

# Oracle SQL\*Plus 명령어 실습2

빅 데이터 마이닝 연구실  
컴퓨터공학과  
한양대학교

# 목차

- SQL\*Plus 명령어
- SQL\*Plus 스키마 정의
- 형식 명령어 실습
- 환경 명령어 실습
- 기타 명령어 실습
- 실습
- 과제 1

# SQL\*Plus 명령어

구분	명령어		내용
환경명령어	SET	ARRAYSIZE {n}	<b>SQL</b> 실행 시 한번 패치 되는 데이터 열 수 제한 데이터베이스 변경 내용의 자동 저장여부 결정 <b>SQL</b> 스크립트를 실행 할 때 명령어의 출력 여부 <b>SQL</b> 문이 실행된 후 작업처리 메시지 출력 여부 결과 출력 시 컬럼 명의 화면 표시 여부 출력될 한 라인의 길이 출력 페이지 당 라인 수 화면 이동제어(한 페이지씩 보고 싶을 때) <b>NULL</b> 값을 대체할 <b>text</b> 정보를 설정할 때 출력된 컬럼 간의 여유공간을 설정할 때 컬럼의 <b>heading</b> 밑에 사용될 <b>UnderLine</b> 을 설정 컬럼들이 지정된 <b>LineSize</b> 를 초과할 때 출력여부 <b>SQL</b> 문의 수행 시간이 나타난다.
		AUTOCOMMIT	
		ECHO {off on}	
		FEED[BACK] {off on}	
		HEAD[ING]	
		LINE[SIZE] {n}	
	SHOW	PAGE[SIZE] {n}	
		PAU[SE] {on off}	
		NULL {text}	
		SPACE {n}	
		UNDERLINE {기호 on off}	
		WRAP {on off}	
		TIMING {on off}	

# SQL\*Plus 명령어 (cont'd)

구분	명령어	내용
형식 명령어	<b>COLUMN</b>	컬럼의 <b>FORMAT</b> 을 변경할 때
	<b>TTITLE</b>	보고서의 제목을 설정할 때
	<b>BTITLE</b>	보고서의 꼬리말을 설정할 때
	<b>BREAK</b>	컬럼 또는 행의 값이 바뀔 때 마다 새로운 보고서 <b>FORMAT</b> 을 설정 할 때
	<b>COMP[UTE] [function [LAB[EL] text] ... OF {expr column alias} ... ON {expr column alias REPORT  ROW} ...]</b>	<b>COMPUTE</b> 명령어는 <b>BREAK</b> 명령어에 의해 정의된 행 그룹에 대해 계산할 수 있다.

# SQL\*Plus 스키마 정의

- Alter table add : 이미 존재하는 테이블에 속성을 추가할 수 있다.
  - Alter table 테이블명 add 컬럼명 타입;

```

SQL> select * from jack;

NAME                DEPT                S_ID SEX
-----
김재익              컴퓨터공학과        11111111 남자

SQL> alter table jack add phonenum number(11);

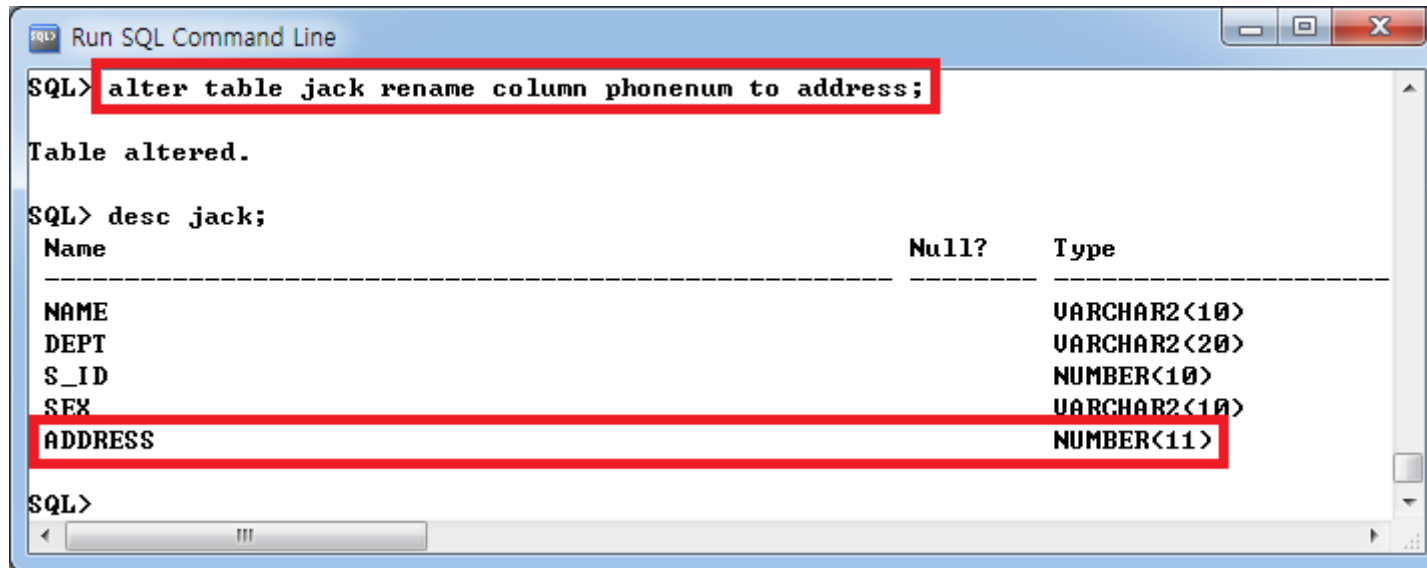
Table altered.

SQL> desc jack;

Name                Null?    Type
-----
NAME                VARCHAR2(10)
DEPT                VARCHAR2(20)
S_ID                NUMBER(10)
SEX                VARCHAR2(10)
PHONENUM          NUMBER(11)
  
```

# SQL\*Plus 스키마 정의 (cont'd)

- Alter table rename : 이미 존재하는 테이블에 컬럼 이름을 변경 할 수 있다.
  - Alter table 테이블명 rename column 컬럼명 to 컬럼명;



```
SQL> alter table jack rename column phonenum to address;
```

Table altered.

```
SQL> desc jack;
```

Name	Null?	Type
NAME		VARCHAR2(10)
DEPT		VARCHAR2(20)
S_ID		NUMBER(10)
SEX		VARCHAR2(10)
ADDRESS		NUMBER(11)

```
SQL>
```

# SQL\*Plus 스키마 정의 (cont'd)

- Update : 특정 상황에서 속성들의 모든 값을 바꾸지 않고 속성의 하나의 값만을 변경할 때 사용
  - Update 테이블명 set 컬럼명 = 값 where 찾을 조건;

The screenshot shows a SQL\*Plus window titled "Run SQL Command Line". It displays the execution of an UPDATE statement and the resulting table state.

```
SQL> select * from jack;
```

NAME	DEPT	S_ID	SEX	ADDRESS
김재익	컴퓨터공학과	11111111	남자	

```
SQL> update jack set address = 1111;
```

1 row updated.

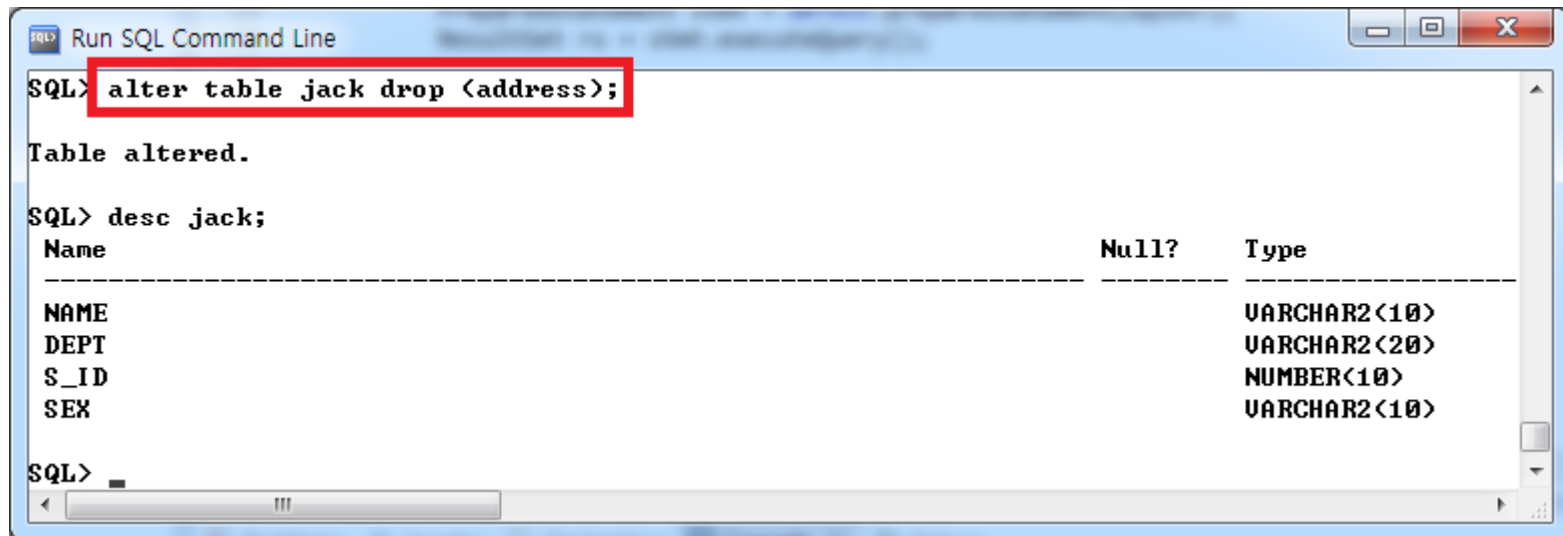
```
SQL> select * from jack;
```

NAME	DEPT	S_ID	SEX	ADDRESS
김재익	컴퓨터공학과	11111111	남자	1111

The screenshot highlights the UPDATE statement and the resulting change in the ADDRESS column of the 'jack' table.

# SQL\*Plus 스키마 정의 (cont'd)

- Alter table drop : 이미 존재하는 테이블에 속성을 삭제할 수 있다.
  - Alter table 테이블명 drop (컬럼명);

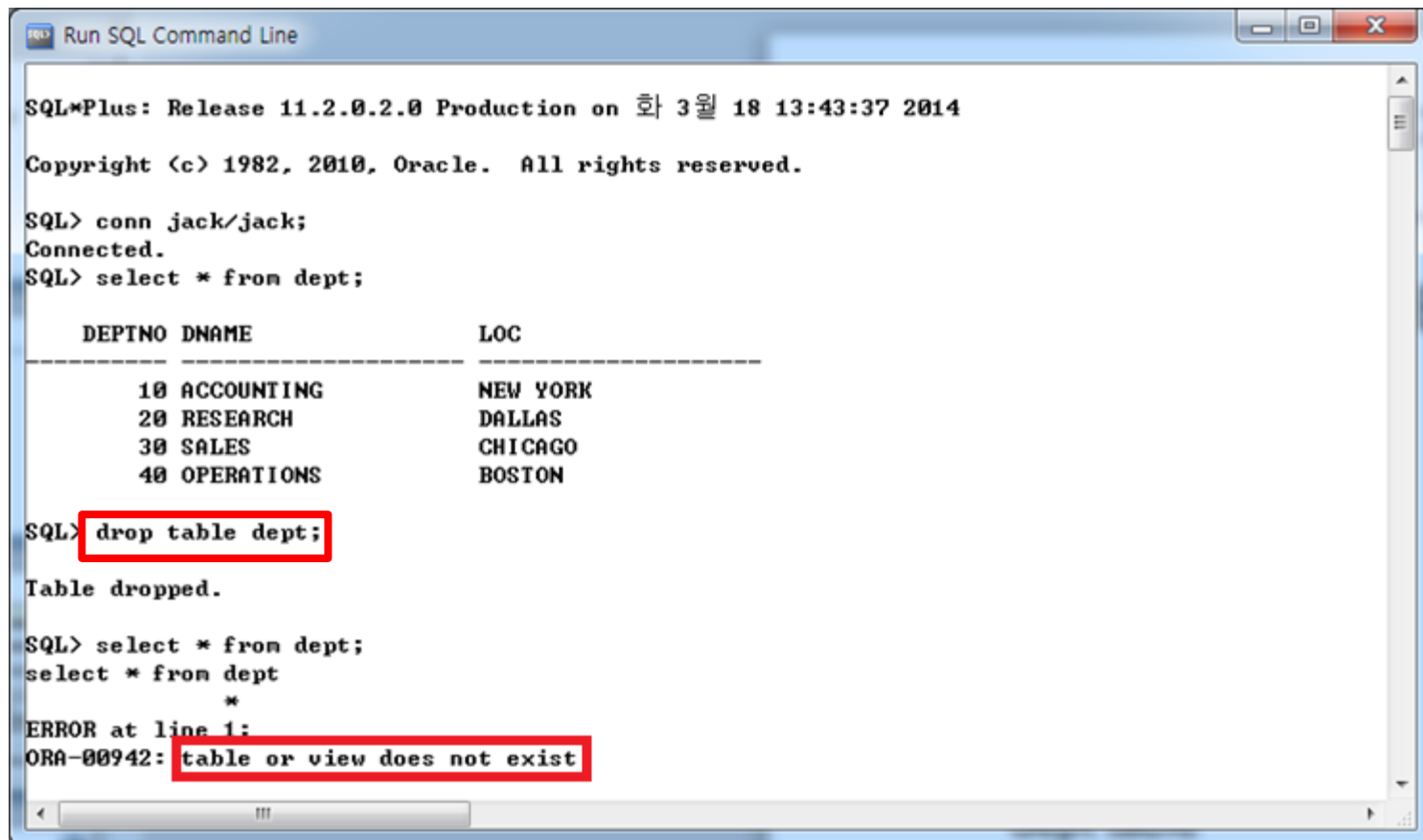


```
SQL> alter table jack drop <address>;  
Table altered.  
SQL> desc jack;  
Name Null? Type  
-----  
NAME VARCHAR2(10)  
DEPT VARCHAR2(20)  
S_ID NUMBER(10)  
SEX VARCHAR2(10)  
SQL>
```



# SQL\*Plus 스키마 정의 (cont'd)

- drop table : SQL 테이블을 제거할 수 있다.
  - drop table 테이블명;



```
Run SQL Command Line

SQL*Plus: Release 11.2.0.2.0 Production on 화 3월 18 13:43:37 2014

Copyright (c) 1982, 2010, Oracle. All rights reserved.

SQL> conn jack/jack;
Connected.
SQL> select * from dept;

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

SQL> drop table dept;

Table dropped.

SQL> select * from dept;
select * from dept
          *
ERROR at line 1:
ORA-00942: table or view does not exist
```

# 환경 명령어 실습

- Line[size] : 출력될 한 라인의 길이

```

SQL> set line 100
SQL> show line
linesize 100
SQL> select * from emp;

      EMPNO ENAME      JOB              HIREDATE          SAL          COMM          DEPTNO
-----
      7369 SMITH        CLERK              80/12/17          800              20
      7499 ALLEN        SALESMAN           81/02/02         1600             300             30
      7586 WADD         SALESMAN           91/02/23         1250             500             30
      7866 ALIS         MANAGER            83/02/22         2000              10

SQL> set line 50
SQL> select * from emp;

      EMPNO ENAME
-----
      7369 SMITH
CLERK              80/12/17          800
      20

      7499 ALLEN
SALESMAN           81/02/02         1600
      300             30
  
```

# 환경 명령어 실습 (cont'd)

- Echo : SQL 스크립트를 실행 할 때 명령어의 출력 여부
- Feed[back] : SQL문이 실행된 후 작업처리 메시지 출력 여부

```
SQL> set feed on
SQL> set echo on
SQL> show feed
FEEDBACK ON for 1 or more rows
SQL> show echo
echo ON
SQL> drop table dept;

Table dropped.

SQL> @c:\wtest\createdept;
SQL> create table dept
2 <
3         deptno         number(6),
4         dname          varchar2(10),
5         loc            varchar2(10)
6 >;

Table created.

SQL> set feed off
SQL> set echo off
SQL> drop table dept;
SQL> @c:\wtest\createdept;
SQL> @c:\wtest\deptinsert
```

The screenshot shows a 'Run SQL Command Line' window. The output of the commands is as follows: 'SQL> set feed on', 'SQL> set echo on', 'SQL> show feed' returns 'FEEDBACK ON for 1 or more rows', 'SQL> show echo' returns 'echo ON', 'SQL> drop table dept;' returns 'Table dropped.', 'SQL> @c:\wtest\createdept;' followed by 'SQL> create table dept' and its definition returns 'Table created.', 'SQL> set feed off', 'SQL> set echo off', 'SQL> drop table dept;', 'SQL> @c:\wtest\createdept;', and 'SQL> @c:\wtest\deptinsert'. A callout box labeled 'feed' points to the 'Table dropped.' message. Another callout box labeled 'echo' points to the 'Table created.' message.

# 환경 명령어 실습 (cont'd)

- Pagesize : 출력 페이지 당 라인수

```

Run SQL Command Line
SQL> show pagesize
pagesize 14
SQL> select * from emp;

  EMPNO ENAME      JOB              HIREDATE          SAL        COMM         DEPTNO
-----
    7369 SMITH      CLERK            80/12/17          800             20
    7499 ALLEN      SALESMAN         81/02/02         1600            300            30
    7586 WADD      SALESMAN         91/02/23         1250            500            30
    7866 ALIS      MANAGER          83/02/22         2000             10

4 rows selected.

SQL> set pagesize 5
SQL> select * from emp;

  EMPNO ENAME      JOB              HIREDATE          SAL        COMM         DEPTNO
-----
    7369 SMITH      CLERK            80/12/17          800             20
    7499 ALLEN      SALESMAN         81/02/02         1600            300            30

  EMPNO ENAME      JOB              HIREDATE          SAL        COMM         DEPTNO
-----
    7586 WADD      SALESMAN         91/02/23         1250            500            30
    7866 ALIS      MANAGER          83/02/22         2000             10

4 rows selected.

SQL>

```

# 환경 명령어 실습 (cont'd)

- Null : null값을 대체할 text정보를 설정할 때

```
SQL> show null
null ""
SQL> set null *****
SQL> select * from emp;
```

EMPNO	ENAME	JOB	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	80/12/17	800	*****	20
7499	ALLEN	SALESMAN	81/02/02	1600	300	30
7586	WADD	SALESMAN	91/02/23	1250	500	30
7866	ALIS	MANAGER	83/02/22	2000	*****	10

```
4 rows selected.

SQL>
```

# 환경 명령어 실습 (cont'd)

- Space : 출력된 컬럼 간의 여유공간을 설정할 때

```
SQL> show space
space 1
SQL> select * from emp;
```

EMPNO	ENAME	JOB	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	80/12/17	800	*****	20
7499	ALLEN	SALESMAN	81/02/02	1600	300	30

```
SQL> set space 2
SQL> /
```

EMPNO	ENAME	JOB	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	80/12/17	800	*****	20
7499	ALLEN	SALESMAN	81/02/02	1600	300	30

```
SQL>
```

# 환경 명령어 실습 (cont'd)

- Wrap : 컬럼들이 지정된 LineSize를 초과할 때 출력여부

```

Run SQL Command Line
SQL> set space 3
SQL> select * from emp;

  EMPNO  ENAME      JOB              HIREDATE          Salary          COMM
-----
DEPTNO
-----
   7369   SMITH      CLERK            80/12/17          $800.00          *****
   7499   ALLEN      SALESMAN         81/02/02          $1,600.00          300
   7586   WADD      SALESMAN         91/02/23          $1,250.00          500
   7866   ALIS      MANAGER          83/02/22          $2,000.00          *****

  EMPNO  ENAME      JOB              HIREDATE          Salary          COMM
-----
DEPTNO
-----
   10

SQL> show wrap
wrap : lines will be wrapped
SQL> set wrap off
SQL> /
truncating (as requested) before column DEPTNO

  EMPNO  ENAME      JOB              HIREDATE          Salary          COMM
-----
   7369   SMITH      CLERK            80/12/17          $800.00          *****
   7499   ALLEN      SALESMAN         81/02/02          $1,600.00          300
   7586   WADD      SALESMAN         91/02/23          $1,250.00          500
   7866   ALIS      MANAGER          83/02/22          $2,000.00          *****

SQL> set wrap on
SQL>

```

# 환경 명령어 실습 (cont'd)

- Timing : SQL 문의 수행 시간이 나타난다.

```

SQL> set space 3
SQL> show space
space 3
SQL> set space 1
SQL> show timing
timing OFF
SQL> set timing on
SQL> select * from emp;

```

EMPNO	ENAME	JOB	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	80/12/17	800	*****	20
7499	ALLEN	SALESMAN	81/02/02	1600	300	30
7586	WADD	SALESMAN	91/02/23	1250	500	30
7866	ALIS	MANAGER	83/02/22	2000	*****	10

```

Elapsed: 00:00:00.00
SQL> set timing off
SQL>

```



# 환경 명령어 실습 (cont'd)

- Head : 결과 출력 시 컬럼 명의 화면 표시 여부

```
SQL> show head
heading ON
SQL> select * from emp;
```

EMPNO	ENAME	JOB	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	80/12/17	800	*****	20
7499	ALLEN	SALESMAN	81/02/02	1600	300	30

```
SQL> set head off
SQL> /
```

7369	SMITH	CLERK	80/12/17	800	*****	20
7499	ALLEN	SALESMAN	81/02/02	1600	300	30
7586	WADD	SALESMAN	91/02/23	1250	500	30
7866	ALIS	MANAGER	83/02/22	2000	*****	10

```
SQL> set head on
SQL>
```

# 형식 명령어 실습

- Ttitle : 보고서의 제목을 설정할 때
- Btitle : 보고서의 꼬리말을 설정할 때

```

SQL> set pagesize 15
SQL> ttitle Empolyee_Report
SQL> btitle Hanyang-University
SQL> select * from emp;

```

화 3월 18 page 1

Empolyee\_Report

EMPNO	ENAME	JOB	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	80/12/17	800	*****	20
7499	ALLEN	SALESMAN	81/02/02	1600	300	30
7586	WADD	SALESMAN	91/02/23	1250	500	30
7866	ALIS	MANAGER	83/02/22	2000	*****	10

Hanyang-University

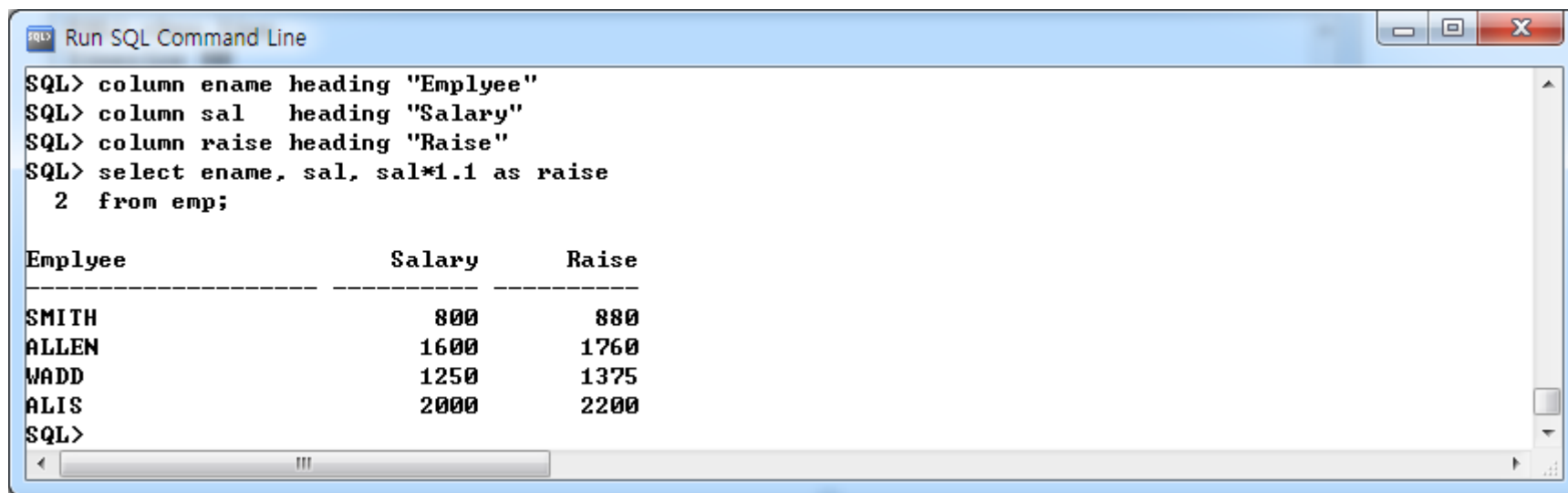
```

SQL> ttitle off
SQL> btitle off
SQL>

```

# 형식 명령어 실습 (cont'd)

- Column : 컬럼의 format을 변경할때 사용. (heading)
- As : 컬럼을 출력할때 자신이 입력한 값으로 출력하게 함.



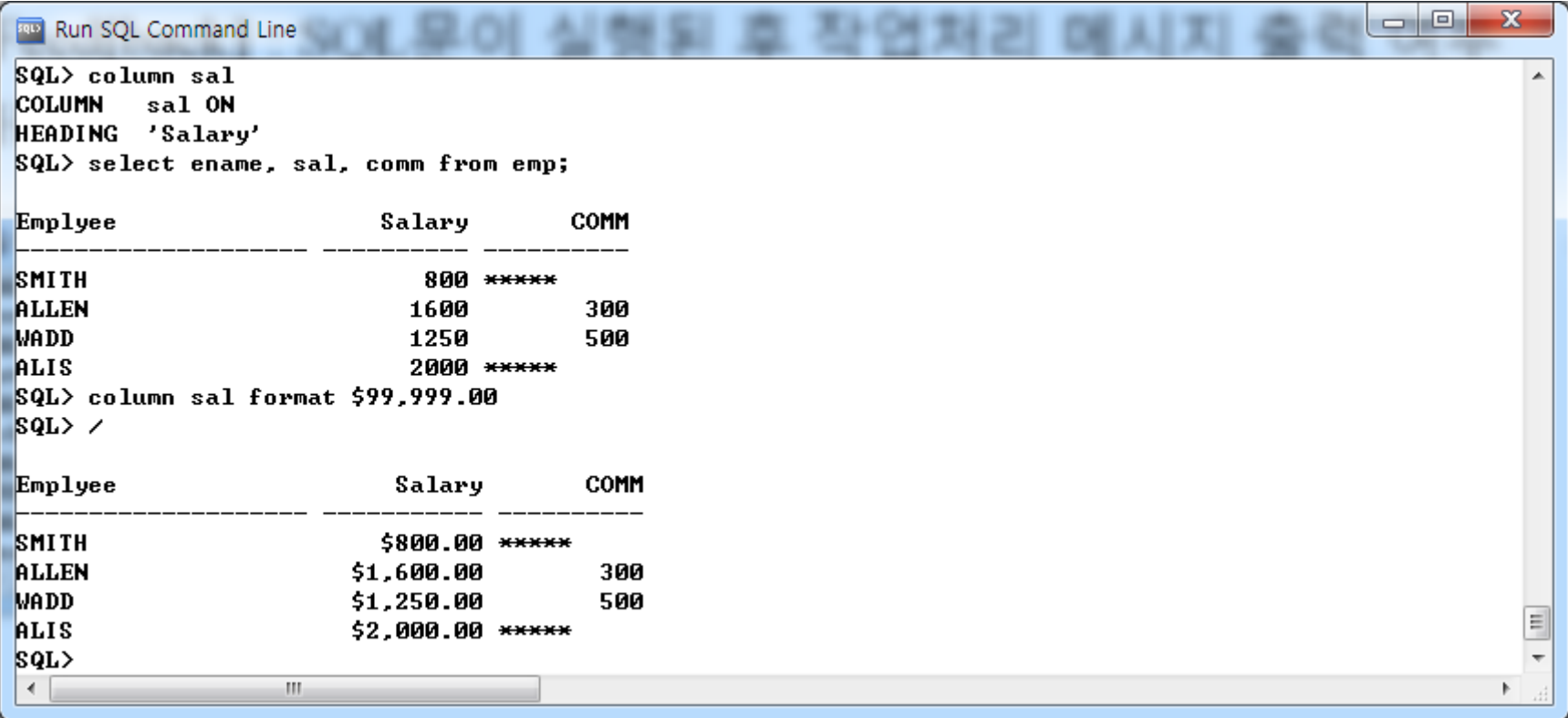
```
SQL> column ename heading "Emplyee"
SQL> column sal heading "Salary"
SQL> column raise heading "Raise"
SQL> select ename, sal, sal*1.1 as raise
  2  from emp;
```

Emplyee	Salary	Raise
SMITH	800	880
ALLEN	1600	1760
WADD	1250	1375
ALIS	2000	2200

The screenshot shows a 'Run SQL Command Line' window. It contains a series of SQL commands to format columns and execute a query. The output is a table with three columns: 'Emplyee', 'Salary', and 'Raise'. The data rows are SMITH, ALLEN, WADD, and ALIS. The 'Raise' column is calculated as 'Salary \* 1.1'.

# 형식 명령어 실습 (cont'd)

- Column : 컬럼의 format을 변경할때 사용. (format)



```
SQL> column sal
COLUMN      sal ON
HEADING     'Salary'
SQL> select ename, sal, comm from emp;

Employee                Salary          COMM
-----
SMITH                    800 *****
ALLEN                    1600           300
MADD                     1250           500
ALIS                     2000 *****
SQL> column sal format $99,999.00
SQL> /

Employee                Salary          COMM
-----
SMITH                    $800.00 *****
ALLEN                    $1,600.00       300
MADD                     $1,250.00       500
ALIS                     $2,000.00 *****
SQL>
```

# 형식 명령어 실습 (cont'd)

- Column : 컬럼의 format을 변경할때 사용. (clear)

```

SQL> select ename, sal, comm from emp;

Emplye      Salary      COMM
-----
SMITH      $800.00 *****
ALLEN     $1,600.00      300
WADD      $1,250.00      500
ALIS      $2,000.00 *****
SQL> column ename format A4
SQL> /

Emp1      Salary      COMM
-----
SMIT      $800.00 *****
H
ALLE     $1,600.00      300
N
WADD     $1,250.00      500
ALIS     $2,000.00 *****
SQL> column ename
COLUMN      ename ON
HEADING    'Employee'
FORMAT     A4
SQL> column ename clear
SQL> /

ENAME      Salary      COMM
-----
SMITH      $800.00 *****
ALLEN     $1,600.00      300
WADD      $1,250.00      500
ALIS      $2,000.00 *****
SQL>
  
```

# 형식 명령어 실습 (cont'd)

- Column : 컬럼의 format을 변경할때 사용 (clear).

```

SQL> column sal
COLUMN      sal ON
HEADING     'Salary'
FORMAT      $99,999.00
SQL> column sal off
SQL> select ename, sal, comm from emp;

ENAME                SAL          COMM
-----
SMITH                 800      *****
ALLEN                1600           300
MADD                 1250           500
ALIS                 2000      *****
SQL> column sal on
SQL> select ename, sal, comm from emp;

ENAME                Salary          COMM
-----
SMITH                 $800.00      *****
ALLEN                $1,600.00           300
MADD                 $1,250.00           500
ALIS                 $2,000.00      *****
SQL> clear columns
columns cleared
SQL>
  
```

# 형식 명령어 실습 (cont'd)

- Break : 컬럼 또는 행의 값이 바뀔 때 마다 새로운 보고서 format을 설정

```

SQL> break on deptno
SQL> select deptno, ename, sal
2  from emp
3  order by deptno;

```

DEPTNO	ENAME	SAL
10	ALIS	2000
20	SMITH	800
30	WADD	1250
	ALLEN	1600

```

SQL> break on deptno skip 1
SQL> /

```

DEPTNO	ENAME	SAL
10	ALIS	2000
20	SMITH	800
30	WADD	1250
	ALLEN	1600

```

SQL> clear break
breaks cleared
SQL>

```

# 형식 명령어 실습 (cont'd)

- Compute : break 명령어에 의해 정의된 행 그룹에 대해 계산할 수 있다.

```

SQL> break on deptno skip 1
SQL> compute sum of sal on deptno
SQL> select deptno, job, ename, sal
  2  from emp
  3  order by deptno, job;

```

DEPTNO	JOB	ENAME	SAL
10	MANAGER	ALIS	2000
*****			
	sum		2000
20	CLERK	SMITH	800
*****			
	sum		800
30	SALESMAN	WADD	1250
	SALESMAN	ALLEN	1600
*****			
	sum		2850

```

SQL> break
break on deptno skip 1 nodup

SQL> clear break
breaks cleared
SQL>

```



# 형식 명령어 실습 (cont'd)

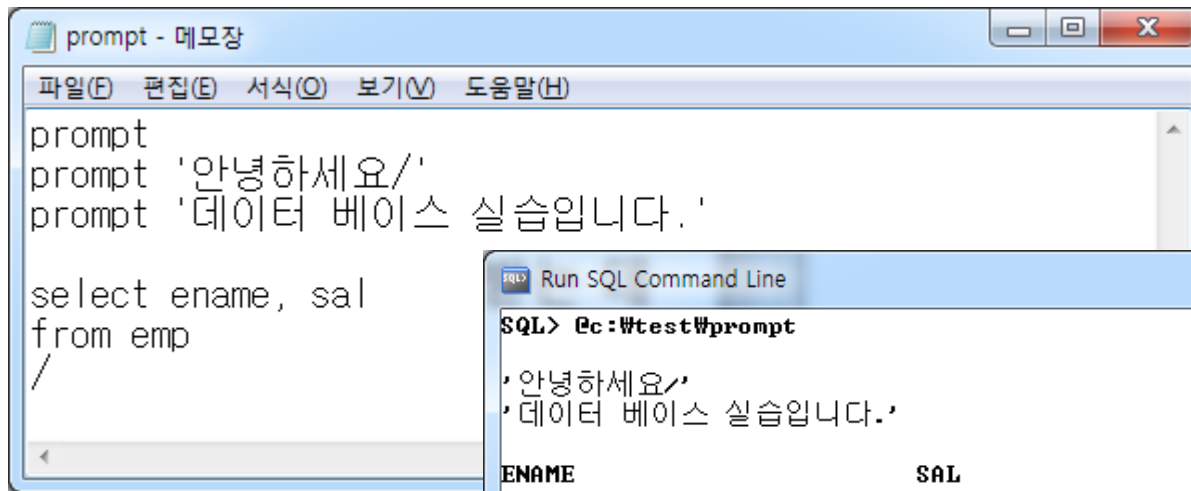
Function	Effect
<b>SUM</b>	<b>column</b> 에 있는 값의 합을 계산한다.
<b>MINIMUM</b>	<b>column</b> 에 있는 값 중 최소값을 계산한다.
<b>MAXIMUM</b>	<b>column</b> 에 있는 값 중 최대값을 계산한다.
<b>AVG</b>	<b>column</b> 에 있는 값의 평균을 계산한다.
<b>STD</b>	<b>column</b> 에 있는 값의 표준편차를 계산한다.
<b>VARIANCE</b>	<b>column</b> 에 있는 값의 분산을 계산한다.
<b>COUNT</b>	<b>column</b> 에 있는 값 중 <b>NULL</b> 이 아닌 값의 개수를 보여준다.
<b>NUMBER</b>	<b>column</b> 에 있는 열의 개수를 계산한다.



# 기타 명령어 실습 (cont'd)

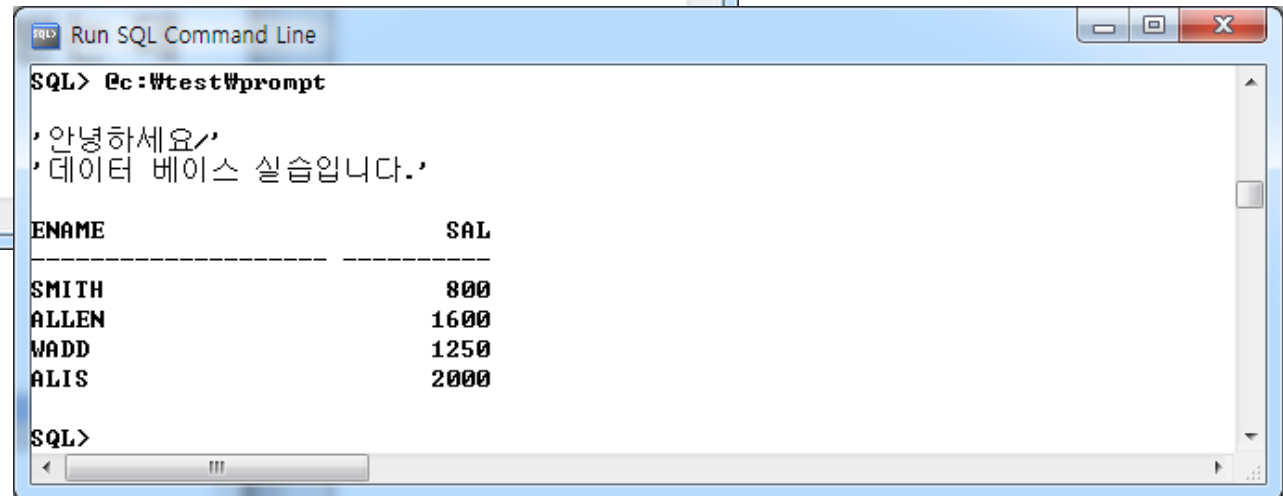
- PROMPT

- 한 라인의 데이터를 화면에 출력하는 기능을 가지고 있으며, 저장된 스크립트 내에서 사용자와 정보를 주고 받는 데 유용하게 이용된다.



```
prompt
prompt '안녕하세요/'
prompt '데이터 베이스 실습입니다.'

select ename, sal
from emp
/
```



```
SQL> @c:\wtest\prompt

'안녕하세요/'
'데이터 베이스 실습입니다.'
```

ENAME	SAL
SMITH	800
ALLEN	1600
WARD	1250
ALIS	2000

```
SQL>
```

# 기타 명령어 실습 (cont'd)

- ACCEPT

- 사용자가 입력한 데이터를 변수에 할당하는 방법으로 스크립트를 실행하는 사용자와 스크립트 간의 상호 작용을 가능하게 해준다.

```
ACC[EPT]  variable [ NUM[BER] | CHAR | DATE ]  
           [ FOR[MAT] format ]  
           [ DEFAULT default ]  
           [ PROMPT text | NOPR[OMPT] ]  
           [ HIDE ]
```

# 기타 명령어 실습 (cont'd)

**accept - 메모장**

```

prompt dept테이블 정보를 입력하시오
prompt
accept    my_deptno    number    format    '9999'    prompt 'my_deptno : '
accept    my_dname     char      format    a10       prompt 'my_dname : '
accept    my_loc       char      prompt 'my_loc : '

insert into dept(deptno, dname, loc)
values (&my_deptno, '&my_dname', '&my_loc')
/
  
```

**Run SQL Command Line**

```

SQL> @c:\wtest\waccept;
dept테이블 정보를 입력하시오

my_deptno : 100
my_dname  : 김재익
my_loc    : 서울
old 2: values (&my_deptno, '&my_dname', '&my_loc')
new 2: values (      100, '김재익', '서울')

1 row created.

SQL> select * from dept;

DEPTNO DNAME                                LOC
-----
10 ACCOUNTING                             NEW YORK
20 RESEARCH                               DALLAS
30 SALES                                   CHICAGO
40 OPERATIONS                             BOSTON
100 김재익                                서울
  
```

# 실습

Oracle SQL\*Plus

파일(F) 편집(E) 검색(S) 옵션(O) 도움말(H)

SQL> @C:\WHOME

월 Jul 19

페이지 1

**EMPLOYEE-REPORT**

EMPNO	NAME	JOB	HIREDATE	SAL	COMM	DEPARTMENT
7369	SMITH	CLERK	80/12/17	\$800.00		20
7499	ALLEN	SALESMAN	81/02/20	\$1600.00	300	30
7499	WADD	SALESMAN	81/02/22	\$1250.00	500	30
7499	ALIS	MANAGER	91/02/23	\$2000.00		10
7499	MOON	GUITARIST	79/02/23	\$1800.00	300	10

8

\$표시  
포맷 : 9999.99

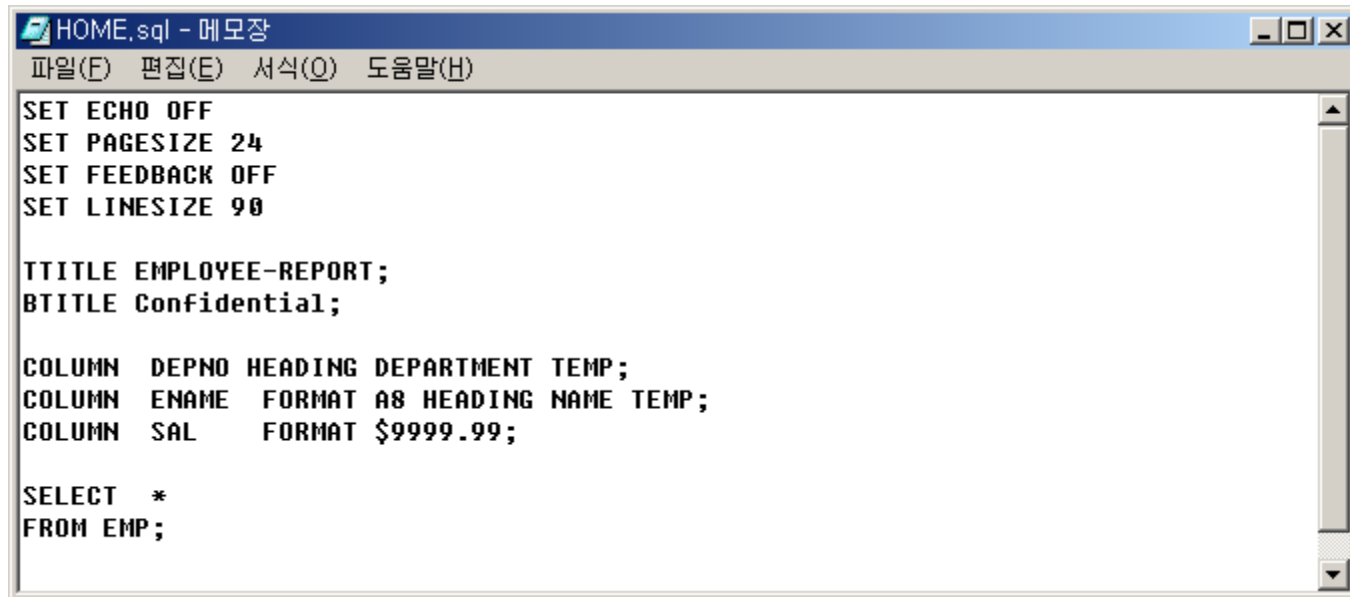
24

**Confidential**

SQL>

90

# 실습 - solution



```
HOME.sql - 메모장
파일(F)  편집(E)  서식(O)  도움말(H)

SET ECHO OFF
SET PAGESIZE 24
SET FEEDBACK OFF
SET LINESIZE 90

TTITLE EMPLOYEE-REPORT;
BTITLE Confidential;

COLUMN  DEPNO HEADING DEPARTMENT TEMP;
COLUMN  ENAME  FORMAT A8 HEADING NAME TEMP;
COLUMN  SAL    FORMAT $9999.99;

SELECT  *
FROM EMP;
```

# 과제1 용 테이블 생성, 삽입 SQL파일 작성

```
createdept.sql - 메모장
파일(F) 편집(E) 서식(O) 도움말(H)

create table dept
(
    deptno      number(6),
    dname       varchar2(10),
    loc         varchar2(10)
);
```

```
deptinsert.sql - 메모장
파일(F) 편집(E) 서식(O) 도움말(H)

insert into dept values (10,'ACCOUNTING','NEW YORK');
insert into dept values (20,'RESEARCH','DALLAS');
insert into dept values (30,'SALES','CHICAGO');
insert into dept values (40,'OPERATIONS','BOSTON');
```

```
createemp.sql - 메모장
파일(F) 편집(E) 서식(O) 도움말(H)

create table emp
(
    empno      number(6),
    ename      varchar2(10),
    job        varchar2(10),
    hiredate   date,
    sal        number(10),
    comm       number(6),
    deptno     number(6)
);
```

```
empinsert.sql - 메모장
파일(F) 편집(E) 서식(O) 도움말(H)

insert into emp values (7369,'SMITH','CLERK','80/12/17',800,NULL,20);
insert into emp values (7499,'ALLEN','SALESMAN','81/02/02',1600,300,30);
insert into emp values (7586,'WADD','SALESMAN','91/02/23',1250,500,30);
insert into emp values (7866,'ALIS','MANAGER','83/02/22',2000,NULL,10);
```



# 과제 1

- 다음의 조건을 만족하도록 Employee table에 데이터를 넣는 new\_emp라는 sql문서를 작성하시오.
  - Db table 은 다음과 같이 구성되어 있다.
    - emp(empno, ename, job, hiredate, sal, comm, deptno)
    - Empno의 format은 '9999'이다.
    - Ename의 format은 10자리의 수이다.
    - Sal의 format은 '99999.99'이다. Default 값은 '1000.00'이다.
    - Comm의 format은 '9999.99'이다. Default 값은 '0'이다.
    - Hiredate의 format은 'mm/dd/yyyy'이다.
  - 각 column값을 사용자에게 입력 받기 위하여 parameter를 사용하여라.

# 과제 1 (cont'd)

제목 없음 - 메모장

파일(F) 편집(E) 서식(O) 도움말(H)

CREATE NEW EMPLOYEE RECORD

Enter the employee's information:

Last name: \_\_\_\_\_

Employee #: \_\_\_\_\_

Salary [1000]: \_\_\_\_\_

Commission % [0]: \_\_\_\_\_

Hire date (mm/dd/yyyy): \_\_\_\_\_

List of available jobs:

[ emp table 에서 job에 의해 정렬된 그리고 값의 중복이 없도록 job을 출력하여라. ]

Job: \_\_\_\_\_

List of department numbers and names:

[ dept table 에서 deptno에 의해 정렬된 deptno, dname를 출력하여라. ]

Department #: \_\_\_\_\_

[ Emp table에 값의 입력 완료 ]

마지막에 변경된 값 출력

\_\_\_\_\_ : 및 줄 친 장소에서 값을 입력 받도록 한다.

# 과제 1 (cont'd)

- Distinct : 질의에 대한 결과에서 중복된 값을 제거하여 출력한다.
  - select distinct (컬럼명) from 테이블명;
- Order by : 질의에 대한 결과에서 해당 컬럼의 순서대로 출력한다.
  - select 컬럼명 from 테이블명 order by 컬럼명 [asc|desc];

Run SQL Command Line

```
SQL> SELECT DISTINCT <deptno> FROM emp;
```

DEPTNO
30
20
10

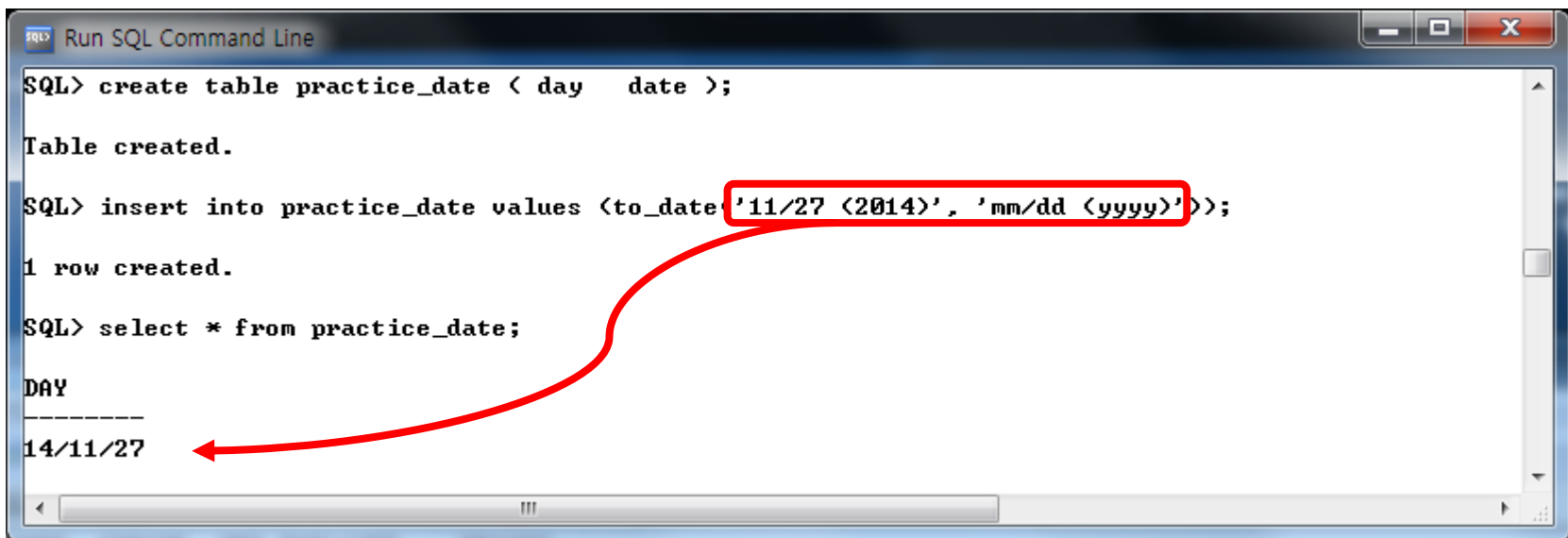
SQL> SELECT \* FROM emp order by empno asc;

EMPNO	ENAME	JOB	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	80/12/17	800		20
7499	ALLEN	SALESMAN	81/02/02	1600	300	30
7586	WADD	SALESMAN	91/02/23	1250	500	30
7866	ALIS	MANAGER	83/02/22	2000		10

asc : 오름차순  
desc : 내림차순

# 과제 1 (cont'd)

- To\_date : 임의의 날짜표현을 정규형식의 날짜표현으로 변경시켜 준다.
  - to\_date('바꾸고 싶은 날짜표현', '임의의 날짜표현 형식')



```
SQL> create table practice_date ( day    date );
Table created.
SQL> insert into practice_date values (to_date('11/27 <2014>', 'mm/dd <yyyy>'));
1 row created.
SQL> select * from practice_date;
DAY
-----
14/11/27
```

# 실습 과제 예시

```

SQL> @ d:\data\work_sql\practice
CREATE NEW EMPLOYEE RECORD

Enter the employee's information:

Last name: kis
Employee #: 1
Salary [1000]:
Commission % [0]:
Hire date (mm/dd/yyyy): 03/23/2014
List of available jobs:

JOB
-----
CLERK
MANAGER
SALESMAN

Job: MASTER
List of department numbers and names:

DEPTNO DNAME
-----
10 ACCOUNTING
20 RESEARCH
30 SALES
40 OPERATION

Department #: 20
old 2: values (&enum, '&ename', '&ejob', to_date('&edate', 'mm/dd/yyyy'), &esal, &ecomm, &dnum)
new 2: values (1, 'kis', 'MASTER', to_date('03/23/2014', 'mm/dd/yyyy'), 1000, 0, 20)

1 row created.

SQL>

```

# 실습 과제 예시 (cont'd)

Run SQL Command Line

```
SQL> SELECT * FROM emp;
```

EMPNO	ENAME	JOB	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	80/12/17	800		20
7499	ALLEN	SALESMAN	81/02/02	1600	300	30
7586	WADD	SALESMAN	91/02/23	1250	500	30
7866	ALIS	MANAGER	83/02/22	2000		10
1	kis	MASTER	14/03/23	1000	0	20

SQL> \_

- 위의 두 캡처 파일과 SQL문서를 파일 하나로 압축해서 제출

# 과제 1 (cont'd)

- 제출 형식
  - 화면 캡처 + 자신이 입력한 **sql** 파일 하나로 압축
  - 하나의 파일로 **HY-IN**에 과제 제출
  - 이번 주 일요일 밤 **11:59**까지 제출
  - 기간을 잘 확인하고 꼭 지켜주시길 바랍니다.