DISTRIBUTED DATABASE IN ORACLE:

Consider a distributed database system involving three users: TIFFY, RUNNY, and KITTY.

1. **Write SQL queries to create the following tables at TIFFY:**
    a. STUDENT
    b. COLLEGE
    c. BRANCH
    d. COLL_BRANCH
    e. ALLOTMENT
2. **Explain how database links are created between these users and why they are necessary.**
3. **Write a trigger at TIFFY to automatically update the number of available seats in COLL_BRANCH when a new record is inserted into ALLOTMENT.**
4. **How does RUNNY ensure that an ALLOTMENT entry made at its database is reflected in TIFFY's ALLOTMENT table? Write the necessary trigger for this.**
5. **If a student with ROLLNO = 12345 is allotted to CSE at CEG, explain how the distributed database system ensures data consistency across all users.**

solution:

The following SQL code demonstrates the creation of users, tables, and database links in a **distributed Oracle database system** involving three users: **TIFFY, RUNNY, and KITTY**. The **ALLOTMENT** table is replicated across all three databases, and triggers ensure data consistency across them.

1. User Creation & Privileges (Executed at SYSTEM)

```
CREATE USER C##TIFFY IDENTIFIED BY TIFFY;
GRANT DBA TO C##TIFFY;

CREATE USER C##RUNNY IDENTIFIED BY RUNNY;
GRANT DBA TO C##RUNNY;

CREATE USER C##KITTY IDENTIFIED BY KITTY;
GRANT DBA TO C##KITTY;
```

- **Creates three database users** with DBA privileges.
- These users will manage their respective parts of the distributed database system.

2. Table Creation (Executed at TIFFY)

```
CONN C##TIFFY/TIFFY@ORCL;

CREATE TABLE STUDENT (
    HSCNO NUMBER(5) PRIMARY KEY,
    NAME VARCHAR(30),
    DOB DATE,
    GENDER CHAR CHECK(GENDER IN('M', 'F')),
    AGGR NUMBER(3)
);

CREATE TABLE COLLEGE (
    COLL_CODE NUMBER(3) PRIMARY KEY,
    NAME VARCHAR(50)
);

CREATE TABLE BRANCH (
    BCODE VARCHAR(3) PRIMARY KEY,
    BNAME VARCHAR(50)
);

CREATE TABLE COLL_BRANCH (
    COLL_CODE NUMBER(3) REFERENCES COLLEGE,
    BCODE VARCHAR(3) REFERENCES BRANCH,
    NOS NUMBER(3),
    PRIMARY KEY (COLL_CODE, BCODE)
);

CREATE TABLE ALLOTMENT (
    ROLLNO NUMBER(5) PRIMARY KEY REFERENCES STUDENT(HSCNO),
    COLL_CODE NUMBER(3),
    BCODE VARCHAR(3),
    FOREIGN KEY (COLL_CODE, BCODE) REFERENCES COLL_BRANCH
);
```

- **TIFFY is the primary database** containing all student, college, and branch details.
- The **ALLOTMENT table is replicated** in **RUNNY and KITTY** for distributed transaction handling.

3. Inserting Sample Data (Executed at TIFFY)

```
INSERT INTO STUDENT VALUES(12345, 'PETER', '13-NOV-2005', 'M', 200);
INSERT INTO STUDENT VALUES(12344, 'JAMES', '12-OCT-2005', 'M', 200);
INSERT INTO STUDENT VALUES(12355, 'JOHN', '13-JAN-2006', 'M', 200);
INSERT INTO STUDENT VALUES(12356, 'MARY', '12-MAR-2006', 'F', 200);
INSERT INTO COLLEGE VALUES(1,'CEG');
INSERT INTO COLLEGE VALUES(2,'ACTECH');
INSERT INTO COLLEGE VALUES(3,'SAP');
INSERT INTO COLLEGE VALUES(4,'MIT');

INSERT INTO BRANCH VALUES('CSE', 'COMPUTER SCIENCE AND ENGINEERING');
INSERT INTO BRANCH VALUES('IT', 'INFORMATION TECHNOLOGY');
INSERT INTO BRANCH VALUES('CH', 'CHEMICAL ENGINEERING');
INSERT INTO BRANCH VALUES('AR', 'ARCHITECTURE');

INSERT INTO COLL_BRANCH VALUES(1, 'CSE', 120);
INSERT INTO COLL_BRANCH VALUES(1, 'IT', 120);
INSERT INTO COLL_BRANCH VALUES(4, 'CSE', 120);
INSERT INTO COLL_BRANCH VALUES(4, 'IT', 120);
INSERT INTO COLL_BRANCH VALUES(3, 'AR', 20);
INSERT INTO COLL_BRANCH VALUES(2, 'CH', 30);
```

- **Initial data population** for testing.

4. Creating Database Links

At TIFFY:

```
CREATE DATABASE LINK TIFFYTORUNNY
CONNECT TO RUNNY IDENTIFIED BY RUNNY USING 'ORCL';

CREATE DATABASE LINK TIFFYTOKITTY
CONNECT TO KITTY IDENTIFIED BY KITTY USING 'ORCL';
```

At RUNNY:

```
CREATE DATABASE LINK KITTYTOTIFFY
CONNECT TO TIFFY IDENTIFIED BY TIFFY USING 'ORCL';
```

At KITTY:

```
CREATE DATABASE LINK RUNNYTOTIFFY
CONNECT TO TIFFY IDENTIFIED BY TIFFY USING 'ORCL';
```

- **Database links enable cross-database queries and updates.**

5. Creating Triggers for Data Synchronization

At TIFFY:

```
CREATE OR REPLACE TRIGGER AVAILABILITY
AFTER INSERT ON ALLOTMENT
FOR EACH ROW
BEGIN
    UPDATE COLL_BRANCH SET NOS = NOS - 1
    WHERE COLL_CODE = :NEW.COLL_CODE AND BCODE = :NEW.BCODE;
    UPDATE COLL_BRANCH@TIFFYTORUNNY SET NOS = NOS - 1
    WHERE COLL_CODE = :NEW.COLL_CODE AND BCODE = :NEW.BCODE;

    UPDATE COLL_BRANCH@TIFFYTOKITTY SET NOS = NOS - 1
    WHERE COLL_CODE = :NEW.COLL_CODE AND BCODE = :NEW.BCODE;
END;
```

At RUNNY:

```
CREATE OR REPLACE TRIGGER AVAILABILITY
AFTER INSERT ON ALLOTMENT
FOR EACH ROW
BEGIN
    INSERT INTO ALLOTMENT@RUNNYTOTIFFY VALUES
(:NEW.ROLLNO, :NEW.COLL_CODE, :NEW.BCODE);
END;
```

At KITTY:

```
CREATE OR REPLACE TRIGGER AVAILABILITY
AFTER INSERT ON ALLOTMENT
FOR EACH ROW
BEGIN
    INSERT INTO ALLOTMENT@KITTYTOTIFFY VALUES
(:NEW.ROLLNO, :NEW.COLL_CODE, :NEW.BCODE);
END;
```

- **Triggers ensure consistency of seat availability and ALLOTMENT updates across all databases.**

6. Testing the Distributed System

At TIFFY:

```
INSERT INTO ALLOTMENT VALUES(12345, 1, 'CSE');
COMMIT;
SELECT * FROM COLL_BRANCH;
SELECT * FROM ALLOTMENT;
```

At RUNNY:

```
INSERT INTO ALLOTMENT VALUES(12344, 1, 'CSE');
COMMIT;
SELECT * FROM COLL_BRANCH;
SELECT * FROM ALLOTMENT;
```

At KITTY:

```
INSERT INTO ALLOTMENT VALUES(12355, 1, 'CSE');
COMMIT;
SELECT * FROM COLL_BRANCH;
SELECT * FROM ALLOTMENT;
```

- **Each insert triggers updates across all connected databases.**
- **Cross-database consistency is maintained through triggers and database links.**

Conclusion

- **TIFFY stores the master data** (STUDENT, COLLEGE, BRANCH, COLL_BRANCH).
- **ALLOTMENT is replicated** at **RUNNY and KITTY** to support distributed transactions.
- **Database links and triggers ensure data synchronization** across all three sites.
- **After each ALLOTMENT entry, available seats are updated automatically** at all locations.

This setup represents a **distributed database management system (DDBMS) in Oracle** using **database links and triggers** for seamless data replication and synchronization.

Since this is a **distributed database setup** with multiple users (**TIFFY, RUNNY, KITTY**), you need to **open three SQL*Plus sessions** (or three separate terminals) and execute the corresponding commands in each session.

How to Set Up and Execute the Code?

1. **Open three SQL*Plus sessions:**
    a. One for **TIFFY**
    b. One for **RUNNY**
    c. One for **KITTY**
2. **At TIFFY:** (Execute in TIFFY's session)
    a. Create all tables (STUDENT, COLLEGE, BRANCH, COLL_BRANCH, ALLOTMENT)
    b. Create database links to RUNNY and KITTY
    c. Create triggers for updating COLL_BRANCH seats
    d. Insert data into tables

```
-- Connect as TIFFY
CONN C##TIFFY/TIFFY@ORCL;

-- Create tables at TIFFY
CREATE TABLE STUDENT (
  HSCNO NUMBER(5) PRIMARY KEY,
  NAME VARCHAR(30),
  DOB DATE,
  GENDER CHAR CHECK(GENDER IN('M', 'F')),
  AGGR NUMBER(3)
);
```

```
CREATE TABLE COLLEGE (
  COLL_CODE NUMBER(3) PRIMARY KEY,
  NAME VARCHAR(50)
);

CREATE TABLE BRANCH (
  BCODE VARCHAR(3) PRIMARY KEY,
  BNAME VARCHAR(50)
);

CREATE TABLE COLL_BRANCH (
  COLL_CODE NUMBER(3) REFERENCES COLLEGE,
  BCODE VARCHAR(3) REFERENCES BRANCH,
  NOS NUMBER(3),
  PRIMARY KEY(COLL_CODE, BCODE)
);

CREATE TABLE ALLOTMENT (
  ROLLNO NUMBER(5) PRIMARY KEY REFERENCES STUDENT(HSCNO),
  COLL_CODE NUMBER(3),
  BCODE VARCHAR(3),
  FOREIGN KEY (COLL_CODE, BCODE) REFERENCES COLL_BRANCH
);

-- Create DB Links at TIFFY
CREATE DATABASE LINK TIFFYTORUNNY
CONNECT TO RUNNY IDENTIFIED BY RUNNY USING 'ORCL';

CREATE DATABASE LINK TIFFYTOKITTY
CONNECT TO KITTY IDENTIFIED BY KITTY USING 'ORCL';

-- Create Trigger at TIFFY
CREATE OR REPLACE TRIGGER AVAILABILITY
AFTER INSERT ON ALLOTMENT
FOR EACH ROW
BEGIN
  UPDATE COLL_BRANCH SET NOS=NOS-1
  WHERE COLL_CODE=:NEW.COLL_CODE AND BCODE=:NEW.BCODE;

  UPDATE COLL_BRANCH@TIFFYTORUNNY SET NOS=NOS-1
  WHERE COLL_CODE=:NEW.COLL_CODE AND BCODE=:NEW.BCODE;

  UPDATE COLL_BRANCH@TIFFYTOKITTY SET NOS=NOS-1
  WHERE COLL_CODE=:NEW.COLL_CODE AND BCODE=:NEW.BCODE;
END;
```

```
/

COMMIT;
```

3. **At RUNNY:** (Execute in RUNNY's session)
    a. Create a database link to TIFFY
    b. Create a trigger to sync ALLOTMENT data with TIFFY

```
-- Connect as RUNNY
CONN C##RUNNY/RUNNY@ORCL;

-- Create DB Link at RUNNY
CREATE DATABASE LINK RUNNYTOTIFFY
CONNECT TO TIFFY IDENTIFIED BY TIFFY USING 'ORCL';

-- Create Trigger at RUNNY
CREATE OR REPLACE TRIGGER AVAILABILITY
AFTER INSERT ON ALLOTMENT
FOR EACH ROW
BEGIN
  INSERT INTO ALLOTMENT@RUNNYTOTIFFY VALUES
  (:NEW.ROLLNO, :NEW.COLL_CODE, :NEW.BCODE);
END;
/

COMMIT;
```

4. **At KITTY:** (Execute in KITTY's session)
    a. Create a database link to TIFFY
    b. Create a trigger to sync ALLOTMENT data with TIFFY

```
-- Connect as KITTY
CONN C##KITTY/KITTY@ORCL;

-- Create DB Link at KITTY
CREATE DATABASE LINK KITTYTOTIFFY
CONNECT TO TIFFY IDENTIFIED BY TIFFY USING 'ORCL';

-- Create Trigger at KITTY
CREATE OR REPLACE TRIGGER AVAILABILITY
AFTER INSERT ON ALLOTMENT
FOR EACH ROW
BEGIN
```

```
  INSERT INTO ALLOTMENT@KITTYTOTIFFY VALUES
  (:NEW.ROLLNO, :NEW.COLL_CODE, :NEW.BCODE);
END;
/

COMMIT;
```