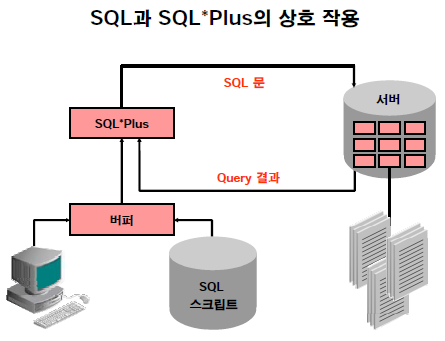
**SQL 과 SQL\*Plus**

1. **SQL과 SQL\*Plus 명령의 개념**

|  |
| --- |
| **SQL(STRUCTURED QUERY LANGUAGE) 란?**   * 관계DB를 처리하기 위해 고안된 언어로, 독자적인 문법을 갖는 DB표준언어 (ISO에서 지정) * 대다수 데이터 베이스는 SQL를 통하게 된다.   **SQL\*PLUS란?**   * SQL언어를 구현하여 오라클 RDBMS를 관리할 수 있는 오라클사의 CLIENT TOOL 제품명 |



**2. SQL과 SQL\*Plus 명령의 특징**

**SQL 명령**

* 데이터에 접근하기 위한 언어 입니다.
* ANSI 표준 SQL을 기반으로 합니다.
* 데이터베이스의 데이터 및 테이블을 조작합니다.
* 모든 명령의 끝은 ;(세미콜론)으로 끝나야 한다.
* 여러 줄에 입력할 수 있다.
* 가장 최근의 SQL명령은 SQL버퍼에 남아있다.
* 프로그래밍에 별로 경험이 없는 사용자는 물론 다양한 이용자들이 사용 가능 하다.
* 비절차적 언어이다.
* 시스템 개발 및 유지보수에 소요되는 시간을 줄일 수 있다.
* 영어와 유사한 언어이다.

**SQL\*PLUS 명령**

* 종료 문자를 사용할 필요 없이 명령을 즉시 시행합니다.
* 버퍼의 SQL문을 편집할 수 있다.
* 스크립트 파일을 저장, 로드 및 실행합니다.
* 파일에서 SQL 및 PL/SQL 입력문을 받아들인다.
* 환경설정을 제어한다.
* 변수를 생성하여 SQL문에 전달하고, 변수 값을 인쇄하고, 화면에 메시지를 출력합니다.
* 검색된 결과를 보고서 형태로 출력 할 수 있도록 설정 가능하다.
* 약어를 사용할 수 있습니다.
* 데이터베이스의 값을 조작할 수 없습니다.

**3. SQL\*Plus 로그인**

**명령 라인에서**

|  |
| --- |
| 형식  sqlplus [username[/password[@connect\_identifier]]] |

1. DB 서버에 로그인 한다.
2. 명령어를 입력한다.

* Username : 데이터베이스 사용자명

쓰지 않으면 errOR.

* Password : 데이터베이스 암호

쓰지 않으면 재 요청.

* @connect\_identifier : 데이터베이스 연결 스트링

예) sqlplus hr/hr@127.0.0.1

**4. Table 구조**

만들어 놓은 테이블의 구조를 확인할 때 사용한다.

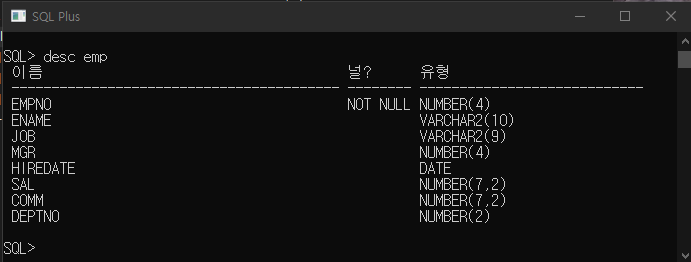
형식

DESC[RIBE] *table\_name*

사용법)

SQL> DESCRIBE emp

SQL> DESC emp



**테이블 구조 확인 결과**

**5. Editing Commands**

SQL\*Plus 명령어는 한번에 한 라인이 입력되며 SQL 버퍼에 저장되지 않는다.

- 명령어를 완료하기 전에 리턴 키를 눌렀을 경우, SQL\*Plus는 라인번호를 나타낸다.

- 종료 문자 (세미콜론 또는 슬래시)를 입력하거나 또는 리턴 키를 두 번 눌러서 SQL 버퍼를 종료한다. 그러면 SQL 프롬프트가 나타난다.

- SQL 프롬프트마다 오직 하나의 SQL\*Plus 명령어만을 입력할 수 있다. SQL\*Plus 명령어를 다음 라인에 계속 하려면 현재 라인의 끝에 하이픈(-)을 추가한다.

|  |  |
| --- | --- |
| L[IST] | SQL Buffer내의 바로 직전 수행한 SQL문의 모든 Line을 List한다. |

SQL> L

|  |  |
| --- | --- |
| L[IST] *n* | SQL Buffer내의 n line 을 Display 한다. |

SQL> SELECT empno, ename

2 FROM emp

3\* WHERE deptno =10;

SQL> L 2

2\* FROM emp

|  |  |
| --- | --- |
| L[IST] *m n* | m Line부터 n Line까지 List 한다. |

SQL> SELECT \*

2 FROM emp

3\* WHERE empno = '7369';

SQL> L 2 3

2 FROM emp

3\* WHERE empno = '7369'

|  |  |
| --- | --- |
| R[UN] | SQL Buffer내의 내용을 Display 하고 실행한다. |

SQL> R

1 SELECT \*

2 FROM emp

3\* WHERE empno = '7369'

SQL문 실행

|  |  |
| --- | --- |
| / | SQL Buffer 내의 내용을 실행만 한다. |

SQL> /

SQL문 실행

|  |  |
| --- | --- |
| A[PPEND] *text* | Current line 끝에 text를 추가한다. |

SQL> SELECT \*

2 FROM emp

3\* WHERE empno = '7369' ;

SQL> APPEND AND ename = 'SMITH'

3 WHERE empno = ‘7369’ AND ename = 'SMITH'

SQL> /

SQL> R

1 SELECT \*

2 FROM emp

3\* WHERE empno = '7369' AND ename = 'SMITH'

|  |  |
| --- | --- |
| *n text* | n line 내용을 text로 교체  n=0이면 1 line 앞으로 line 추가 후 입력 |

SQL> SELECT \*

2 FROM emp

3\* WHERE empno = '7369';

SQL> 3 WHERE ename = 'SMITH'

SQL> L

1 SELECT \*

2 FROM emp

3\* WHERE ename = 'SMITH'

|  |  |
| --- | --- |
| CHANGE | *C/old/new* : Current line의 old String을 new String으로 교체한다. 대/소문자 구분이 없다.  new string이 없으면 old string이 삭제된다. |

SQL> SELECT \*

2\* FROM empp;

SQL> L2

SQL> FROM empp

SQL> c/empp/emp/

SQL> R

*C/text* : text 를 삭제한다. //new string이 존재 하지 않는다.

|  |  |
| --- | --- |
| DEL | Current Line을 삭제한다. |

SQL> SELECT \*

2 FROM emp

3\* WHERE sal > 1000;

SQL> DEL

SQL> L

SQL> SELECT \*

2\* FROM emp

|  |  |
| --- | --- |
| CL[EAR] BUFF[ER] | SQL Buffer 의 내용 Clear한다. |

SQL> SELECT \*

2 FROM emp

3\* WHERE sal > 1000;

SQL> CL BUFF

buffer 소거되었습니다.

SQL> R

SP2-0103: SQL 버퍼에 실행할 사항이 없습니다.

|  |  |
| --- | --- |
| I[NPUT] *text* | Current line 밑에 한 Line 추가 후 text 입력 |

SQL> SELECT \*

2 FROM emp

3\* WHERE sal > 900 AND job = ‘CLERK’;

SQL> I AND hiredate >= To\_Date(‘910101’,’YYMMDD’)

SQL> R

SQL> SELECT \*

2 FROM emp

3 WHERE sal > 900 AND job = ‘CLERK’;

4\* AND hiredate >= To\_Date(‘910101’,’YYMMDD’)

수행 결과 표시

SQL> SELECT \*

2 FROM emp

3 WHERE empno = '7934';

SQL> i

4 AND ename = 'MILLER'

5 AND deptno = '10'

6 CTRL+C

**6. File Commands**

SQL문장은 오라클 서버와 통신한다. SQL\*Plus 명령어는 환경을 제어하고, 질의 결과를 포멧 하며 파일을 관리한다. (Default 확장자는 .sql)

|  |  |
| --- | --- |
| SAVE *filename*[Replace][APPEND] | SQL Buffer내의 현재 내용을 주어진 *filename*으로 저장한다.  *filename*이 이미 존재하는 경우  Replace option을 주면 Overwrite되며, Append option을 주면 file 끝에 Append 된다. |

SQL> SELECT empno, ename

2 FROM emp

3 WHERE ename Like ‘S%’

4\* AND hiredate >= to\_date(‘910101’, ’YYMMDD’);

SQL> SAVE emp

작업 directory 위치에 file emp.sql(이)가 생성되었습니다

SQL> host

Host 명령어로 표시되는 working directory 위치에 지정한 이름으로 파일이 생성된다.

|  |  |
| --- | --- |
| GET *filename* | *filename*내의 내용을 Display 하고 SQL Buffer에 기록된다. |

SQL> GET emp

SQL> SELECT empno, ename

2 FROM emp

3 WHERE ename Like ‘S%’

4\* AND hiredate >= to\_date(‘910101’, ’YYMMDD’);

|  |  |
| --- | --- |
| START *filename* | *filename*내의 내용을 실행만 한다. |

SQL> START emp

SQL문 실행

|  |  |
| --- | --- |
| @filename | START와 동일하다. |

SQL> @emp

|  |  |
| --- | --- |
| ED[IT] *filename* | *filename*내의 내용을 Editor로 읽어서 수정을 할 수 있도록 한다.  *filename*을 지정하지 않으면 afiedt.buf 이라고 하는 File로 SQL Buffer의 내용을 수정 할 수 있다. |

SQL> ED emp

|  |  |
| --- | --- |
| SPOOL *filename* | *filename*으로 Query의 결과를 저장한다. |

SQL> SPOOL title

SQL> SELECT \*

2 FROM emp;

SQL문 실행

SQL> SPOOL OFF

Host 명령어로 표시되어지는 working directory 위치에 title.LST 파일 생성

|  |  |
| --- | --- |
| HOST  [OS command] | SQL\*Plus를 종료 하지 않고 OS Command를 수행할 수 있도록 하며, OS Prompt상에서 종료 하면 다시 SQL\*Plus로 되돌아 간다. |

SQL> HOST

OS상의 작업 directory 위기 경로가 표시된다.

|  |  |
| --- | --- |
| EXIT | SQL\*Plus를 종료 한다. |

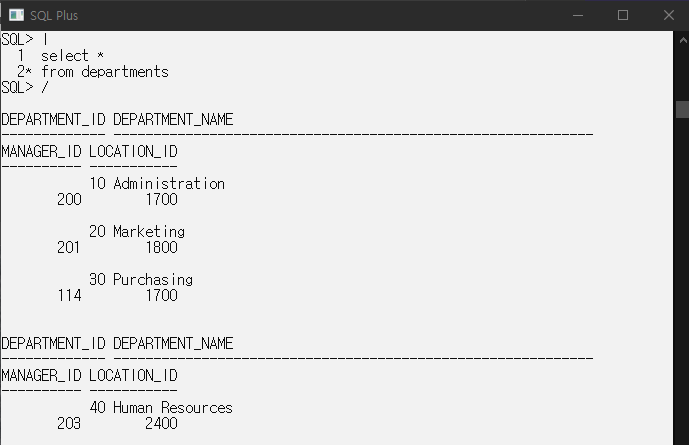
SQL> EXIT

**7. 환경설정 Commands**

|  |  |
| --- | --- |
| COL[UMN] column FORMAT A*nn*|9*nn* | 화면에 표시되는 컬럼명 자리수를 지정함. |

SQL> SELECT \*

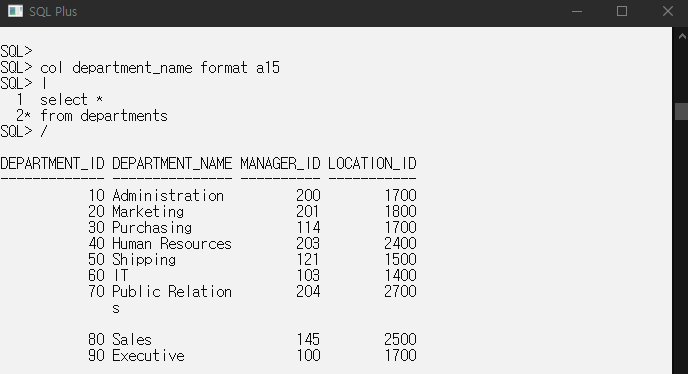
2\* FROM departments;



SQL> COL department\_name FORMAT a15

SQL> SELECT \*

2\* FROM departments;



|  |  |
| --- | --- |
| SET ECHO ON|OFF | @filename으로 실행 시 file내의 명령을 Display 하고 수행할지를 결정한다. |

SQL> SET ECHO OFF

SQL> @emp //실행 결과만 출력

SQL> SET ECHO ON

SQL> @emp //SQL 명령 Display후 실행 결과 출력

|  |  |
| --- | --- |
| SET FEED[BACK]n|ON|OFF | 주어진 n 보다 같거나 많은 Row가 조회되면 조회된 건수를 Display한다.  Default는 6이다. |

SQL> SET FEEDBACK 6

SQL> SELECT \*

2 FROM emp

3\* WHERE deptno = 10;

SQL> SET FEEDBACK 2

SQL> SELECT \*

2 FROM emp

3\* WHERE deptno = 10;

|  |  |
| --- | --- |
| SET HEA[DING] ON|OFF | 조회되는 결과에 Column Heading의 표기 여부를 결정한다. |

SQL> SET HEADING OFF

SQL> SELECT \*

2\* FROM emp; // column heading이 나타나지 않는다.

SQL> SET HEADING ON

SQL> SELECT \*

2\* FROM emp; // column heading이 나타난다.

|  |  |
| --- | --- |
| SET LIN[ESIZE] *n* | 조회되는 결과의 Line당 Character수를 결정한다.  Default line은 80이다. |

SQL> SET LINESIZE 20

SQL> SELECT \*

2\* FROM emp;

SQL> SET LINESIZE 80

SQL> SELECT \*

2\* FROM emp;

|  |  |
| --- | --- |
| SET PAGES[IZE] *n* | 조회되는 결과의 Page당 Line 수를 결정한다.  (SPACE + Column Heading 포함)  Default는 24이다. |

SQL> SET PAGESIZE 10

SQL> SELECT \*

2\* FROM emp;

SQL> SET PAGESIZE 20

SQL> SELECT \*

2\* FROM emp;

|  |  |
| --- | --- |
| SET PAU[SE] ON|OFF|*text* | 조회되는 결과가 많아 화면에 Scrolling될 때 일시 중지 여부를 결정한다.  text인 경우 PAUSE 될 때 마다 text가 Display된다. |

SQL> SET PAUSE ON

SQL> SELECT \*

2\* FROM emp; // ON 인 경우 PAGESIZE단위로 PAUSE된다.

|  |  |
| --- | --- |
| SET TI[ME] ON|OFF | 현재 시각을 SQL\*Plus Prompt앞에 Display할지 여부를 결정한다. |

SQL> SET TIME ON

21:58:56 SQL> SET TIME OFF

SQL>

|  |  |
| --- | --- |
| SET TIMI[NG]ON|OFF | SQL이나 PL/SQL 이 처리되는 시간을(mili Second단위) Display할지 여부를 결정한다.  단 주의해야 할 사항은 화면에 Display시키는 시간이 포함된다는 점이다. |

SQL> SET TIMING ON

SQL> SELECT \*

2\* FROM emp;

조회 결과 출력

경과: 00:00:00.10

SQL> SET TIMING OFF