

## 3장 SQL\*Plus 명령어

### 3.1 SQL\*Plus 명령어의 개념

- SQL\*Plus는 SQL문을 실행시키고 그 결과를 볼 수 있도록 오라클에서 제공하는 툴이다.
- SQL은 데이터베이스에서 자료를 검색하고 수정하고 삭제하는 등을 위한 데이터베이스 언어인 반면, SQL\*Plus 명령어는 툴에서 출력 형식을 지정하는 등 환경을 설정한다.
- SQL과 SQL\*Plus 명령어의 차이점

SQL 문	SQL*Plus 명령문
관계형 데이터베이스의 ANSI 표준 언어	SQL 문을 실행 시킬 수 있는 오라클의 툴
여러 줄 실행	한줄 실행
종결문자(;) 필요	종결문자 불요
연결문자 불요	연결문자(-) 필요
키워드 단축 불가	키워드 단축 가능
버퍼에 마지막 명령문 저장	버퍼 저장 안함

- SQL\*Plus 명령어

명령어	기능
LIST, RUN, @, /	편집 명령
SAVE, GET, EDIT, SPOOL	파일 명령
HOST, EXIT	데이터베이스 접속 및 종료
LINE, PAGE	출력 형식

### 3.2 SQL\*Plus 편집 명령

명령어(약어)	설명
LIST (L)	버퍼에 저장된 모든 SQL 문 또는 검색한 라인의 SQL 문을 나타낸다.
/	SQL 문을 보여주지 않고 바로 실행한다.
RUN (R)	버퍼에 저장된 SQL 문을 보여주고 실행한다.

```
SQL> SELECT DISTINCT DEPTNO
FROM EMP;
SQL> L
SQL> /
SQL> R
```

### 3.3 SQL\*Plus 파일 명령어

명령어(약어)	설 명
EDIT (ED)	파일의 내용을 vi(유닉스)나 notepad(윈도우즈)와 같은 에디터로 읽어 편집할 수 있도록 한다.
HOST	오라클을 종료하지 않고 OS 명령을 수행할 수 있도록 OS 환경으로 잠시 빠져 나갈 수 있도록 한다. OS Prompt 상에서 Exit 하면 다시 오라클 환경으로 돌아온다.
SAVE	SQL 버퍼 내의 현재 내용을 실제 파일로 저장한다.
@	SQL 파일에 저장된 내용을 실행한다.
SPOOL	오라클 화면을 갈무리하여 파일로 저장한다.
GET	파일의 내용을 SQL 버퍼로 읽어 들인다.
EXIT	오라클을 종료한다.

### 3.3.1 파일에 내용을 메모장에서 편집하게 하는 EDIT(ED)

- SQL은 파일의 내용을 메모장에서 쉽게 편집할 수 있도록 ED[IT] 명령어를 제공한다.
- 주의 할 점은 SQL 버퍼를 편집기로 열었을 때 명령어문 끝에 붙였던 종결문자 ; 가 편집화면에서는 /로 대체 된다는 점이다.

```
SQL> SELECT DISTINCT DEPTNO
FROM EMP;
SQL> ED
```

### 3.3.2 DOS 프롬프트로 나가게 하는 HOST

- SQL 명령문이 저장된 버퍼는 파일 형태인데 그 파일 이름은 "afiedt.buf"입니다.
- 오라클을 종료하지 않고 DOS 명령어인 dir를 사용하여 afiedt.buf 파일이 존재하는 것을 확인하려면 DOS 환경으로 나가는 HOST 명령어를 사용한다.

```
SQL> HOST
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

D:\Temp>EXIT

SQL>
```

### 3.3.3 사용자가 현재 수행 중인 쿼리문을 저장하는 SAVE

- SQL\*Plus 에서는 사용자가 가장 최근에 수행한 쿼리문을 파일로 저장할 수 있도록 하는 SAVE 명령어를 제공한다.

```
SQL> L
 1 SELECT DISTINCT DEPTNO
 2* FROM EMP
SQL> SAVE TEST.SQL
FILE TEST.SQL(이)가 생성되었습니다
SQL> HOST
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

D:\Temp>DIR
```

### 3.3.4 SQL 파일에 저장된 명령어를 실행하는 @

- @ 명령어는 확장자가 .SQL인 파일에 저장되어 있는 쿼리문을 실행시키기 위해서 사용한다.

```
SQL> @TEST.SQL
```

```
DEPTNO
-----
      30
      20
      10

SQL>
```

### 3.3.5 갈무리 기능을 하는 SP00L

- SAVE 명령어가 SQL문 자체를 저장하는데 비해 SP00L명령어는 SQL문과 실행된 쿼리 결과를 파일로 기록하는 명령어이다.
- 즉 화면에 보여지는 내용 전체를 갈무리 해서 하나의 파일로 만든다.

### 3.3.6 저장한 SQL 명령어를 가져오는 GET

- SAVE 명령어를 사용하여 저장한 SQL 명령어를 다시 사용할 수 있는데 이때 사용하는 명령어가 GET 이다. GET 명령어도 SAVE 명령어와 마찬가지로 파일 이름만 기술하고 확장자를 기술하지 않으면 기본적으로 확장자를 .SQL로 인식한다.

```
SQL> GET TEST.SQL
1  SELECT DISTINCT DEPTNO
2* FROM EMP
SQL>
```

## 3.4 시스템 변수 조작을 위한 SET 명령어

### 3.4.1 컬럼 제목의 출력 여부를 결정하는 HEADING(HEA) 변수

- HEADING은 SELECT 명령어를 수행한 후 실행결과가 출력될 때 컬럼의 제목을 출력할 것인지의 여부를 제어한다.

```
SQL> SET HEADING OFF
SQL> SET HEADING ON
```

### 3.4.2 라인 당 출력할 문자의 수를 결정하는 LINESIZE 변수

- 시스템 변수 LINESIZE 는 라인 당 출력될 문자 수를 결정한다.
- 디폴트 값은 80 이므로 SELECT 문의 출력 결과를 출력할 때 한 라인에 80까지만 출력한다.

```
SQL> SET LINESIZE 100
```

### 3.4.3 페이지 당 출력할 라인의 수를 결정하는 PAGESIZE 변수

- PAGESIZE 변수는 SQL 명령문의 실행 결과에 대해서 출력될 수 있는 페이지의 크기를 설정하는 변수로서 한 페이지에 컬럼 제목, 컬럼 제목과 데이터 구분선, 페이지를 구분하기 위한 공백 라인을 포함한다.

```
SQL> SET PAGESIZE 10
```

#### 3.4.4 컬럼에 저장된 데이터의 출력 형식 변경을 위한 COLUMN FORMAT

```
SQL> COLUMN ENAME FORMAT A25  
SQL> COLUMN SAL FORMAT 9,999,999  
SQL> COLUMN COMM FORMAT 0,000,000
```