076', 'Service', TO_DATE('1976-07-12', 'YYYY-MM-DD'));
```

```
command 7 ==> INSERT INTO branch VALUES('B00001','Asaf ali road','Delhi');
INSERT INTO branch VALUES('B00002','New delhi main branch','Delhi');
INSERT INTO branch VALUES('B00003','Delhi cantt','Delhi');
INSERT INTO branch VALUES('B00004','Jasola','Delhi');
INSERT INTO branch VALUES('B00005','Mahim','Mumbai');
INSERT INTO branch VALUES('B00006','Vile parle','Mumbai');
INSERT INTO branch VALUES('B00007','Mandvi','Mumbai');
INSERT INTO branch VALUES('B00008','Jadavpur','Kolkata');
INSERT INTO branch VALUES('B00009','Kodambakkam','Chennai');
```

```
command 8 ==>
INSERT INTO account VALUES('A00001', 'C00001', 'B00001', 1000, TO_DATE('2012-12-15', 'YYYY-MM-DD'), 'Saving', 'Active');
INSERT INTO account VALUES('A00002', 'C00002', 'B00001', 1000, TO_DATE('2012-06-12', 'YYYY-MM-DD'), 'Saving', 'Active');
INSERT INTO account VALUES('A00003', 'C00003', 'B00002', 1000, TO_DATE('2012-05-17', 'YYYY-MM-DD'), 'Saving', 'Active');
INSERT INTO account VALUES('A00004', 'C00002', 'B00005', 1000, TO_DATE('2013-01-27', 'YYYY-MM-DD'), 'Saving', 'Active');
INSERT INTO account VALUES('A00005', 'C00005', 'B00006', 1000, TO_DATE('2012-12-17', 'YYYY-MM-DD'), 'Saving', 'Active');
INSERT INTO account VALUES('A00006', 'C00007', 'B00007', 1000, TO_DATE('2010-08-12', 'YYYY-MM-DD'), 'Saving', 'Suspended');
INSERT INTO account VALUES('A00007', 'C00007', 'B00001', 1000, TO_DATE('2012-10-02', 'YYYY-MM-DD'), 'Saving', 'Active');
INSERT INTO account VALUES('A00008', 'C00001', 'B00003', 1000, TO_DATE('2009-11-09', 'YYYY-MM-DD'), 'Saving', 'Terminated');
INSERT INTO account VALUES('A00009', 'C00003', 'B00007', 1000, TO_DATE('2008-11-30', 'YYYY-MM-DD'), 'Saving', 'Terminated');
INSERT INTO account VALUES('A00010', 'C00004', 'B00002', 1000, TO_DATE('2013-03-01', 'YYYY-MM-DD'), 'Saving', 'Active');
```

command 9 ==> DELETE FROM account WHERE custid = 'C00002';

command 10 ==> SELECT * FROM customer;

Here are the **SQL commands** to execute the practical involving **Fragmentation** and **Replication** schema design and implementation:

1. Fragmentation Schema

Horizontal Fragmentation (For `customer` Table based on `city`)

- This divides the `customer` table into fragments based on the customer's city.

```
sql
Copy code
-- Fragment 1: Customers from Delhi
CREATE TABLE customer_delhi AS
SELECT * FROM customer WHERE city = 'Delhi';
SELECT * FROM customer_delhi;

-- Fragment 2: Customers from Mumbai
CREATE TABLE customer_mumbai AS
SELECT * FROM customer WHERE city = 'Mumbai';
SELECT * FROM customer_mumbai;

-- Fragment 3: Customers from Kolkata
CREATE TABLE customer_kolkata AS
SELECT * FROM customer WHERE city = 'Kolkata';
SELECT * FROM customer_kolkata;

-- Fragment 4: Customers from Chennai
CREATE TABLE customer_chennai AS
SELECT * FROM customer WHERE city = 'Chennai';
SELECT * FROM customer_chennai;
```

Vertical Fragmentation (For `account` Table)

- This splits the `account` table into two fragments: one for personal information and another for account-related information.

```
sql
Copy code
-- Fragment 1: Personal information of the account holder
CREATE TABLE account_personal AS
SELECT acnumber, custid FROM account;
SELECT * FROM account_personal;

-- Fragment 2: Account-related information
CREATE TABLE account_financial AS
SELECT acnumber, bid, opening_balance, aod, atype, astatus FROM account;
SELECT * FROM account_financial;
```

2. Replication Schema

Full Replication (For `branch` Table)

- Replicate the entire `branch` table on two different servers to ensure data availability.

On Server 1:

```
sql
Copy code
```

```
-- Copy of the branch table on Server 1
CREATE TABLE branch_copy AS
SELECT * FROM branch;
SELECT * FROM branch_copy;
```

On Server 2:

```
sql
Copy code
-- Copy of the branch table on Server 2
CREATE TABLE branch_copy2 AS
SELECT * FROM branch;
SELECT * FROM branch_copy2;
```

Partial Replication (For customer_delhi Fragment)

- Replicate the customer_delhi fragment to another server to improve the availability of customers from Delhi.

On Main Server:

```
sql
Copy code
-- Original fragment on Main Server
CREATE TABLE customers_delhi AS
SELECT * FROM customer WHERE city = 'Delhi';
SELECT * FROM customers_delhi;
```

On Replica Server:

```
sql
Copy code
-- Copy of the customer_delhi fragment on Replica Server
CREATE TABLE customer_delhi_copy AS
SELECT * FROM customer WHERE city = 'Delhi';
SELECT * FROM customer_delhi_copy;
```

Optional: Synchronization Commands

If you need to synchronize the replicated data on different servers, you can use UPDATE or INSERT statements to maintain consistency.

```
sql
Copy code
-- Update the replicated table in case of any data change
UPDATE customer_delhi_copy
SET fname = 'Nikhil'
WHERE custid = 'C00001';
SELECT fname FROM customer_delhi_copy WHERE custid='C00001';
```

Show all the commands