

# Multi Insert

본 문서는 티베로 **Multi Insert** 시나리오에 대해 기술합니다.

## 테스트 환경 구축

```
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 시나리오

```
-- 2 건 입력

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

-- 2 건 입력

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

-- 2 건 입력

SQL> INSERT ALL

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

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

SELECT * FROM DUAL;

-- 2 건 입력

SQL> INSERT ALL

    INTO T1 VALUES (1,2)

    INTO T2 VALUES (3,4)

SELECT * FROM DUAL;

-- 2 건 입력

SQL> INSERT ALL

    INTO T1 (A, B)
```

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

## 2. Conditional INSERT ALL

UNION ALL 로 SELECT 결과를 2 행으로 만들어 INSERT 결과가 각각 들어가도록 하였다.

```
--2 건 입력
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;
-- 2 건 입력
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

```
-- 2 건 입력  
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;  
-- 2 건 입력  
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 시나리오

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

```

-- 데이터 입력
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;
-- 데이터 확인시 sequence 는 모두 1 로 삽입되었다.
SQL> SELECT * FROM T1;
SQL> SELECT * FROM T2;

```

## 성능 비교 시나리오

데이터 건수가 많아질수록 Multi Insert 시 성능상 이점을 확인할 수 있다.

1. T1(1,2)에 100 만건, T2(3,4)에 100 만건 입력 시간 비교

```

SQL>SET timing ON
-- 그냥 INSERT 시 43 초 소요
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 시 39 초 소요
```

```
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
```