BTEQ

BTEQ 개요

- BTEQ는 Teradata DB에 SQL등을 수행하기 위한 Command Line Utility임(오라클의 SQLPlus 와 유사).
- BTEQ는 두가지 모드로 동작
  - Interactive
  - Batch
- Unix, Linux, Window Platform 지원

BTEQ 스크립트에서 사용되는 주요 Command/Term

- .LOGON: Teradata DB에 Logon 시 사용. 접속 HOST 정보, Username, Password 입력함.
- .ACTIVITYCOUNT: 이전 Query의 반환 row 건수 출력
- .DATABASE: Default Database 설정.
- .IMPORT: 파일을 테이블로 import
- .EXPORT: 테이블을 파일로 export
- .LOGOFF

Command는 대소문자를 구분하지 않고 수행됨. interactive mode에서 .help bteq를 수행하면 모든 Command 를 출력.

기본 BTEQ Script

- interactive mode 수행.

```
bteq
.logon 127.0.01/hr_user
-- 패스워드 입력 <password>;

-- SQL 수행

.logoff;
.exit;
```

- batch mode 수행

1. script 파일이 있는 곳으로 이동하여 아래를 수행
2. bteq < script파일명 > 수행결과파일명 2> 에러발생시_기록파일
3. <는 입력 스트림을 의미. >는 출력 스트림을 의미(없을 경우 모니터 출력). 2> 에러스트림을 의미(없을 경우 모니터 출력)

```
vi test01.bteq
.logon 127.0.0.1/hr_user, hr_user
SELECT TOP 3 * FROM DBC.TABLESV;
.logoff;
.quit

# bteq 실행
bteq < test01.bteq
```

```
#bteq 실행 후 결과를 별도 로그로 저장.
bteq < test01.bteq > test01_result.log
```

Logon, Logoff, Exit 수행

- .LOGON Command는 BTEQ에서 DB Logon 수행.
- interactive mode에서는 .LOGON host 명(또는 ip address)/접속 유저명을 입력한 후에 Enter를 입력하고 이후 Password Prompt에서 해당 접속 유저의 패스워드를 입력
- batch mode에서는 파일에 .LOGON host 명(또는 ip address)/접속 유저명, 접속유저 패스워드와 같은 형태를 파일에 한 라인으로 기재하여 접속
- .LOGOFF 는 BTEQ에서 DB에 LOGOFF 수행함. 여전히 BTEQ 프로세스는 살아있음.
- .EXIT는 BTEQ 프로세스가 종료됨.

SQL 수행

- SQL을 수행하면 결과를 리턴. 아래는 SELECT TOP 5 * FROM HR.EMP; 를 BTEQ에서 수행함

```
SELECT TOP 5 * FROM HR.EMP;

SELECT TOP 5 * FROM HR.EMP;

 *** Query completed. 5 rows found. 8 columns returned.
 *** Total elapsed time was 1 second.

     empno ename      job          mgr hiredate       sal      comm deptno
---------- ---------- --------- ------ -------- --------- --------- ------
      7839 KING       PRESIDENT      ? 81/11/17   5000.00         ?    10.
      7902 FORD       ANALYST     7566. 81/12/03   3000.00         ?    20.
      7900 JAMES      CLERK       7698. 81/12/03    950.00         ?    30.
      7521 WARD       SALESMAN    7698. 81/02/22   1250.00    500.00    30.
      7698 BLAKE      MANAGER     7839. 81/05/01   2850.00         ?    30.
```

help table 테이블명/Show table 테이블명

테이블이나 View의 컬럼명과 컬럼comment를 간단하게 확인하려면 help table 테이블명을 사용하며, 테이블의 DDL 정보를 확인하기 위해서는 show table 테이블명을, view의 DDL 정보를 확인하려면 show view view명을 사용.

- help table hr.emp 수행

```
HELP TABLE HR.EMP;

HELP TABLE HR.EMP;

 *** Help information returned. 8 rows.
 *** Total elapsed time was 1 second.

Column Name                    Type Comment
------------------------------ ---- -----------
empno                          I    ?
ename                          CV   ?
job                            CV   ?
mgr                            D    ?
hiredate                       DA   ?
sal                            D    ?
comm                           D    ?
deptno                         D    ?
```

- show table 테이블명 수행 결과.

```
------------------------------------------------------------
CREATE MULTISET TABLE HR.EMP ,FALLBACK ,
     NO BEFORE JOURNAL,
     NO AFTER JOURNAL,
     CHECKSUM = DEFAULT,
     DEFAULT MERGEBLOCKRATIO,
     MAP = TD_MAP1
     (
      empno INTEGER NOT NULL,
      ename VARCHAR(10) CHARACTER SET LATIN CASESPECIFIC,
      job VARCHAR(9) CHARACTER SET LATIN CASESPECIFIC,
      mgr DECIMAL(4,0),
      hiredate DATE FORMAT 'YY/MM/DD',
      sal DECIMAL(7,2),
      comm DECIMAL(7,2),
      deptno DECIMAL(2,0),
PRIMARY KEY ( empno ))
;
```

- view의 메타 정보를 보기 위해서는 help view view명, show view view명 수행

```
create view test_view as (select * from hr.emp);

help view test_view;

show view test_view;

drop view test_view
```

BTEQ Session의 Character Set 설정

- BTEQ에서 한글등의 처리를 위해서는 Character Set 를 UTF8 로 변환. 기본은 ASCII임.

```
.SET SESSION CHARSET 'UTF8';
```

BTEQ 스크립트 주석

- 기본 주석 문자는 멀티라인의 경우 /* */ 로 한 라인의 경우 — 로 시작.

```
/* 아래는 수행되지 않음 */
/* SELECT * FROM hr.emp */

-- 아래도 수행되지 않음
--.LOGOUT
```

이전에 수행한 명령어를 다시 수행

- Command로'=' 를 입력하면 이전에 수행한 명령어를 다시 수행

```
help table hr.emp;

= /* 이전에 수행한 help table hr.emp 다시 수행 */
```

Format 관련 주요 Command

- .SHOW CONTROLS: 현 Output Format 출력
- .SET DEFAULTS : Format이 변경되어서 다시 Default Format으로 돌아가고자 할 때

- .SET WIDTH 120: 스크린 너비를 120 문자로 설정. 기본은 75 문자임. 전체 컬럼들의 Length가 75문자를 넘어갈 경우 컬럼 순으로 75 문자 이후의 컬럼들은 출력 하지 않음.
- .SESET TITLEASHES OFF 로 하면 컬럼명 밑에 - - - 기호를 출력하지 않음. 기본은 ON
- .SET NULL AS '—-': NULL 문자를 —-로 대체
- .SET ECHOREQ OFF: 기본은 ON이며 ON일 경우 입력한 COMMAND와 SQL을 다시 한번 출력해줌. OFF일 경우는 출력안 함.
- .SET SEPERATOR ',' : 컬럼별 분리 문자를 ',' 로 설정.
- .SET QUIET ON은 결과를 출력하지 않음.

Format 적용 출력

- .SET WIDTH는 가장 처음에 고려해야 할 포맷. 기본 Format 적용된 상태에서 SQL을 수행하면 출력이 WIDTH 75에 영향을 받아서 모든 컬럼들이 다 출력되지 않음. .SET WIDTH 3000 등을 적용하여 WIDTH 크기를 늘려야 모든 컬럼들이 출력됨.
- emp 테이블에 description 컬럼을 추가하고, 테스트 문자열을 입력

```
alter table hr.dept add description varchar(100);

update hr.dept set description='description_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'
```

- 기본 WIDTH 75 문자에서 SELECT TOP 5 * FROM hr.dept 수행 시 아래와 같이 75문자에 맞춰서 마지막 description 컬럼값이 생략되어 출력됨. 75문자를 넘어갈 경우 아예 컬럼값이 보이지 않을 수 있음.

```
select * from hr.dept;

select * from hr.dept;

 *** Query completed. 4 rows found. 4 columns returned.
 *** Total elapsed time was 1 second.

     deptno dname          loc           description
----------- -------------- ------------- -----------------------------------
         40 OPERATIONS     BOSTON        description_xxxxxxxxxxxxxxxxxxxxx
         30 SALES          CHICAGO       ?
         20 RESEARCH       DALLAS        ?
         10 ACCOUNTING     NEW YORK      ?
```

전체 컬럼값이 아니라 생략되어 출력

- .SET WIDTH 200으로 200문자로 설정 시 200문자에 맞춰서 컬럼들이 출력됨. 단 화면 사이즈의 제약으로 Multi line으로 출력됨.

```
.set width 200;

.set width 200;
 BTEQ -- Enter your SQL request or BTEQ command:


select * from hr.dept;

select * from hr.dept;

 *** Query completed. 4 rows found. 4 columns returned.
 *** Total elapsed time was 1 second.

     deptno  dname          loc           description
-----------  -------------  ------------  ------------------------------------------------------------
----------------------------------------
         40  OPERATIONS     BOSTON        description xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxx
         30  SALES          CHICAGO       ?
         20  RESEARCH       DALLAS        ?
         10  ACCOUNTING     NEW YORK      ?
```

- .SET DEFAULTS 수행하면 다시 기본 세팅이 됨.

- .SET TITLEDASHES OFF 를 하면 컬럼명 밑에 - - - 기호를 출력하지 않음.

```
.SET TITLEDASHES OFF;

.SET TITLEDASHES OFF;
 BTEQ -- Enter your SQL request or BTEQ command:


SELECT TOP 5 * FROM HR.EMP;

SELECT TOP 5 * FROM HR.EMP;

 *** Query completed. 5 rows found. 8 columns returned.
 *** Total elapsed time was 1 second.

     empno ename      job          mgr hiredate       sal      comm deptno
      7900 JAMES      CLERK      7698. 81/12/03    950.00         ?    30.
      7698 BLAKE      MANAGER    7839. 81/05/01   2850.00         ?    30.
      7839 KING       PRESIDENT     ? 81/11/17   5000.00         ?    10.
      7782 CLARK      MANAGER    7839. 81/06/09   2450.00         ?    10.
      7902 FORD       ANALYST    7566. 81/12/03   3000.00         ?    20.
```

- .SET ECHOREQ OFF는 결과 출력 시 입력한 SQL을 ECHO 형태로 출력하지 않음.

```
SET ECHOREQ OFF;
 *** Warning: Commands not prefixed with a '.' may be submitted as SQL.
 BTEQ -- Enter your SQL request or BTEQ command:
SELECT TOP 5 * from irs.irs_returns;


 *** Query completed. 5 rows found. 9 columns returned.
 *** Total elapsed time was 1 second.

  return_id filing_type         ein  tax_period sub_date
----------- ----------- ----------- ----------- --------------------------
   17243950 EFILE         461112698      201909 8/19/2020 10:38:08 AM
   17426988 EFILE         270231596      201912 11/19/2020 7:09:22 AM
   17303107 EFILE         591491258      201906 9/16/2020 7:06:47 PM
   17373398 EFILE         341689428      201906 10/14/2020 2:04:15 PM
   17049778 EFILE         770082185      201909 1/21/2020 6:48:12 PM

 BTEQ -- Enter your SQL request or BTEQ command:
```

- 컬럼값 출력이 Null 일 경우 기본으로 '?' 문자가 사용됨. 이를 공란으로 변경하려면 .SET NULL '' 적용

```
SELECT TOP 5 * FROM HR.EMP;

 *** Query completed. 5 rows found. 8 columns returned.
 *** Total elapsed time was 1 second.

      empno ename      job              mgr hiredate       sal      comm
----------- ---------- --------- ----------- -------- --------- --------- -
       7900 JAMES      CLERK           7698 81/12/03    950.00         ?
       7698 BLAKE      MANAGER         7839 81/05/01   2850.00         ?
       7839 KING       PRESIDENT          ? 81/11/17   5000.00         ?
       7782 CLARK      MANAGER         7839 81/06/09   2450.00         ?
       7902 FORD       ANALYST         7566 81/12/03   3000.00         ?

 BTEQ -- Enter your SQL request or BTEQ command:

.SET NULL '';

.SET NULL '';
 *** Warning: Null string set to ' ' (blank).
 BTEQ -- Enter your SQL request or BTEQ command:


SELECT TOP 5 * FROM HR.EMP;

SELECT TOP 5 * FROM HR.EMP;

 *** Query completed. 5 rows found. 8 columns returned.
 *** Total elapsed time was 1 second.

      empno ename      job              mgr hiredate       sal      comm
----------- ---------- --------- ----------- -------- --------- --------- -
       7900 JAMES      CLERK           7698 81/12/03    950.00
       7698 BLAKE      MANAGER         7839 81/05/01   2850.00
       7839 KING       PRESIDENT            81/11/17   5000.00
       7782 CLARK      MANAGER         7839 81/06/09   2450.00
       7902 FORD       ANALYST         7566 81/12/03   3000.00

 BTEQ -- Enter your SQL request or BTEQ command:
```

- .SET QUIET ON은 결과를 출력하지 않음. 주로 수행 시간만을 확인하거나 batch 모드에서 수행 결과를 파일로 export할 경우에 설정.

```
.SET QUIET ON;

.SET QUIET ON;
 *** Use '.QUIET OFF' to resume output.
 BTEQ -- Enter your SQL request or BTEQ command:


SELECT TOP 5 * FROM HR.EMP;

SELECT TOP 5 * FROM HR.EMP;
 *** Total elapsed time was 1 second.
```

- 다시 .SET DEFAULTS 로 기본 Format으로 원복

대량 데이터 출력 시 유의

- 대량의 데이터를 출력 할 경우 마지막 row까지 출력을 완료할 때까지 BTEQ는 Fetch를 멈추지 않으므로 출력에 시간이 오래 걸릴 수 있음
- Sample 데이터 정도 출력하여 데이터가 어떻게 구성되었는지 확인 하는 정도의 Sample 데이터를 출력할 경우 select top 5 * from 테이블명; 과 같이 top 키워드를 사용하여 일부만 출력
- 만약 전체 데이터를 계속 출력하는 것을 멈추고 싶다면 ctrl + c 로 break하여 잠시 멈추면 Break received. Input Command: 가 출력됨 여기서 ABORT 를 입력하면 Command line으로 다시 돌아가고, 엔터키등을 누르면 이어서 출력을 계속하게 됨.

OS 명령어 수행

- .OS 를 이용하면 BTEQ내에서 OS 명령어를 사용할 수 있음. 잘못된 OS명령어를 수행해도 BTEQ는 종료되지 않음
- .여러개의 명령어는 .OS 이후 명령어를 ; 로 이어서 수행.

```
# 단일 OS 명령어 수행.
.OS pwd

# 여러 OS 명령어를 이어서 수행. ; 으로 이어서 수행.
.OS pwd; echo '#########'; ls -lia
```

.RUN Command - 스크립트내에서 특정 파일을 수행.

- .RUN Command는 스크립트내에서 주로 반복적으로 사용되는 로직들을 별도의 파일로 저장하고 이를 스크립트내에서 호출하여 수행하는 방식을 지원
- .RUN FILE='파일명'은 파일명으로 지정된 명령어들을 수행.
- run_logon.bteq 파일을 생성하여 아래 코드 logon 명령어를 저장. interactive mode와 다르게 file로 logon 할 시에는 host/db user, db password 와 같은 형태가 되어야 함.

```
.LOGON 127.0.0.1/hr_user, hr_user;
```

- run_format.bteq 파일을 생성하여 아래 코드 format 설정을 저장.

```
.SET WIDTH 2000;
.SET ECHOREQ OFF;
```

- run_sql.bteq 파일을 생성하여 아래 코드 SQL을 저장.

```
SELECT TOP 5 * FROM DBC.TABLESV;

SELECT TOP 5 * FROM HR.EMP;
```

- interactive mode에서 아래 수행

```
.RUN FILE=run_logon.bteq
.RUN FILE=run_format.bteq
.RUN FILE=run_sql.bteq
```

BTEQ Batch 모드에서 수행 시 결과 및 오류 출력 설정

- 아래 코드를 bteq_simple01.bteq 로 저장하고 bteq < bteq_simple01.bteq 로 수행하면 일련의 수행 결과가 모니터로 출력

```
.LOGON 127.0.0.1/hr_user, hr_user
SELECT TOP 5 * from hr.emp;
.LOGOFF
.EXIT
```

- bteq 수행 결과를 모니터가 아닌 특정 파일로 저장. 아래를 수행하고 수행 결과가 /root/bteq_script/bteq_simple01.out 파일로 저장되는 지 확인.

```
bteq < bteq_simple01.bteq > /root/bteq_script/bteq_simple01.out
```

- bteq 수행 시 오류가 발생할 경우 별도의 오류 출력 스트림(2 >)을 지정하지 않으며 표준출력(모니터)나 출력 파일에 오류가 나타남. 아래 코드는 일부로 오류를 발생 시키는 코드를 bteq_error.bteq 로 생성함.

```
.LOGON 127.0.0.1/hr_user, hr_user

/* error sql */
SELECT TOP 5 * from xxx;

/* normal sql */
SELECT TOP 5 * from hr.emp;

.LOGOFF
.EXIT
```

- 해당 코드를 로 저장하고 실행하면 실행 중 오류 메시지가 출력됨.

```
bteq < bteq_error.bteq
```

- 만약 오류 메시지를 별도의 오류 파일로 저장하기를 원할 경우 아래와 같이 오류 출력 파일을 2> 로 설정하여 수행. 아래 수행 시 화면에는 오류 메시지가 출력되지 않고 bteq_error.err 파일에 생성됨을 확인.

```
bteq < bteq_error.bteq 2> bteq_error.err
```

OS Shell Script의 변수를 BTEQ 스크립트로 전달하기

- Shell Script 상의 변수값을 BTEQ 스크립트내에 전달하여 Shell Script와 결합된 BTEQ 스크립트를 생성. 아래 코드를 Shell 에서 수행. <<EOF를 bteq로 전달하면 EOF 가 올 때까지의 문자열들을 bteq 스크립트로 인식. 아래 코드를 shell 상에서 수행.

```
bteq << EOF
.LOGON 127.0.0.1/hr_user, hr_user;
SELECT TOP 5 * FROM hr.emp;
.LOGOFF;
EOF
```

- connect.txt의 접속 스트링을 읽어서 이를 기반으로 bteq에 로그온하는 shell script 생성. 먼저 connect.txt 파일에 아래 저장.

```
127.0.0.1/hr_user, hr_user
```

- bteq_simple.sh 로 아래 코드 생성.

```
connect_string=`cat /root/bteq_script/connect.txt`
echo ${connect_string}

bteq <<EOF
.logon ${connect_string};

SELECT TOP 5 * from hr.emp;
.logoff
.exit
EOF
```

- bteq_simple.sh 쉘 스크립트 수행.

```
chmod +x bteq_simple.sh
./bteq_simple.sh
```

- 위의 스크립트에서 bteq 수행 출력을 console이 아닌 특정 파일로 하려면 EOF Stream 뒤에 > 파일명을 기재.

```
connect_string=`cat /root/bteq_script/connect.txt`
echo ${connect_string}

bteq <<EOF > simple01.log
.logon ${connect_string};

SELECT TOP 5 * from hr.emp;
.logoff
.exit
EOF
```

BTEQ Export 기본

- BTEQ Export 는 data(record) 모드와 report 모드, 두가지 종류의 output 모드를 가짐.
- report 모드는 일반적인 text 파일로, human readable한 포맷
- data(record) 모드로 저장되는 파일은 teradata의 loading utility 사용시에만 활용됨.
- .EXPORT RESET 을 수행해야 최종적으로 파일로 저장됨.
- 아래는 data 모드로 SQL 수행 결과를 파일로 저장.

```
.LOGON 127.0.0.1/hr_user, hr_user;
.EXPORT DATA FILE = /root/bteq_script/bteq_data01.dat
SELECT * FROM HR.EMP;
.EXPORT RESET

.LOGOFF
.EXIT
```

- report 모드로 text file export. 컬럼명은 출력 제외 할 수 없음.

```
.LOGON 127.0.0.1/hr_user, hr_user;
/* 컬럼명 밑의 Dash 제거 */
.SET TITLEDASHES OFF;

.EXPORT REPORT FILE = /root/bteq_script/bteq_report01.rpt;
SELECT * FROM HR.EMP;
.EXPORT RESET

.LOGOFF
.EXIT
```

BTEQ Export 시 기존 파일 Override하기

- EXPORT FILE이 이미 존재하는 파일명 이라면 기존 파일에 Append 형식으로 결과를 생성. 만약 위 스크립트를 재 수행할 때 bteq_report01.rpt가 이미 존재한다면 EXPORT 수행 시 결과를 bteq_report01.rpt에 append 함.
- EXPORT 시 기존 파일을 Override 하려면 CLOSE 옵션을 EXPORT에 설정(기본은 OPEN이며 이 경우 Append가 됨)

```
.LOGON 127.0.0.1/hr_user, hr_user;
/* 컬럼명 밑의 Dash 제거 */
.SET TITLEDASHES OFF;
.EXPORT REPORT FILE = /root/bteq_script/bteq_report01.rpt, CLOSE; -- OPEN으로 변경하여 테스
SELECT * FROM HR.EMP;
.EXPORT RESET

.LOGOFF
.EXIT
```

- 또는 OS 명령어로 기존 파일을 삭제하고 EXPORT 를 수행.

```
.LOGON 127.0.0.1/hr_user, hr_user;
/* 컬럼명 밑의 Dash 제거 */
.SET TITLEDASHES OFF;
.OS rm /root/bteq_script/bteq_report01.rpt
.EXPORT REPORT FILE = /root/bteq_script/bteq_report01.rpt;
SELECT * FROM HR.EMP;
.EXPORT RESET

.LOGOFF
.EXIT
```

BTEQ Export 시 출력 파일에서 컬럼명 제거하기

- Report 모드에서는 출력 파일에서 컬럼명을 제거 할 수 없음. 컬럼명을 Concat 한 뒤 하나의 컬럼으로 만든 뒤 이를 (TITLE ‘ ‘) 로 적용하여 컬럼명 출력 제거 할 수 있음. 하지만 컬럼에 NULL 값이 있는 경우에 CONCAT을 할 경우 해당 레코드 전체가 NULL로 변환되기 때문에 해당 레코드가 Export 되지 않는 문제가 발생.

```
.LOGON 127.0.0.1/hr_user, hr_user;
.SET TITLEDASHES OFF;
.OS rm /root/bteq_script/exp_no_field01.rpt
.SET NULL AS '';
.EXPORT REPORT FILE=/root/bteq_script/exp_no_field01.rpt;

SELECT empno||', '||ename||', '||job||', '||mgr||', '||hiredate (TITLE '') FROM HR.EMP;

.EXPORT RESET

.LOGOFF;
.EXIT;
```

- OS 명령어를 이용하여 출력 파일에서 맨 처음 라인을 제거. tail -n +2 ‘파일명’ 은 맨 처음 라인을 제거하고 출력. 아래 내용을 exp_no_file02.bteq 로 저장하고 실행.

```
.LOGON 127.0.0.1/hr_user, hr_user;
.SET TITLEDASHES OFF;
.OS rm /root/bteq_script/exp_no_field02.rpt
.SET NULL AS '';
.EXPORT REPORT FILE=/root/bteq_script/exp_no_field02.tmp, CLOSE;
SELECT * FROM HR.EMP;
.EXPORT RESET

.OS tail -n +2 /root/bteq_script/exp_no_field02.tmp > /root/bteq_script/exp_no_field02.r
```

```
.LOGOFF;
.EXIT;
```

BTEQ Export 시 Seperator 설정으로 CSV 파일 생성.

- .SET SEPERATOR 로 컬럼 단위 분리 Seperator를 기본 공백 2자리에서 다른 문자로 지정할 수 있음. 콤마(,)를 SEPERATOR로 지정하여 CSV 파일로 Export 결과 출력.
- 아래 파일을 exp_csv01.bteq로 설정하여 수행.

```
.LOGON 127.0.0.1/hr_user, hr_user;
.SET TITLEDASHES OFF;
.OS rm /root/bteq_script/exp_csv01.rpt
/* NULL 문자를 공백으로 치환 */
.SET NULL AS '';
.SET SEPARATOR ',';

.EXPORT REPORT FILE=/root/bteq_script/exp_csv01.tmp, CLOSE;
SELECT * FROM HR.EMP;
.EXPORT RESET

.OS tail -n +2 /root/bteq_script/exp_csv01.tmp > /root/bteq_script/exp_csv01.rpt

.LOGOFF;
.EXIT;
```

BTEQ에서 Macro 사용하기

- Macro는 SQL의 코드 블럭 재활용성을 높이기 위해서 활용됨. 동적으로 SQL등을 만들기 위해서 자주 사용됨.
- 반복적인 BTEQ의 환경 설정과 SQL 수행등을 Macro에서 정의하여 호출
- Macro는 Execute(또는 Exec) Macro명; 으로 호출함.
- 아래와 같이 Bteq의 출력 포맷과 SQL 문을 Macro로 생성. .SET WIDTH 200; 를 설정하고 SQL 수행.

```
DROP MACRO deptdisplay;

-- REPLACE MACRO 사용하면 기존 MACRO가 있을 시 REPLACE, 없으면 CREATE
CREATE MACRO deptdisplay AS
( ECHO '.SET WIDTH 200';
SELECT * FROM HR.DEPT;
);

EXECUTE DEPTDISPLAY;
```

- Macro는 인자가 있는 Macro로 작성하여 주로 SQL에 동적 파라미터를 부여하여 자주 사용.

```
DROP MACRO SELECTIVE_EMP;

-- 인자가 있는 Macro 생성.
CREATE MACRO SELECTIVE_EMP(VAR_DEPTNO INTEGER) AS
(
SELECT * FROM HR.EMP WHERE DEPTNO = :VAR_DEPTNO;
);
```

```
-- 특정 인자값으로 Macro 호출
EXECUTE SELECTIVE_EMP(30);
```

실습 - 01

- 요구사항
  - 특정 DB에 있는 모든 테이블들의 DDL을 파일로 생성. DDL 생성은 SELECT REQUESTTEXT FROM DBC.TABLESV 이용
  - BTEQ 를 Shell Script에서 동적으로 인자를 받아서 수행하는 방식으로 수행.
  - DB명은 shell script에서 동적으로 인자로 받아서 생성. bteq의 수행 결과인 DDL 파일은 별도의 디렉토리에 ddl_result_${DB_NAME}.sql 파일로 저장
  - login, format들은 별도의 파일로 설정하여 .RUN FILE로 수행.

- 아래와 같이 수행 결과 파일들을 저장할 bteq_result 디렉토리 생성

```
mkdir bteq_result
```

- logon.run 파일에 아래 저장.

```
.LOGON 127.0.0.1/hr_user, hr_user;
```

- format.run 파일을 생성하여 아래 format 설정을 저장.

```
.SET WIDTH 2000;
.SET ECHOREQ OFF;
```

- exercise_01.sh 를 생성하여 아래 내용을 입력.

```
BASE_DIR="/root/bteq_script/"
TGT_DIR="/root/bteq_result/"
DB_NAME=$1

bteq <<EOF
.RUN FILE = '${BASE_DIR}logon.run'
.RUN FILE = '${BASE_DIR}format.run'

.EXPORT REPORT FILE='${TGT_DIR}ddl_result_${DB_NAME}.sql', CLOSE;
SELECT REQUESTTEXT (TITLE '') FROM DBC.TABLESV WHERE DATABASENAME = '${DB_NAME}';
.EXPORT RESET;

.LOGOFF;
.EXIT;

EOF
```

- exercise_01.sh 를 수행하고 결과 확인

```
chmod +x exercise_01.sh
./exercise_01.sh HR
```

실습 - 02

- 요구 사항.
  - 특정 파일로 지정된 SQL을 수행하고 이 SQL의 수행 시간을 시작시간, 종료시간으로 한줄로 표시하여 특정 로그 파일에 저장할 수 있도록 bteq를 이용하여 shell script 작성.
  - 수행할 SQL 파일명과 결과 로그 파일은 Shell Script의 인자로 전달.
  - login, format들은 별도의 파일로 설정하여 .RUN FILE로 수행.

- emp.sql 파일을 생성하고 아래 SQL을 입력

```
SELECT * FROM EMP;
```

- 아래 내용을 exercise_bteq_02.sh 파일에 저장하고 수행.

```
BASE_DIR="/root/bteq_script/"
TGT_DIR="/root/bteq_result/"
sql_file=$1
log_file=$2
start_time=`date +"%Y-%m-%d %H:%M:%S"`

bteq <<EOF
.RUN FILE = '${BASE_DIR}logon.run'
.RUN FILE = '${BASE_DIR}format.run'

.EXPORT REPORT FILE='${TGT_DIR}${log_file}', CLOSE;
.RUN FILE='${BASE_DIR}${sql_file}';
.EXPORT RESET;

.LOGOFF;
.EXIT;

EOF

end_time=`date +"%Y-%m-%d %H:%M:%S"`

echo "start time: ${start_time}, end time: ${end_time}" >> ${TGT_DIR}${log_file}
```

- 아래와 같이 emp.sql의 수행 결과를 result02.log로 출력할 수 있도록 exercise_02.sh 수행.

```
chmod +x exercise*.sh
./exercise_bteq_02.sh emp.sql result02.log
```

실습 - 03

- 요구 사항.
  - SQL을 수행하되, Where 조건값을 Shell 에서 동적으로 입력 받을 수 있도록 함.
  - Macro를 이용하며, 해당 Macro는 파일로 관리하고, 파일명은 Shell에서 동적으로 입력 받을 수 있도록 함.
  - 수행할 SQL 파일명과 결과 로그 파일은 Shell Script의 인자로 전달.
  - login, format들은 별도의 파일로 설정하여 .RUN FILE로 수행.

- emp_macro.sql 파일을 생성하고 아래 SQL을 입력

```
REPLACE MACRO SELECTIVE_EMP(VAR_DEPTNO INTEGER) AS
(
SELECT * FROM HR.EMP WHERE DEPTNO = :VAR_DEPTNO;
);
```

- 아래를 exercise_03.sh에 저장

```
BASE_DIR="/root/bteq_script/"
TGT_DIR="/root/bteq_result/"
sql_file=$1
var_deptno=$2
log_file=$3

bteq <<EOF
.RUN FILE = '${BASE_DIR}logon.run'
.RUN FILE = '${BASE_DIR}format.run'

.EXPORT REPORT FILE='${TGT_DIR}${log_file}', CLOSE;
.RUN FILE='${BASE_DIR}${sql_file}';
EXECUTE SELECTIVE_EMP(${var_deptno});
.EXPORT RESET;

.LOGOFF;
.EXIT;

EOF
```

- 아래와 같이 deptno가 30인 결과값을 result_03.log로 출력 할 수 있도록 exercise_03.sh 를 수행

```
chmod +x exer*.sh
./exercise_03.sh emp_macro.sql 30 result_03.log
```