

Chapter 8. SQL*Plus 변수와 환경설정

이 장에서는 SQL*Plus와 iSQL*Plus에서 변수의 사용방법 및 사용자 고유의 환경을 설정하는 방법을 알아본다. 또한, 데이터베이스 검색 결과를 알아 보기 쉽도록 포매팅 해주는 명령과 스크립트 파일의 작성 방법을 설명한다.

대체 변수

대체 변수(Substitution variable)는 SQL*Plus와 iSQL*Plus에서 사용할 수 있는 변수이다. 예를 들어, 사원 테이블에서 사번을 조건으로 급여를 검색하는 SQL 문장을 자주 작성해야 한다고 가정하자. 매번 같은 사원의 급여를 검색하는 일보다는 상황에 따라 매번 다른 사원의 급여를 검색하는 일이 더 많을 것이다. 이런 경우에 매번 새로운 SQL을 작성하는 것보다는 검색할 사번을 변수로 지정하여 SQL 문을 작성해두고, 조건에 따라 다른 사번을 변수에 저장한 후 기존에 작성해 두었던 SQL 문을 수정 없이 실행하는 편이 효율적일 것이다. 특히, SQL*Plus나 iSQL*Plus에서 대체 변수를 사용하면 변수가 사용된 SQL 문장이 실행될 때마다 사용자에게 변수의 값을 입력하도록 요청한다.

대체 변수는 대표적으로 다음과 같은 경우에 사용한다.

- 값을 임시 저장하고자 할 때
- SQL 문장들간에 값을 전달하고자 할 때

■ & 대체 변수

변수명 앞에 &를 붙이면 해당 변수가 기술된 SQL 문장을 실행할 때마다 사용자에게 해당 변수의 값을 지정하도록 요구한다. 다음은 SQL*Plus에서 & 대체 변수를 사용한 예이다.

```
SQL> SELECT EMPNO, ENAME, SAL
2 FROM EMP
3 WHERE EMPNO = &EMPNUM;
empnum의 값을 입력하십시오: 7369
구 3: WHERE EMPNO = &EMPNUM
신 3: WHERE EMPNO = 7369
```

EMPNO	ENAME	SAL
7369	SMITH	800

물론, iSQL*Plus에서도 & 대체 변수를 사용할 수 있다. 먼저, iSQL*Plus에서 SQL 문장을 작성하고 **확인**을 클릭한 후, 대체 변수 값을 입력한다.

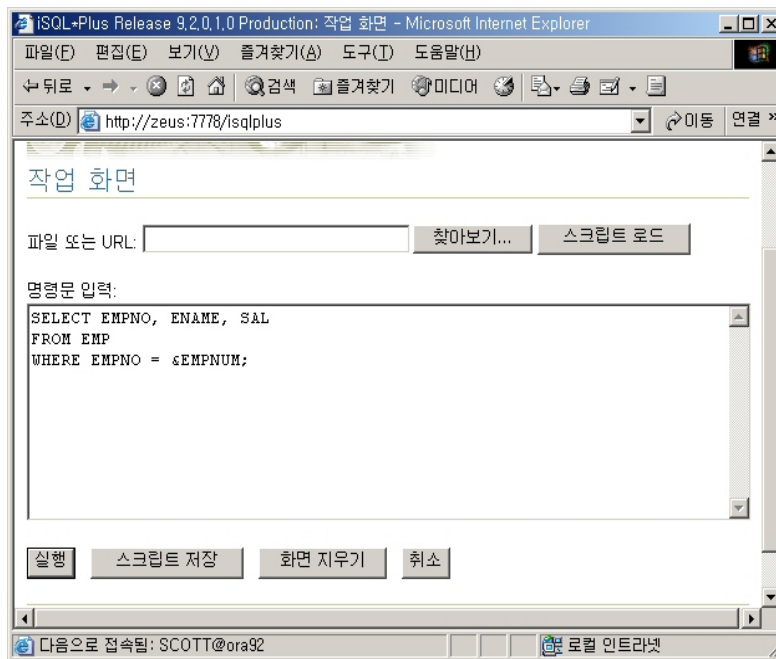


그림 8-1. iSQL*Plus 작업화면

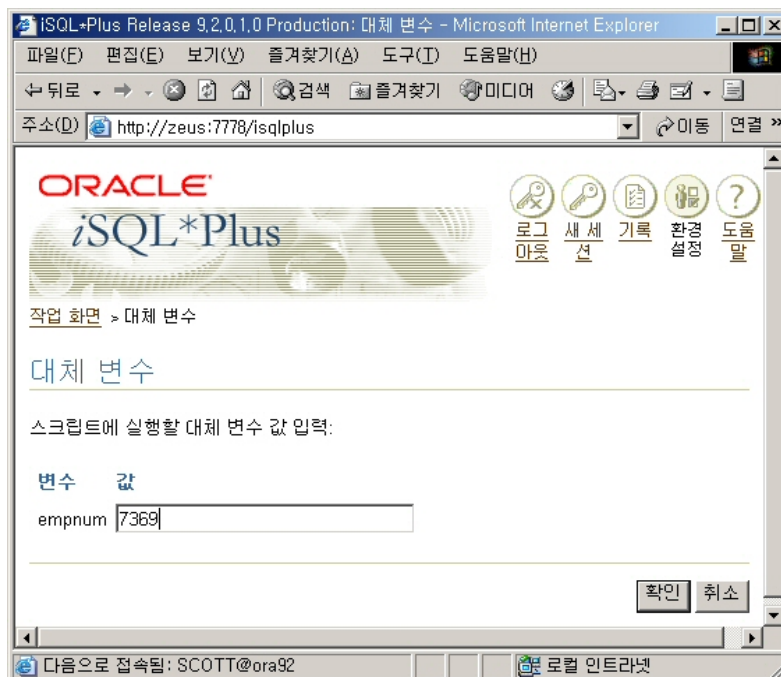


그림 8-2. iSQL*Plus 대체 변수 값 입력

검색 결과는 다음과 같이 나타난다.

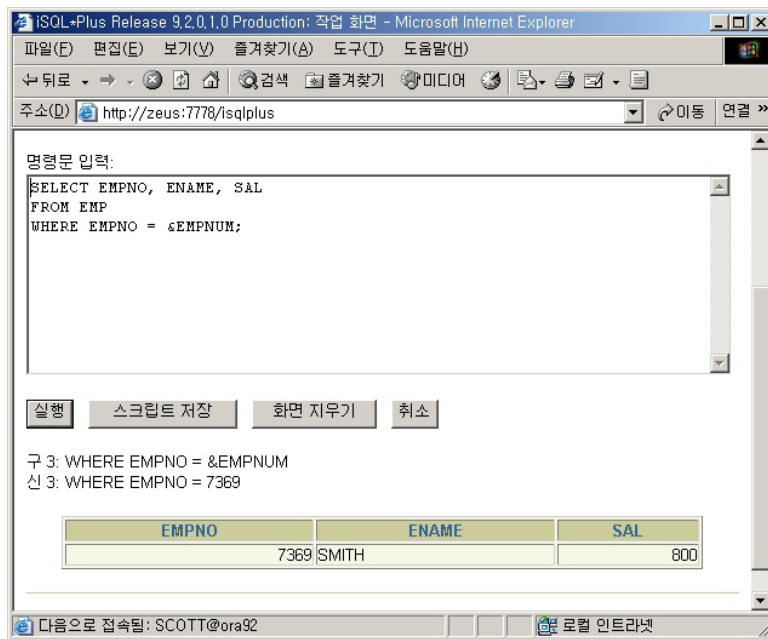


그림 8-3. iSQL*Plus 검색 결과

만약에 대체 변수에 문자 타입이나 날짜 타입의 값을 입력하려면 ' '를 붙여준다.

```
SQL>SELECT EMPNO, ENAME, SAL
2 FROM EMP
3 WHERE ENAME = '&EMPNAME';
empname의 값을 입력하십시오: KING
구 3: WHERE ENAME = '&EMPNAME'
신 3: WHERE ENAME = 'KING'

EMPNO ENAME SAL
-----
7839 KING 5000
```

대체 변수는 다음과 같은 문장에 사용할 수 있다.

- WHERE 절
- ORDER BY 절
- 컬럼명
- 테이블명
- SELECT 절 전체

사용 예는 다음과 같다.

```

SQL> SELECT EMPNO, ENAME, JOB, &COLUMN_NAME
2  FROM EMP
3  WHERE &CONDITION
4  ORDER BY &ORDER_COLUMN;
column_name의 값을 입력하십시오: SAL
구   1: SELECT EMPNO, ENAME, JOB, &COLUMN_NAME
신   1: SELECT EMPNO, ENAME, JOB, SAL
condition의 값을 입력하십시오: SAL > 2000
구   3: WHERE &CONDITION
신   3: WHERE SAL > 2000
order_column의 값을 입력하십시오: ENAME DESC
구   4: ORDER BY &ORDER_COLUMN
신   4: ORDER BY ENAME DESC

```

EMPNO	ENAME	JOB	SAL
7788	SCOTT	ANALYST	3000
7839	KING	PRESIDENT	5000
7566	JONES	MANAGER	2975
7902	FORD	ANALYST	3000
7782	CLARK	MANAGER	2450
7698	BLAKE	MANAGER	2850

6 개의 행이 선택되었습니다.

■ DEFINE, UNDEFINE

대체 변수는 사용되기 전에 미리 선언 할 수 있다. 대체 변수를 선언하는 방법은 다음과 같다.

```
DEFINE variable = value
```

위와 같이 대체 변수를 선언하면 CHAR 타입의 변수가 생성되고 값이 지정 된다. 만약, 변수 값에 여백을 포함하려면 ' '를 붙여주면 되며, 선언 된 대체 변수는 해당 변수가 선언 된 세션내에서만 사용가능하다.

다음은 대체 변수를 선언하고, 선언 된 대체 변수를 확인하는 예이다.

```

SQL> DEFINE job_title = salesman
SQL> DEFINE job_title
DEFINE JOB_TITLE          = "salesman" (CHAR)

```

선언 된 변수를 해제하는 방법은 다음과 같다.

```

SQL> UNDEFINE job_title
SQL> DEFINE job_title
SP2-0135: 기호 job_title은 UNDEFINED

```

DEFINE 명령을 이용하여 대체 변수를 선언한 후, 선언된 대체 변수를 SQL 문장에서 사용하는 경우 해당 변수가 해제되기 전까지는 사용자가 변수 값을 입력할 수 없음을 주의하여야 한다.

```

SQL> DEFINE empnum = 7369
SQL> SELECT EMPNO, ENAME, SAL
  2 FROM EMP
  3 WHERE EMPNO = &empnum;
구  3: WHERE EMPNO = &empnum
신  3: WHERE EMPNO = 7369

  EMPNO ENAME          SAL
-----
  7369 SMITH           800

```

■ && 대체 변수

대체 변수 앞에 &&를 붙이면 사용자가 SQL 문장을 실행할 때마다 매번 변수값을 요구하지 않는다. 즉, 해당 대체 변수가 DEFINE에 의해 선언되어 있지 않다면 SQL 문장의 최초 실행시에만 변수값을 받아 들이고 그 다음부터는 변수값을 요구하지 않는다.

```

SQL> SELECT EMPNO, ENAME, SAL
  2 FROM EMP
  3 WHERE EMPNO = &&EMPNUM;
empnum의 값을 입력하십시오: 7369
구  3: WHERE EMPNO = &&EMPNUM
신  3: WHERE EMPNO = 7369

  EMPNO ENAME          SAL
-----
  7369 SMITH           800

SQL> SELECT EMPNO, ENAME, SAL
  2 FROM EMP
  3 WHERE EMPNO = &&EMPNUM;
구  3: WHERE EMPNO = &&EMPNUM
신  3: WHERE EMPNO = 7369

  EMPNO ENAME          SAL
-----
  7369 SMITH           800

```

대체 변수 사용시, 입력 값의 확인 메시지를 보이지 않게 하려면 VERIFY 명령을 사용한다.

```

SQL> SET VERIFY OFF
SQL> SELECT EMPNO, ENAME, SAL
  2 FROM EMP
  3 WHERE SAL > &EMPSAL;
empsal의 값을 입력하십시오: 2000

  EMPNO ENAME          SAL
-----
  7566 JONES           2975
  7698 BLAKE           2850
  ...
  7902 FORD             3000

6 개의 행이 선택되었습니다.

```

다시 원래대로 메시지가 보이도록 하려면 SET VERIFY ON 명령을 사용하면 된다.

환경 설정

SQL*Plus 및 iSQL*Plus의 환경 설정을 변경하기 위해서는 SET 명령을 이용해 관련 시스템 변수의 설정값을 수정해야 한다.

```
SET system_variable value
```

SHOW 명령을 이용하여 시스템 변수의 설정값을 확인할 수 있다.

```
SHOW system_variable
```

다음은 시스템 변수의 설정값을 수정하고 확인하는 예이다.

```
SQL> SHOW PAGESIZE
pagesize 14
SQL> SET PAGESIZE 30
SQL> SHOW PAGESIZE
pagesize 30
```

또한, SQL*Plus 메뉴의 **옵션-환경**을 선택하면 시스템 변수를 설정할 수 있는 **환경** 대화상자가 표시된다.

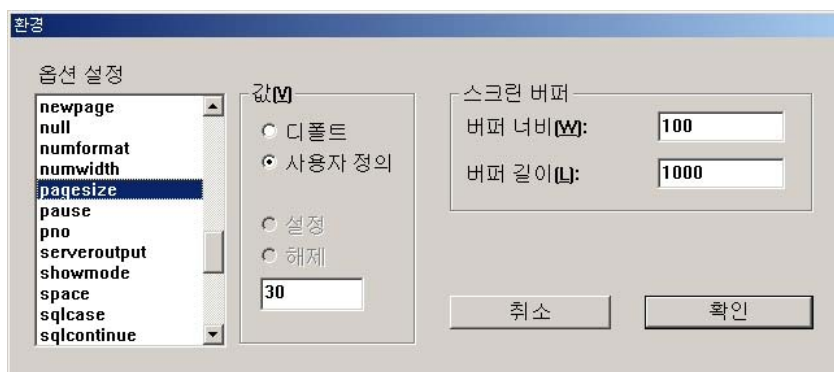


그림 8-4. 시스템 변수 설정

iSQL*Plus에서는 우측 상단의 **환경설정** 링크를 클릭하면 **환경설정** 페이지가 표시되고 **시스템 변수 설정의 이동** 버튼을 클릭한다. **시스템 변수** 페이지에서 빠른 링크를 이용하여 시스템 변수를 입력 또는 선택한 후 **이동** 버튼을 클릭하거나 마우스로 스크롤하여 해당 시스템 변수를 변경할 수 있는 항목이 있는 곳으로 이동한다.

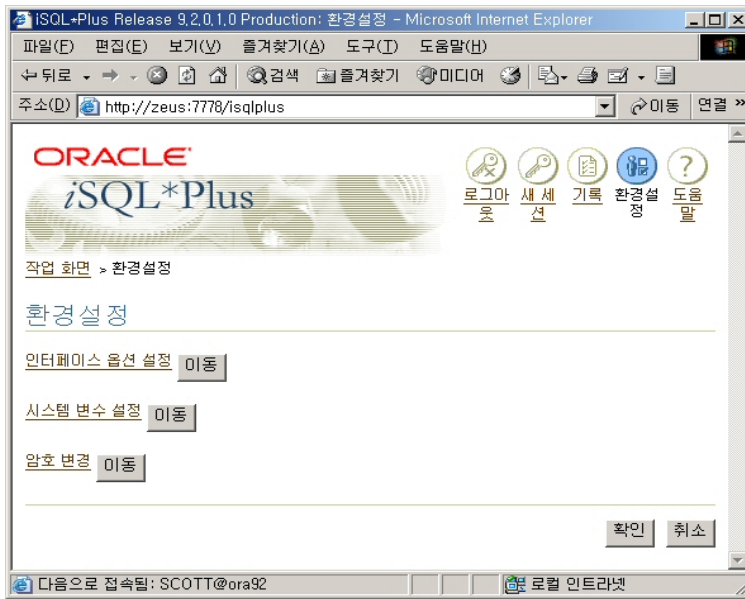


그림 8-5. 환경 설정 페이지

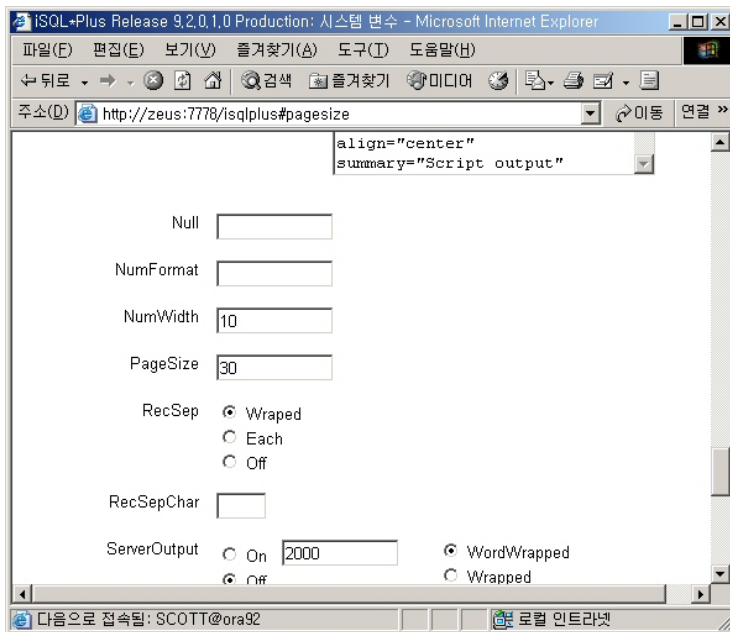


그림 8-6. 시스템 변수 설정

시스템 변수의 개수는 상당히 많지만 실제 사용하는 변수는 그리 많지 않다. 아래 표는 활용도가 높은 시스템 변수의 예이다.

표 8-1. 대표적인 시스템 변수

시스템 변수	설명
LINESIZE	검색 결과의 출력 너비를 지정
PAGESIZE	검색 결과의 한 페이지에 출력될 라인 수를 지정
HEADING	검색 결과에서 컬럼명의 표시 여부
FEEDBACK	SQL 문장 실행 후, 결과 메시지 표시 여부
SQLPROMPT	프롬프트 지정

포맷 명령

SQL*Plus 및 iSQL*Plus의 포맷 명령은 SQL 문장에 의한 검색 결과를 보고서로 작성할 수 있도록 도와주는 기능이다. 포맷 명령어는 다음과 같은 것들이 있다.

표 8-2. 포맷 명령

포맷 명령	설명
COL[UMN] [<i>column option</i>]	컬럼 포맷을 설정
TTI[TLE] [<i>text</i> OFF ON]	검색 결과의 각 페이지 최상단에 표시될 제목
BTI[TLE] [<i>text</i> OFF ON]	검색 결과의 각 페이지 최하단에 표시될 제목
BRE[AK] [ON <i>report_element</i>]	중복된 값의 출력을 억제

위 포맷 명령어 또한 대체 변수처럼 해당 세션에서만 유효하다.

■ COLUMN

COLUMN 명령을 사용하는 방법은 다음과 같다.

```
COL[UMN] [{column|alias} [option]]
```

위에서 *option*으로 지정할 수 있는 항목은 다음과 같다.

표 8-3. COLUMN 명령 *option*

Option	설명
CLE[AR]	컬럼 포맷을 삭제
HEA[DING] <i>text</i>	컬럼명을 <i>text</i> 로 지정(컬럼명의 는 두줄에 걸쳐서 출력)
FOR[MAT] <i>format</i>	컬럼 데이터의 출력 형식
NOPRI[NT]	컬럼을 출력하지 않음
NUL[L] <i>text</i>	컬럼값이 NULL이면 <i>text</i> 출력
PRI[NT]	컬럼을 출력

지정 가능한 컬럼 *format*은 다음과 같다.

표 8-4. 컬럼 *format*

항목	설명	예	결과
9	숫자 1자리	999999	1234
0	숫자 1자리. 숫자가 지정되지 않으면 0으로 출력	099999	001234
\$	달러 표시	\$9999	\$1234
L	지역 화폐단위	L9999	₩1234
.	소수점	9999.99	1234.00
,	천단위 구분자	9,999	1,234

COLUMN 명령을 이용하여 검색 결과를 보기 좋게 수정해보자. 포맷 명령을 사용하기 전의 출력결과가 다음과 같다고 가정한다.

SQL> SELECT EMPNO, ENAME, JOB, MGR, SAL FROM EMP;				
EMPNO	ENAME	JOB	MGR	SAL
7369	SMITH	CLERK	7902	800
7499	ALLEN	SALESMAN	7698	1600
7521	WARD	SALESMAN	7698	1250
7566	JONES	MANAGER	7839	2975
7654	MARTIN	SALESMAN	7698	1250
7698	BLAKE	MANAGER	7839	2850
7782	CLARK	MANAGER	7839	2450
7788	SCOTT	ANALYST	7566	3000
7839	KING	PRESIDENT		5000
7844	TURNER	SALESMAN	7698	1500
7876	ADAMS	CLERK	7788	1100
7900	JAMES	CLERK	7698	950
7902	FORD	ANALYST	7566	3000
7934	MILLER	CLERK	7782	1300

위 검색 결과를 다음과 같이 변경해보자.

- EMPNO 컬럼의 컬럼명을 '사번', ENAME 컬럼은 '사원명', JOB 컬럼은 '업무', MGR 컬럼은 '관리자사번', SAL 컬럼은 '급여'로 변경한다. 단, MGR 컬럼의 컬럼명은 '관리자'와 '사번'으로 두줄에 표현되도록 한다.
- EMPNO 컬럼의 폭은 5문자로 줄인다.
- MGR 컬럼의 폭은 5문자로 줄인다.
- ENAME 컬럼의 폭은 7문자로 줄인다.
- ENAME 컬럼은 우측 정렬한다.
- SAL 컬럼은 천단위마다 ,를 붙인다.
- MGR 컬럼의 값이 NULL인 경우는 '사장'으로 출력한다.

```

SQL> COLUMN EMPNO HEADING '사번'
SQL> COLUMN ENAME HEADING '사원명'
SQL> COLUMN JOB HEADING '업무'
SQL> COLUMN MGR HEADING '관리자|사번'
SQL> COLUMN SAL HEADING '급여'
SQL> COLUMN EMPNO FORMAT 99999
SQL> COLUMN MGR FORMAT 99999
SQL> COLUMN ENAME FORMAT A7
SQL> COLUMN ENAME JUSTIFY RIGHT
SQL> COLUMN SAL FORMAT 999,999
SQL> COLUMN MGR NULL '사장'
SQL> SELECT EMPNO, ENAME, JOB, MGR, SAL FROM EMP;

```

사번	사원명	업무	관리자 사번	급여
7369	SMITH	CLERK	7902	800
7499	ALLEN	SALESMAN	7698	1,600
7521	WARD	SALESMAN	7698	1,250
7566	JONES	MANAGER	7839	2,975
7902	FORD	ANALYST	7566	3,000
...				
7934	MILLER	CLERK	7782	1,300

■ BREAK

BREAK 명령은 중복값의 출력을 억제하는 명령이다.

```
BREAK ON column[|alias|row]
```

BREAK 명령은 대상이 되는 컬럼이 ORDER BY에 의해 정렬된 후 검색해야만 의미가 있다.

```

SQL> BREAK ON JOB
SQL> SELECT EMPNO, ENAME, JOB
2 FROM EMP
3 ORDER BY JOB;

```

EMPNO	ENAME	JOB
7788	SCOTT	ANALYST
7902	FORD	
7369	SMITH	CLERK
7876	ADAMS	
7934	MILLER	
7900	JAMES	
7566	JONES	MANAGER
7782	CLARK	
7698	BLAKE	
7839	KING	PRESIDENT
7499	ALLEN	SALESMAN
7654	MARTIN	
7844	TURNER	
7521	WARD	

14 개의 행이 선택되었습니다.

■ TTITLE, BTITLE

TTITLE, BTITLE은 각각 페이지의 최상단, 최하단에 출력할 제목을 지정하는 것이다.

```
TTI[TLE]|BTI[TLE] [text|OFF|ON]
```

```
SQL> TTITLE '부서 리스트'
SQL> BTITLE '대외비'
SQL> SELECT * FROM DEPT;
```

토 Jun 19

부서 리스트

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

대외비

스크립트 파일

스크립트 파일은 복잡한 SQL 문장의 외부의 텍스트 파일로 저장한 것이다. 외부에 저장된 스크립트 파일은 SQL*Plus 및 iSQL*Plus에서 언제든지 불러들여 실행시킬 수 있다. 다음은 스크립트 파일을 작성하는 방법이다.

1. SQL SELECT 문장을 작성하고 올바르게 실행되는지 테스트한다.
2. 작성된 SELECT 문장을 스크립트 파일로 저장한다.
3. 스크립트 파일을 일반 편집기로 불러들인다.
4. SELECT 문장의 앞 부분에 포맷 명령을 추가한다.
5. SELECT 문장의 끝에 ; 또는 /이 있는지 확인한다.
6. SELECT 문장의 뒷 부분에 앞에서 지정한 포맷을 삭제하는 포맷 명령을 작성한다.
7. 스크립트 파일을 저장한다.
8. SQL*Plus 또는 iSQL*Plus에서 스크립트 파일을 실행한다.

복습

iSQL*Plus에서 다음과 같은 보고서를 생성하는 스크립트를 작성하시오.

iSQL*Plus Release 9.2.0.1.0 Production: 작업 화면 - Microsoft Internet Explorer

주소(D) http://zeus:7778/isqlplus

토 Jun 19 사원 보고서 페이지 1

사원명	업무	급여
SCOTT	ANALYST	₩3,000.00
FORD		₩3,000.00
SMITH	CLERK	₩800.00
ADAMS		₩1,100.00
MILLER		₩1,300.00
JAMES		₩950.00
JONES	MANAGER	₩2,975.00
CLARK		₩2,450.00
BLAKE		₩2,850.00
KING	PRESIDENT	₩5,000.00
ALLEN	SALESMAN	₩1,600.00
MARTIN		₩1,250.00

대외비

토 Jun 19 사원 보고서 페이지 2

사원명	업무	급여
TURNER		₩1,500.00
WARD		₩1,250.00

대외비

breaks 소거되었습니다.

다음으로 접속됨: SCOTT@ora92 로컬 인트라넷