

---

# Goodus 기술노트 [44 회]

## SQL Loader

<b>Author</b>	나지혜, 이규열
<b>Creation Date</b>	2009-09-8
<b>Last Updated</b>	2009-09-8
<b>Version</b>	1.0
Copyright(C) 2004 Goodus Inc. All Rights Reserved	

Version	변경일자	변경자(작성자)	주요내용
1	2009-09-8	나지혜, 이규열	문서 최초 작성
2			
3			

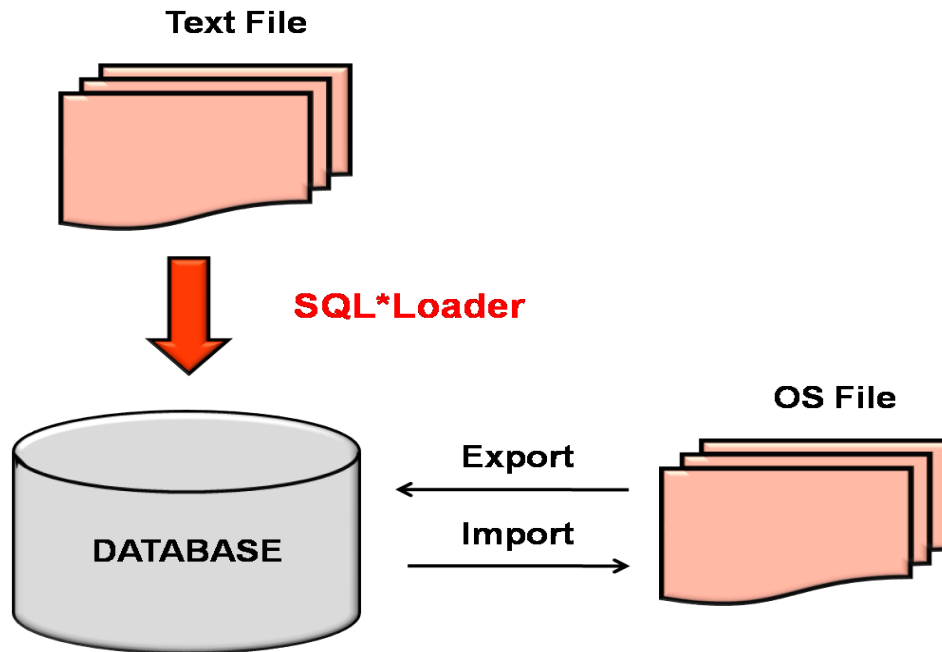
---

# Contents

1. SQL LOADER 란? .....	3
2. SQL LOADER 에 사용되는 파일 .....	4
2.1. 컨트롤파일(Control File).....	4
2.2. 데이터 파일(Data file) .....	6
2.3. 로그 파일 (Log File).....	7
2.4. 배드 파일 (Bad File) .....	7
2.5. 디스카드 파일(Discard file).....	8
3. SQL LOADER 로 데이터 로드 방법.....	8
3.1. 기본 경로 로드 방법 (Conventional path load).....	8
3.2. 직접 경로 로드 방법 (Direct patch load) .....	9
3.3. 병렬 직접 경로 로드 방법 (Paraller direct path load) .....	9
4. SQL LOADER 실습.....	10
4.1. 기본 경로 로드 방식 .....	10
4.2. 직접 경로 로드 방식 .....	11
5. ORANGE 를 이용한 Loader 쉽게 따라하기 .....	12
6. 부록 .....	17
6.1. SQL LOADER 성능 향상 기법 .....	17
6.2. 키워드 설명 .....	17
6.3. 참조문헌 .....	18

---

# 1. SQL LOADER란?



기존의 응용 프로그램 데이터나 다른 데이터베이스로 저장된 데이터를 오라클 데이터베이스 테이블에 넣기 위한 유틸리티로서 IBM의 DB2 load 유틸리티와 흡사하다.

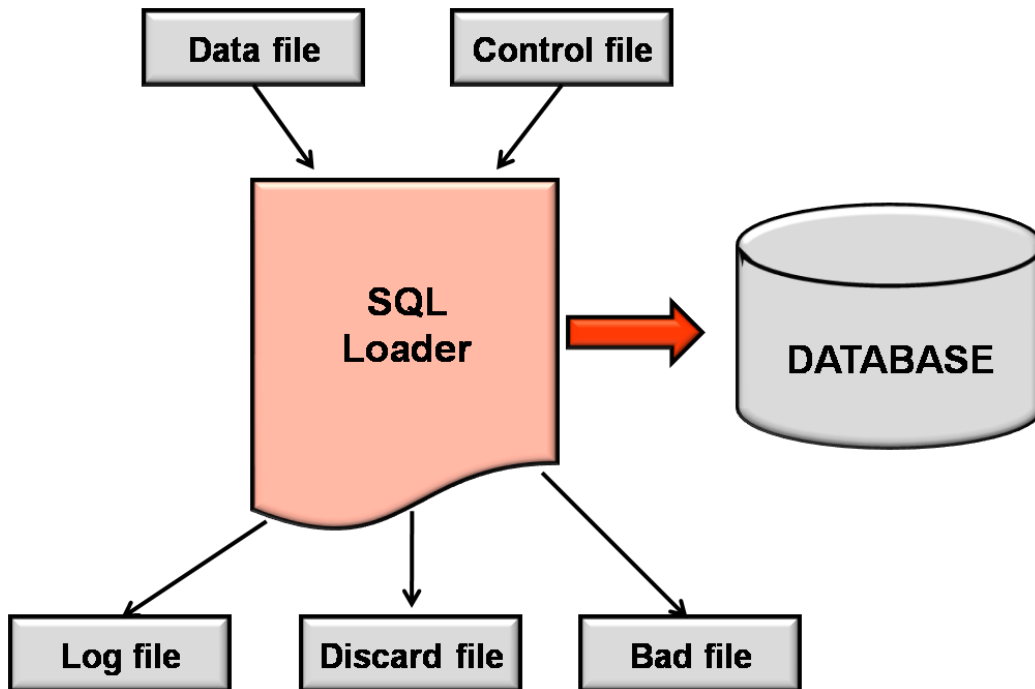
오라클 데이터베이스를 설치하면 기본적으로 설치되며 간단하고 편리하게 데이터를 데이터베이스에 로드할 수 있다.

## 1.1 SQL LOADER의 특징

- SQL Loader는 하나 이상의 입력 파일을 사용할 수 있다.
- 로드할 때 여러 개의 입력 레코드를 하나의 논리적 레코드로 결합할 수 있다.
- 입력 필드는 고정 길이 또는 가변 길이가 가능하다.
- 문자, 이진, 팩형 십진 형식(packed decimal format), 날짜 및 존 십진 형식(zoned decimal format)과 같은 임의의 형식이 입력 데이터가 될 수 있다.
- 데이터를 디스크, 테이프 또는 명명된 파이프와 같은 다양한 매체 유형에서 로드할 수 있다.
- 데이터를 한번 실행하여 여러 테이블로 로드할 수 있다.
- 테이블의 기존 데이터를 바꾸거나 추가하기 위해 옵션을 사용할 수 있다.
- 데이터 베이스 행을 저장하기 전에 SQL 함수를 입력 데이터에 적용할 수 있다.
- 규칙에 따라 열 값을 자동으로 생성할 수 있다. 예를 들어, 순차 키 값을 열에 생성하고 저장할 수 있다.

---

## 2. SQL LOADER에 사용되는 파일



SQL LOADER 는 5 개의 (control file, data file, log file, bad file, discard file)을 사용한다.

### 2.1. 컨트롤파일(Control File)

Control file 은 SQL Loader 를 사용하는데 필수 적인 파일의 하나로써 데이터 정의어(DDL) 지 침을 포함하는 텍스트 파일이다. 확장자는 'ctl'이다.

#### 2.1.1 기능

- SQL Loader 가 로드할 데이터를 찾을 위치 명시
- SQL Loader 가 형식이 지정될 데이터를 예상하는 방법
- 데이터를 로드하는 동안 SQL Loader 가 메모리 관리, 레코드 거부, 중단된 로드 처리 등을 구성하는 방법
- SQL Loader 가 로드 중인 데이터를 조작하는 방법

#### 2.1.2 컨트롤파일 작성시 고려사항

- 구문은 자유로운 형식이므로 여러 행으로 확장 될 수 있다.
- 대소문자를 구분하지 않는다.
- 작은 따옴표나 큰 따옴표로 묶인 문자열은 대소문자를 포함하여 문자 그대로 사용한다.
- 제어 파일 구문에서 주석은 주석의 시작을 나타내는 두 개의 하이픈(-- )으로 표시한다.

#### 2.1.3 컨트롤파일 형식

- 
- 1 -- This is a sample control file → 주석의 입력
  - 2 LOAD DATA → 새 데이터 로드가 시작됨을 의미  
진행 중에 중단된 로드를 계속할 경우 CONTINUE LOAD DATA 문을 사용
  - 3 INFILE 'GOODUS.DAT' → 외부 데이터 파일 지정, 포함하지 않을 경우 '\*'로 표시
  - 4 BADFILE 'goodus.bad' → 거부된 레코드를 배치할 파일 이름을 지정
  - 5 DISCARDFILE 'sample.dsc' → 폐기된 레코드를 배치할 파일 이름을 지정
  - 6 REPLACE → 테이블에 데이터를 삽입하는 방법 지정
    - REPLACE : 테이블의 기존 행을 모두 삭제 하고 삽입
    - APPEND : 새로운 행을 기존의 데이터에 추가
    - INSERT : 비어 있는 테이블에 넣을 때 사용
    - TRUNCATE : 테이블의 기존 데이터를 모두 삭제하고 삽입
  - 7 INTO TABLE test → 데이터를 로드 할 테이블 지정
  - 8 WHEN (10) = '.' → 데이터를 로드하기 전에 만족시켜야 할 각 레코드의 필드 조건을 지정  
이 예제는 10 번째 문자가 소수점인 경우에만 레코드를 삽입한다
  - 9 FIELDS TERMINATED BY ',' → 데이터 필드의 종결 문자를 지정

#### 2.1.4 컨트롤파일 작성 예제

컨트롤 파일에서 데이터를 포함하지 않은 경우 INFILE 다음에 파일을 지정하고 BEGINDATA 부터 생략하면 된다.

```

- 데이터를 포함하는 경우
LOAD DATA
INFILE *
REPLACE
INTO TABLE test
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(NUMBER, NAME, PHONE)
BEGINDATA
1, "이규열", "011-9401-0001"
2, "장동건", "010-777-7777"
3, "신민아", "010-555-4744"

- 데이터를 포함하지 않는 경우
LOAD DATA
INFILE '/GOODUS/sql_loader.dat'
REPLACE
INTO TABLE test
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(NUMBER, NAME, PHONE)

```

---

## 2.2. 데이터 파일(Data file)

SQL Loader 로부터 데이터베이스에 입력될 텍스트 형식으로 구성된 데이터 파일이다.

### 2.2.1 특징

- SQL Loader 는 제어파일에 지정된 하나 이상의 파일에서 데이터를 읽는다
- SQL Loader 의 관점에서 데이터 파일의 데이터는 레코드로 구성
- 컨트롤 파일의 INFILE 매개변수에 레코드 형식을 명시하며, 그렇지 않으면 스트림 형식으로 인식한다.

### 2.2.2 종류

#### ▶ 고정 레코드 형식(fixed record format)

- 데이터 파일의 모든 레코드가 동일한 바이트 일 경우를 의미
- 가장 융통성이 적지만 가변 형식이나 스트림 형식보다 더 나은 성능을 제공
- 고정 레코드 형식 : INFILE 데이터 파일이름 "fix\_n"

ex)

모든 길이를 바이트로 해석한다.  
데이터 파일은 3개의 레코드로 이루어져 있다.  
빈칸은 공백을 의미한다.

컨트롤 파일 :

```
load data
infile 'example.dat' "fix 16" →16byte임을 명시
into table example
fields terminated by ',' optionally enclosed by ''
(name,message)
```

데이터 파일 :

```
lee, hi goodus
goodus,      hi
lee,goodus lock
```

#### ▶ 가변 레코드 형식(variable record format)

- 각 레코드의 길이가 데이터 파일의 각 레코드 시작부분에 포함될 경우
- 고정 레코드 형식보다 더 나은 융통성을 제공, 스트림 레코드 형식보다 더 나은 성능상 이점을 제공
- 가변 레코드 형식 : INFILE 데이터 파일이름 "var n"
- n 이 지정되지 않으면 기본값 5 로 가정함
- n 을 40 보다 큰 수로 지정하면 오류발생

ex)

컨트롤 파일 :

```
load data
infile 'goodus.dat' "var 3" → 3개의 필드로 구성되어 있음 명시
into table test
```

```

fields terminated by ',' optionally enclosed by '"'
(col1 char(5),col2 char(7))
데이터 파일 :
003hi,
011my name is
007goodus

```

▶ 스트림 레코드 형식(Stream record format)

- 레코드의 크기가 지정되지 않은 경우
- 융통성은 가장 뛰어 나지만 성능은 떨어짐
- 스트림 레코드 형식 : INFILE 데이터파일이름 "문자열 terminator\_string"
- terminator\_string 은 'char\_string' 또는 X'hex\_string'중 하나로 지정됨  
 (char\_string : " "로 둘러싸인 문자열 의미  
 Hexadecimal\_string : 16 진수 형식의 바이트 문자열 의미)

```

ex) '|'를 기준으로 레코드를 읽어들이는 예제
컨트롤 파일 :
load data
infile 'goodus.dat' "str '|'Wn"
into table example
fields terminated by ',' optionally enclosed by '"'
(col1 char(5),
col2 char(7))
데이터 파일 :
hello,goodus,|
hello,di기술팀,|

```

### 2.3. 로그 파일 (Log File)

로그 파일은 데이터 로드 상태 정보, 로드 된 행의 수, 로드 진행중 데이터 에러 때문에 거부된 행의 수, 버려진 행의 수 등 자세한 테이블 로드 정보가 포함된다.  
 즉, 로그 파일은 읽어 들인 레코드의 수, 데이터 로드 시간, 입력되지 않은 레코드의 수와 같은 정보를 포함하고 있다. 기본적으로 컨트롤 파일과 동일한 이름으로 생성 된다.

### 2.4. 배드 파일 (Bad File)

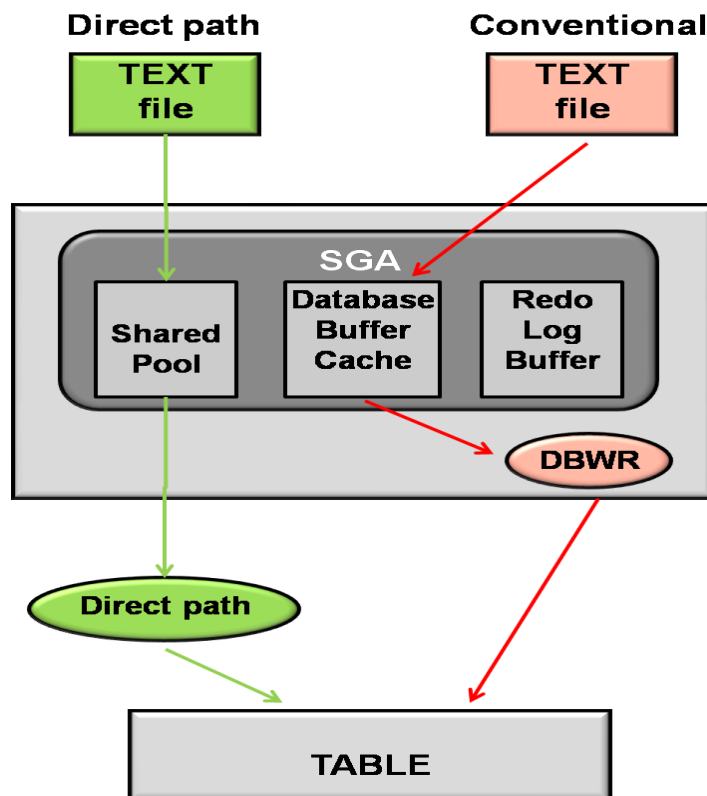
BAD 파일은 SQL\*Loader 또는 오라클 데이터베이스가 거부한 레코드를 포함한다.  
 이 파일의 포맷은 물론 원래의 데이터 파일과 동일하게 되며, 이러한 Bad File 은 이 파일내의 잘못된 부분을 수정한 뒤에 다시 데이터 파일로 사용할 수 있으며 필요 할 경우에만 생성이 된다.

- 문법구조 : 컨트롤 파일 내에서 badfile '파일명.bad'
- 배드 파일에 기록되는 경우
  - : 입력 형식이 부적합할 경우 (ex : 구분자 누락, 필드의 최대 길이 초과)
  - : 제약조건을 위배 하였을 때  
 (ex : 고유하지 않은 키 값, 필수 필드가 null, oracle 데이터 유형에 부적합한 데이터가 포함)

## 2.5. 디스카드 파일(Discard file)

- SQL\*Loader 가 실행되는 동안 폐기 파일이라는 파일을 생성할 수 있다.
- 이 파일은 필요한 경우에만 생성 된다.
- 컨트롤 파일에 명시한 WHEN 절의 조건에 맞지 않는 데이터를 저장하는 파일
- 배드 파일처럼 입력형식으로 저장되므로 오류를 수정하여 재사용 할 수 있다.
- 문법구조 : 컨트롤 파일 내에서 discardfile '파일명.dsc'  
명령행에서 discard(파일명)

## 3. SQL LOADER로 데이터 로드 방법



SQL\*Loader 를 사용하여 LOAD 하는 방식에는 기본경로(conventional), 직접경로(direct path), 병렬 직접 경로(paraller direct path) 로드방식이 있다.

### 3.1. 기본 경로 로드 방법 (Conventional path load)

- 삽입할 행 배열을 생성하고 SGA 메모리 영역에 저장한 다음 SQL INSERT 문을 사용하여 데이터를 로드 한다.
- 기본 경로 로드 동안에는 필드 사양을 기준으로 입력 레코드의 구문이 분석되고 레코드 배열은 제어 파일에서 지정한 테이블에 생성 및 삽입된다.
- 필드 사양을 준수하지 않는 레코드는 거부되고 선택 기준에 맞지 않는 이러한 레코드는 폐기 된다.



- 기본 경로 로드를 사용하여 데이터를 클러스터화된 테이블 및 클러스터화 되지 않는 테이블 모두에 로드 할 수 있다.
- 로드 되는 테이블에 대한 로그 기록 속성은 리두 생성을 제어한다.

### 3.2. 직접 경로 로드 방법 (Direct patch load)

- 직접 경로 로드는 데이터 블록을 메모리에 생성하고 이 블록을 로드 되는 테이블에 대해 할당된 확장 영역에 직접 저장한다.
- 데이터 베이스가 ARCHIVELOG MODE 가 아니면 온라인 리두 로그 항목이 생성되지 않는다
- 필드 사양을 사용하여 전체 데이터 블록을 생성하고 이 블록을 데이터 파일에 직접 쓴다.
- 확장 영역 관리 및 고수위 조정을 위해 데이터베이스 버퍼 캐시를 통과하고 SGA 를 액세스 한다.
- 직접 경로 로드는 대개 기본 경로 로드보다 빠르지만 모든 상황에서 사용 할 수 는 없다.
- 다음의 표에서 기본 경로 로드와 직접 경로 로드를 비교하여 각 로드 방식을 사용할 수 있다.

일반경로	직접경로
Commit 문을 사용하여 영구적으로 변경	DATA SAVE 사용
항상 리두 항목 생성	특정 조건에서만 리두 생성
모든 제약조건 적용	기본키, 유일키, not null 제약만 적용
클러스터화된 테이블로 로드 할 수 있음	클러스터화된 테이블로 로드 불가능
INSERT 트리거 실행	로더중에 INSERT 트리거 사용 불가능
다른 사용자가 테이블을 변경할 수 있음	다른 사용자가 로드되는 테이블 사용할 수 없음

### 3.3. 병렬 직접 경로 로드 방법 (Paraller direct path load)

- PARALLEL DIRECT LOAD 는 LOAD 할 때 여러 개의 LOAD SESSION 이 한 TABLE 로 데이터를 LOAD 하여 속도가 더 증가 한다.
- 각각의 SESSION 은 서로 다른 DATA log 를 사용해야 한다. 따라서 CONTROL log 파일 이름도 달라야 한다.
- PARALLEL 옵션을 사용할 때 SQL LOADER 는 TEMPORARY SEGMENT 에 DATA 를 LOAD 하여 LOAD 가 끝나는 시점에서 TEMPORARY SEGMENT 를 하나로 묶고 이것을 TABLE SEGMENT 로 추가하는 것이다.
- LOAD 시에 INDEX 가 걸려 있다면 DISABLE 한 후 작업이 끝나면 ENABLE 해야 한다.
- 컨트롤 파일 명시절의 APPEND mode 에서만 가능하다. 즉, INSERT, REPLACE, TRUNCATE mode 에서는 지원되지 않는다.

## 4. SQL LOADER 실습

### 4.1. 기본 경로 로드 방식

- 컨트롤 파일에 데이터까지 함께 포함되어 있는 경우

SQL> create table test -> 로드할 테이블 생성

```
(id number(38),
 name varchar(20),
 phone varchar(20));
```

Table created.

[starlee:ORA10:/oracle/sqlldr]vi test.ctl -> 컨트롤 파일 작성

```
load data
infile *
append into table test
fields terminated by ','
(id,name,phone)
```

begindata

```
1,이규열,011-9401-2222
2,장동건,011-9823-1231
3,원 빈,018-555-4744
4,정우성,010-9876-1234
5,전지현,019-325-9876
6,신민아,010-2345-9123
```

[starlee:ORA10:/oracle/sqlldr]ls -alrt

합계 12

```
-rw-r--r-- 1 oracle dba 231 9월 2 15:56 test.ctl
```

```
drwxr-xr-x 9 oracle dba 4096 9월 2 15:56 ..
```

```
drwxr-xr-x 2 oracle dba 4096 9월 2 15:56 .
```

[starlee:ORA10:/oracle/sqlldr]sqlldr userid=lee/lee control=test.ctl -> sql loader 실행

SQL\*Loader: Release 10.2.0.4.0 - Production on Wed Sep 2 15:57:09 2009

Copyright (c) 1982, 2007, Oracle. All rights reserved.

Commit point reached - logical record count 7

[starlee:ORA10:/oracle/sqlldr]

test.bad test.ctl test.log -> badfile과 logfile은 지정안해도 controlfile 명과 함께 생성됨

SQL> select \* from test; -> 로드된 내용 확인

ID	NAME	PHONE
1	이규열	011-9401-2222
2	장동건	011-9823-1231
3	원 빈	018-555-4744
4	정우성	010-9876-1234
5	전지현	019-325-9876
6	신민아	010-2345-9123

6 rows selected.

- log file을 열어보면 Path used : Conventional 인것을 확인 할 수 있다.

## 4.2. 직접 경로 로드 방식

- 컨트롤 파일과 데이터 파일이 분리되어 있는 경우

[starlee:ORA10:/oracle/sqlldr]vi goodus.ctl **-> 컨트롤 파일 작성**

```
load data
infile * '/oracle/sqlldr/goodus.dat'
append into table test ->append mode적용
fields terminated by ','
(id,name,phone)
```

[starlee:ORA10:/oracle/sqlldr]vi goodus.dat **-> 데이터 파일 작성**

```
10,호나우딩요, 브라질phone
20,리오넬 메시, 아르헨티나phone
30,사비올라,아르헨티나phone
40,아드리아누,브라질phone
50,크리스티아누 호날두, 포르투갈phone
```

[starlee:ORA10:/oracle/sqlldr]sqlldr userid=lee/lee control=goodus.ctl **direct=true ->sqlldr 실행**

SQL\*Loader: Release 10.2.0.4.0 - Production on Wed Sep 2 19:54:38 2009

Copyright (c) 1982, 2007, Oracle. All rights reserved.

Load completed - logical record count 5.

SQL> select \* from test; **-> 기본경로 로드 방식 다음에 로드 된 것을 확인할 수 있음**

ID	NAME	PHONE
1	이규열	011-9401-2222
2	장동건	011-9823-1231
3	원 빈	018-555-4744
4	정우성	010-9876-1234
5	전지현	019-325-9876
6	신민아	010-2345-9123
10	호나우딩요	브라질phone
20	리오넬 메시	아르헨티나phone
30	사비올라	아르헨티나phone
40	아드리아누	브라질phone
50	크리스티아누 호날두	포르투갈phone

11 rows selected.

- log file을 열어보면 **Path used : Direct** 인것을 확인 할 수 있다.

## 5. ORANGE를 이용한 Loader 쉽게 따라하기

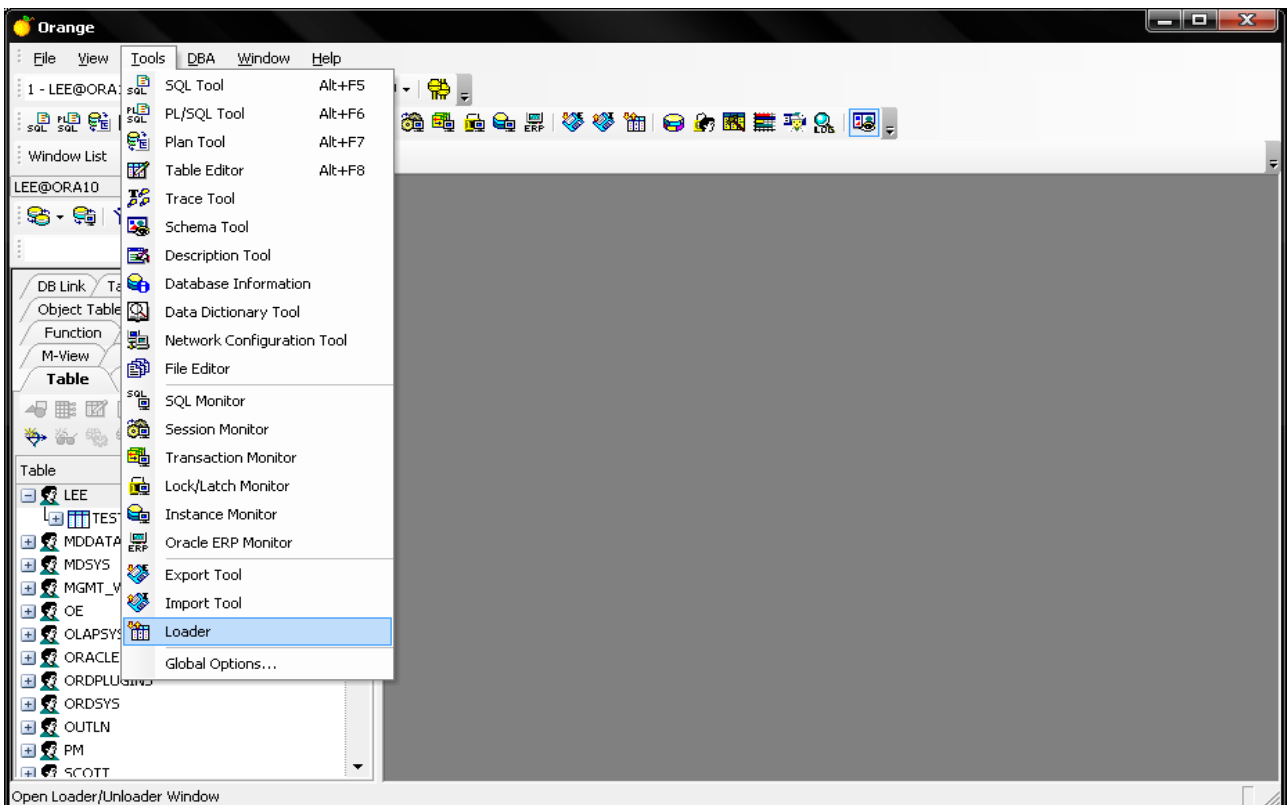
DBA 나 개발자들이 흔히 사용하는 TOOL 에는 ORANGE 나 TOAD 같은 것들이 있다. 이 같은 TOOL 을 사용하면 보다 더 쉽게 LOADER 를 다루는 것이 가능하다.

GUI 환경의 TOOL 을 이용함으로써 LOADER 의 기능을 자유자재로 활용가능하고 쉽게 접근이 가능하며 결과값을 엑셀 파일이나 텍스트 파일 등 다양한 포맷으로 변환해 필터등을 이용하여 쉽게 자신이 원하는 값을 추출 할 수 있다.

여기에서는 국산 TOOL 인 ORANGE 를 이용하여 쉽게 loader 하는 법에 대해서 알아 보겠다.

### 1. ORANGE 시작후 Loader 시작

- Loader 할 유저로 접속한 다음 TOOL -> Loader 를 클릭한다. 클릭하면 다시 한번 Logging 창 이 나오는 데 다시 한번 접속을 해준다.



## 2. Load 할 datafile 작성

- 이번 예제에서는 업무에서 흔히 데이터 입력용으로 작성하는 엑셀의 csv 파일(쉼표로 분리)의 예제를 들어 보겠다.(해당유저는 Loader 전에 미리 테이블 정의가 되어져 있어야 한다.)

```
SQL> create table fifa_ranking(  
2 순위 varchar2(100),  
3 국가 varchar2(100),  
4 포인트 varchar2(100),  
5 순위변동 varchar2(100));
```

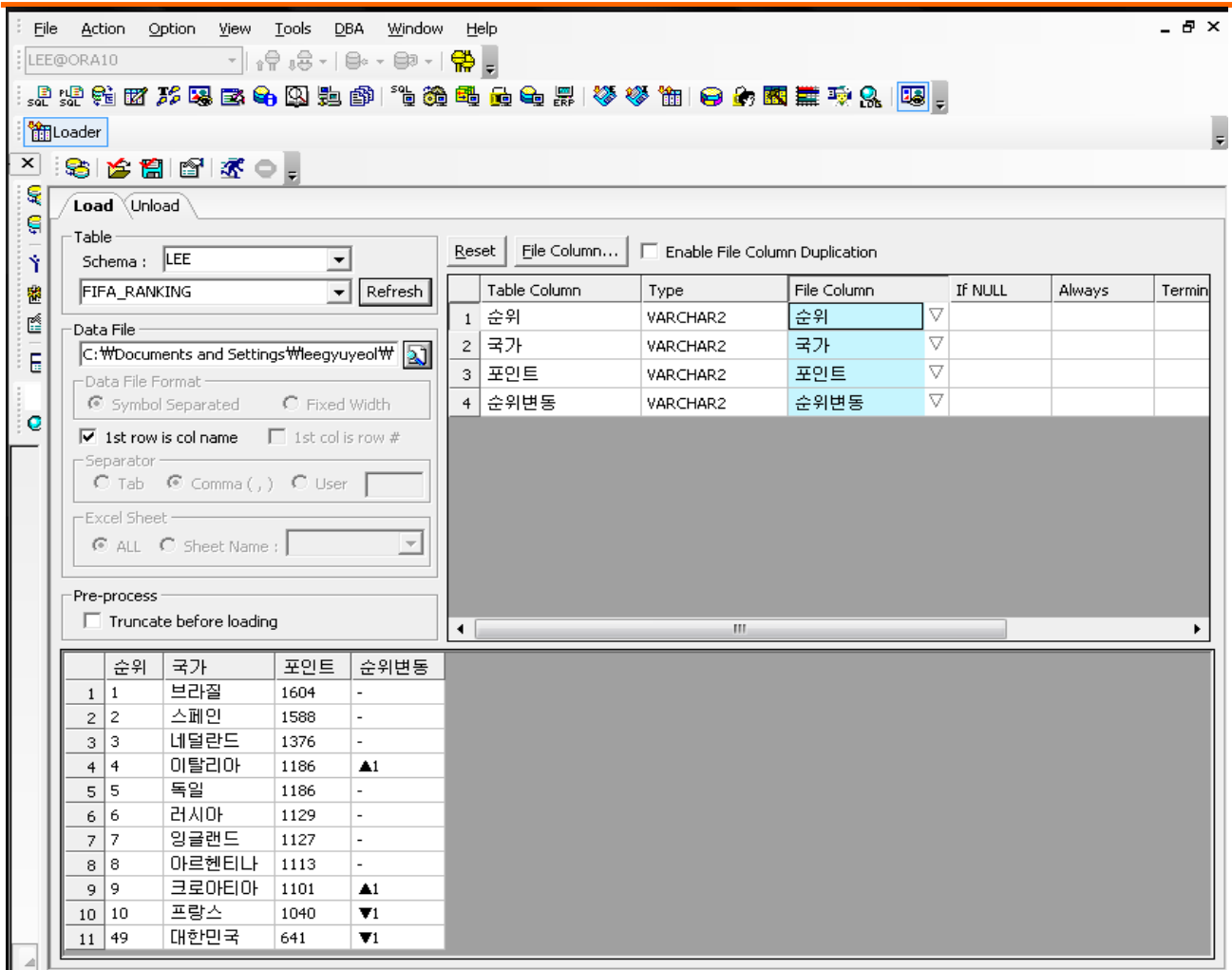
Table created.

### - Data file 작성

	A	B	C	D	E	F	G	H	I	J
1	순위	국가	포인트	순위변동						
2	1	브라질	1604	-						
3	2	스페인	1588	-						
4	3	네덜란드	1376	-						
5	4	이탈리아	1186	▲1						
6	5	독일	1186	-						
7	6	러시아	1129	-						
8	7	잉글랜드	1127	-						
9	8	아르헨티나	1113	-						
10	9	크로아티아	1101	▲1						
11	10	프랑스	1040	▼1						
12	49	대한민국	641	▼1						

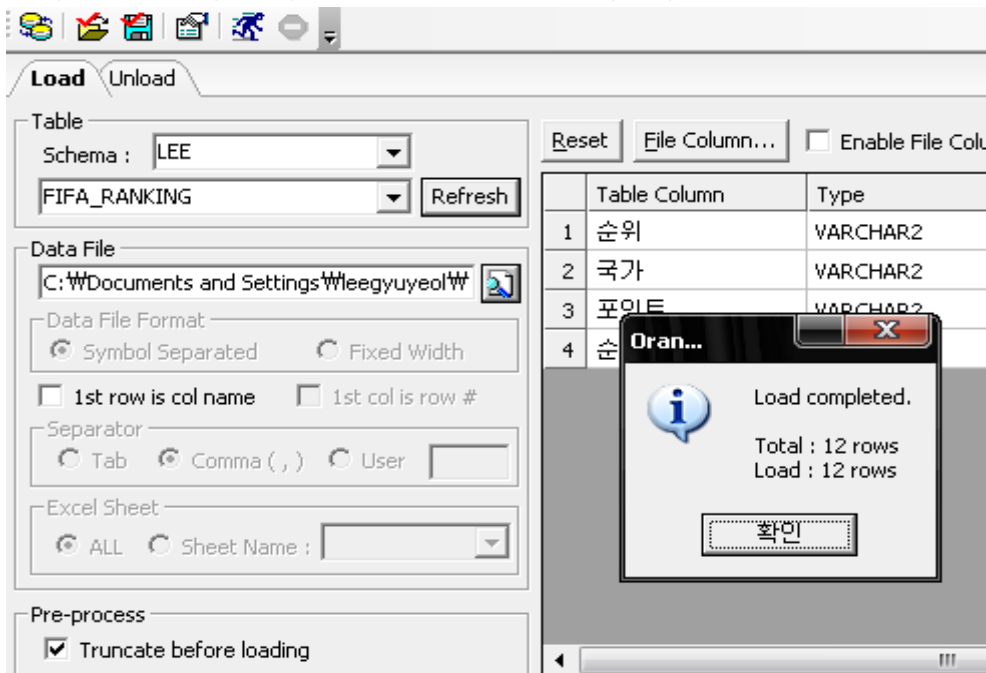
## 3. Loader 의 화면구성

- TABLE 그룹에서 스키마나 loader 할 테이블을 지정한다. Refresh 누르면 테이블을 갱신한다. 테이블을 선택하면 옆의 화면서 테이블 컬럼 정보가 표시된다.
- Data file 그룹에서는 loader 할 데이터 파일을 선택하고 파일 포맷에 컬럼 구분기호가 있으면 Symbol Separated, 일정한 폭으로 되어 있다면 Fixed Width 를 선택한다. Symbol Separated 를 선택하면 Separator 그룹에서 Tab, Comma, User(입력창에 직접 입력)선택 창이 활성화 된다.
- Data file 을 선택하면 Loader 화면 아래에 파일 포맷이 제대로 입력되었나 확인 할 수 있다. 선택한 Datafile 의 첫 번째 행에 컬럼 명이 표시되어 있으면 1st row is col name, 첫 번째 컬럼이 행번호면 1st col is row# 을 체크하면 된다.
- Pre-process 그룹에서 Truncate before loading 을 체크하면 loader 되기 전 테이블의 데이터를 삭제하고 loader 한다.
- 실습에 사용된 data file 은 쉼표로 분리되어 저장되는 csv 로 했기 때문에 나머지사항은 선택 안 해도 무관하다.



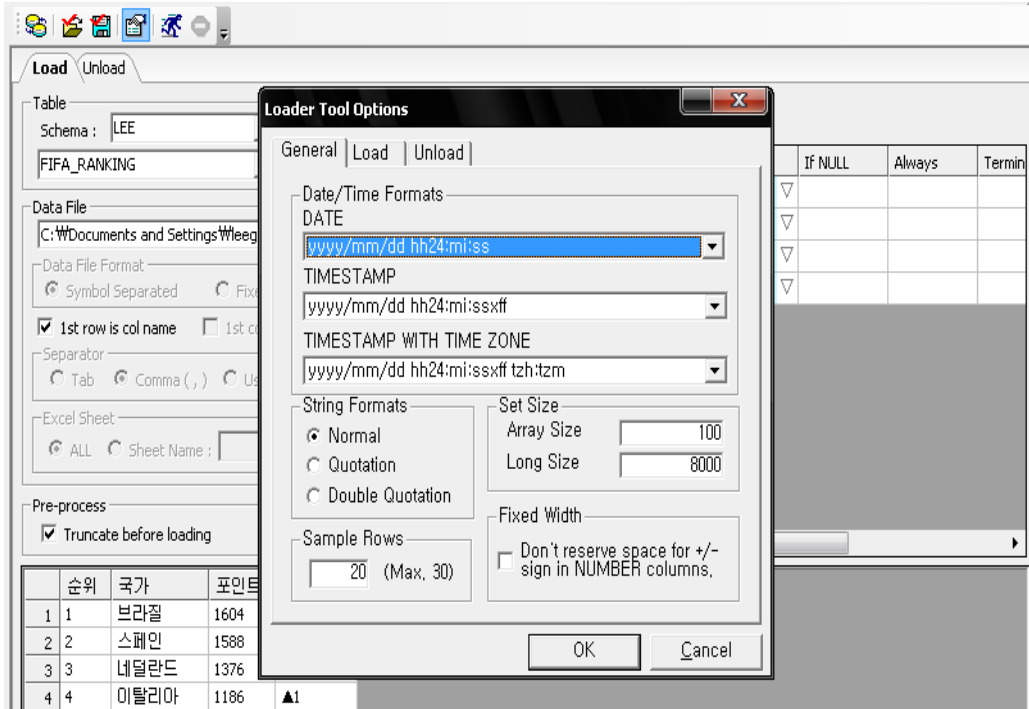
#### 4. 로더 실행 화면

- 로더 메뉴 중 실행 아이콘(사람 달리는 아이콘)을 클릭하면 실제 로더가 수행되며 다음과 같이 총 rows와 실제 loader 된 rows를 보여준다



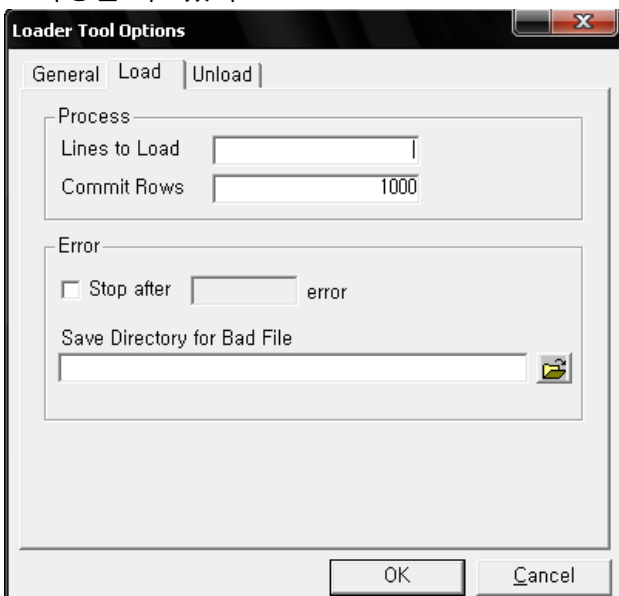
## 5. 옵션 버튼사용(General)

- 옵션버튼을 클릭하면 Loader Tool Options 가 열린다. General 탭에서 Data/Time Formats 그룹에서 포맷을 입력해야 성공적으로 loader 할 수 있다.
- String Formats 그룹은 데이터 파일에 사용된 string data 포맷을 지정한다.



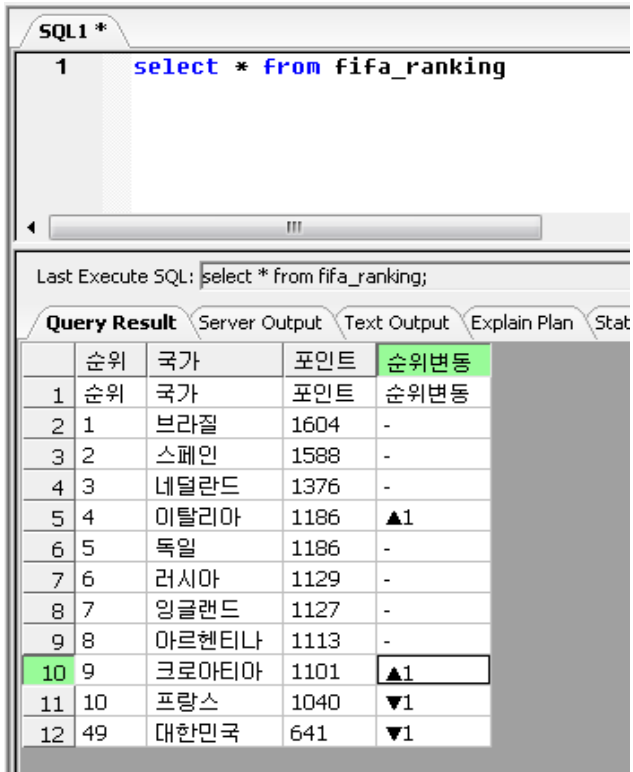
## 6. 옵션 버튼사용(Load)

- Load 탭을 이용해서 데이터 파일의 일부분만 loader 할 수 있다.
- Lines to Load 에서는 loader 할 행 수를 입력 Commit Rows 에서는 몇 행 마다 commit 을 실행 할 것인지를 지정한다 .
- Error 그룹에서는 몇 번의 error 가 발생했을 때 작업을 중지 할 것인지 설정하며 badfile 을 지정할 수 있다.



## 7. Loader 된 데이터 확인

- 정상적으로 데이터가 loader 된 것을 확인 할 수 있다.



SQL1 \*

```
1 select * from fifa_ranking
```

Last Execute SQL: select \* from fifa\_ranking;

Query Result Server Output Text Output Explain Plan Stat

	순위	국가	포인트	순위변동
1	순위	국가	포인트	순위변동
2	1	브라질	1604	-
3	2	스페인	1588	-
4	3	네덜란드	1376	-
5	4	이탈리아	1186	▲1
6	5	독일	1186	-
7	6	러시아	1129	-
8	7	잉글랜드	1127	-
9	8	아르헨티나	1113	-
10	9	크로아티아	1101	▲1
11	10	프랑스	1040	▼1
12	49	대한민국	641	▼1



## 6. 부록

### 6.1. SQL LOADER 성능 향상 기법

- 제어파일과 데이터 파일을 분리하면 여러 로드 세션에서 제어 파일을 다시 사용 할 수 있다.
- 예상되는 데이터 볼륨을 기반으로 공간을 미리 할당하면 로드 시 확장 영역의 동적 할당을 방지하여 로드 속도가 향상된다.
- 테이블에 인덱스가 걸려있을 경우, 먼저 인덱스를 drop 시키고 SQL Loader 로 데이터를 올린 후 인덱스를 생성한다.
- 데이터의 양이 많을 경우 Redo log size 를 충분히 크게 설정한다. Redo log size 가 작을 경우 Redo log switch 할 때 마다 LGWR 에서 timeout 이 발생하여 SQL Loader 에서 hang 현상이 발생 할 수 있다.
- 데이터를 로딩하기 전 데이터를 삭제 후 로딩하기 위해서는 REPLACE 옵션 대신 TRUNCATE 옵션을 사용한다. REPLACE 옵션은 redo log 및 rollback 데이터가 발생하게 되지만 TRUNCATE 옵션은 그렇지 않다.
- Commit 문의 실행을 자주 하지 않을 경우 SQL Loader 가 보다 더 빨리 실행된다
- DIRECT PATH 옵션을 사용하면 데이터가 SQL engine 을 거치지 않고 바로 데이터파일에 write 됨으로 속도가 빨라진다.
- PARALLEL 옵션을 사용한다. 다중 SQL Loader 세션을 사용할 경우 한 개의 테이블도 parallel 하게 loading 될 수 있다.

### 6.2. 키워드 설명

키워드	설 명
USERID	SQL LOADER 로 데이터 로드할 오라클 사용자 및 비밀번호
CONTROL	컨트롤 파일 이름, SQL*Loader을 수행하기 위해서는 항상 지정해 주어야 함
LOG	로그 파일 이름을 지정
BAD	거부된 레코드 모두를 저장하는 배드(Bad) 파일 이름을 지정
DATA	입력 데이터 파일 이름을 지정
DISCARD	로드시 선택되지 않은 레코드가 저장되는 디스카드 파일(선택 사항)
DISCARDMAX	discard의 최대 허용 개수 지정(기본값 : all )
SKIP	스킵 할 논리적 레코드 수(기본값 : 0 )
LOAD	로드 할 논리적 레코드 수(기본값 : all )
ERRORS	허용하는 배드 레코드의 최대 수 지정(기본값 : 50 )

ROWS	일반 경로나 직접 경로에 의해 저장되는 행 수 (기본 값 : 일반 경로 -> 64, 직접 경로 : all)
BINDSIZE	일반 경로에 의한 BIND ARRAY 크기(기본값 : 256000 바이트)
DIRECT	TRUE로 설정되면 SQL*Loader는 직접 경로(Direct Path) 사용 생략 할 경우는 기본 값인 일반 경로(Conevntional Path) 사용
PARFILE	추가적으로 파라미터 파일을 지정
PARALLEL	직접(DIRECT) 경로 방법에서만 사용하며, 다중 병렬 직접 경로에 의해 수행되도록 지정
FILE	병렬 직접 경로에 의한 로드의 경우 임시 세그먼트가 생성될 파일을 지정

### 6.3. 참조문헌

- [http://www.oracle.com/technology/products/database/utilities/htdocs/sql\\_loader\\_overview.html](http://www.oracle.com/technology/products/database/utilities/htdocs/sql_loader_overview.html)
- <http://kr.forums.oracle.com/forums/thread.jspa?messageID=1593858?>
- ORACLE Database Administration Fundamentals