Multi Insert

This document describes the Tibero Multi Insert scenario.

Setting a test environment

```
SQL> DROP TABLE T1;

SQL> DROP TABLE T2;

SQL> CREATE TABLE T1 (A NUMBER, B NUMBER);

SQL> CREATE TABLE T2 (A NUMBER, B NUMBER);
```

1. Unconditional INSERT ALL senario

```
-- Enter 2 items
SQL> INSERT INTO T1 (A, B) VALUES (1,2), (3,4);

-- Enter 2 items

SQL> INSERT INTO T1 VALUES(1, 2), (3, 4);

-- Enter 2 items

SQL> INSERT ALL

   INTO T1 (A, B) VALUES (1,2)

   INTO T2 (A, B) VALUES (3,4)

SELECT * FROM DUAL;

-- Enter 2 items

SQL> INSERT ALL

   INTO T1 VALUES (1,2)

   INTO T2 VALUES (3,4)

SELECT * FROM DUAL;

-- Enter 2 items

SQL> INSERT ALL

          INTO T1 (A, B)
```

```
        INTO T2 (A, B)
SELECT 1, 2 FROM DUAL;
-- Enter 2 items
SQL> INSERT ALL
        INTO T1
        INTO T2
SELECT 1, 2 FROM DUAL;
```

2. Conditional INSERT ALL

The SELECT result was divided into two rows using UNION ALL, and the INSERT result was entered into each row.

```
--Input 2
SQL> INSERT ALL
        WHEN A = 1 THEN
                INTO T1
        ELSE
                INTO T2
SELECT 1 A, 2 FROM DUAL UNION ALL
SELECT 3 A, 4 FROM DUAL;
--Input 2
SQL> INSERT ALL
        WHEN A = 1 THEN
                INTO T1
        ELSE
                INTO T2
```

```
SELECT 1 A, 2 FROM DUAL UNION ALL
SELECT 3 A, 4 FROM DUAL;
```

3. Conditional INSERT FIRST

```
-- Enter 2 items
SQL> INSERT FIRST
        WHEN A = 1 THEN
                INTO T1 (A, B)
        ELSE
                INTO T2 (A, B)
SELECT 1 A, 2 FROM DUAL UNION ALL
SELECT 3 A, 4 FROM DUAL;
-- Enter 2 items
SQL> INSERT FIRST
        WHEN A = 1 THEN
                INTO T1
        ELSE
                INTO T2
SELECT 1 A, 2 FROM DUAL UNION ALL
SELECT 3 A, 4 FROM DUAL;
```

4. Sequence senario

```
SQL> CREATE SEQUENCE SEQ START WITH 1 INCREMENT BY 1;

-- enter the data

SQL> INSERT ALL

        INTO T1 VALUES(1, SEQ.NEXTVAL)

        INTO T2 VALUES(2, SEQ.NEXTVAL)

        INTO T1 VALUES(3, SEQ.NEXTVAL)

        INTO T2 VALUES(4, SEQ.NEXTVAL)

        INTO T1 VALUES(5, SEQ.NEXTVAL)

        INTO T2 VALUES(6, SEQ.NEXTVAL)

SELECT * FROM DUAL;

--  When checking the data, all sequences were inserted as 1.

SQL> SELECT * FROM T1;

SQL> SELECT * FROM T2;
```

Performance comparison scenario

As the number of data items increases, the performance benefits of Multi-Insert can be observed.

1. Comparison of the insertion times for 1 million items in T1 (1,2) and 1 million items in T2 (3,4)

```
SQL>SET timing ON

-- Just INSERT takes 43 seconds

SQL> DECLARE

        I NUMBER :=1;

BEGIN
```

```
   LOOP
   INSERT INTO T1 (A, B) VALUES (1,2);
    INSERT INTO T2 (A, B) VALUES (3,4);
   I := I+1;
   EXIT WHEN I >1000000;
   END LOOP;
END;
/
PSM completed.
Total elapsed time 00:00:43.249616
-- Multi Insert takes 39 seconds
SQL> DECLARE
          I NUMBER :=1;
BEGIN
   LOOP
   INSERT ALL
                    INTO T1 (A, B) VALUES (1,2)
                    INTO T2 (A, B) VALUES (3,4)
          SELECT * FROM DUAL;
   I := I+1;
   EXIT WHEN I >1000000;
   END LOOP;
END;
/
PSM completed.
Total elapsed time 00:00:39.223399
```