



05강.oracle

- 정렬 ORDER BY
 - ASC, DESC
 - 숫자함수

1 정렬을 위한 order by 절

Asc, desc

- ❖ 정렬이란 크기 순서대로 나열하는 것을 의미합니다.
- ❖ 오름차순(ascending) 정렬 방식
 - 작은 것이 위에 출력되고 아래로 갈수록 큰 값이 출력
- ❖ 내림차순(descending) 정렬 방식이
 - 큰 값이 위에 출력되고 아래로 갈수록 작은 값이 출력
- ❖ 로우를 정렬하기 위해서는 SELECT 문에 ORDER BY 절을 추가하고 어떤 컬럼을 기준으로 어떤 정렬을 할 것인지를 결정해야 합니다.

	ASC(오름차순)	DESC(내림차순)
숫자	작은 값부터 정렬	큰 값부터 정렬
문자	사전 순서로 정렬	사전 반대 순서로 정렬
날짜	빠른 날짜 순서로 정렬	늦은 날짜 순서로 정렬
NULL	가장 마지막에 나온다.	가장 먼저 나온다.

1 정렬을 위한 order by 절

오름차순 asc

- ❖ 오름차순 정렬은 작은 값부터 큰 값으로 정렬하는 것을 의미합니다. (예: 1~9, 'A' ~ 'Z') 이를 위해서는 ASC를 컬럼 다음에 기술해야 하는데 만일 생략하게 되면 디폴트로 ASC로 지정되어 있기 때문에 오름차순으로 출력됩니다.
- ❖ 다음은 급여 컬럼을 기준으로 오름차순으로 정렬한 예입니다.

예

```
SELECT *  
FROM EMP  
ORDER BY SAL ASC;
```

- ❖ 정렬방식을 지정하지 않은 경우에는 디폴트로 오름차순으로 정렬합니다.

예

```
SELECT *  
FROM EMP  
ORDER BY SAL;
```

아래와 같이 출력하시오.

9. EMP 테이블의 자료를 입사일을 오름차순으로 정렬하여 최근 입사한 직원을 먼저 출력하되 사원번호, 사원명, 직급, 입사일 칼럼을 출력하는 쿼리문을 작성하시오.

1 정렬을 위한 order by 절

내림차순 desc

- ❖ 내림차순 정렬은 큰 값부터 작은 값으로 정렬을 하는 것이다.(예:9~1, Z~A)
- ❖ 이번에는 급여를 많이 받는 사람부터 적게 받는 사람 순으로 순차적으로 출력해 봅시다.

예	<pre>SELECT * FROM EMP ORDER BY SAL DESC;</pre>
---	---

예	<pre>SELECT * FROM EMP ORDER BY ENAME;</pre>
---	--

예	<pre>SELECT * FROM EMP ORDER BY HIREDATE DESC;</pre>
---	--

1 정렬을 위한 order by 절

2개의 컬럼 검색

- ❖ 급여를 많이 받는 사람부터 적게 받는 사람 순으로 순차적으로 출력하는 결과 화면을 살펴보면 동일한 급여를 받는 사람이 존재합니다.
- ❖ 급여가 같은 사람이 존재할 경우 이름의 철자가 빠른 사람부터 출력되도록 하려면 정렬 방식을 여러 가지로 지정해야 합니다.

예	<pre>SELECT * FROM EMP ORDER BY SAL DESC, ENAME ASC;</pre>
---	--

아래와 같이 출력하시오.

11. 사원 번호가 빠른 사원부터 출력하되 같은 부서내의 사원을 출력할 경우 최근에 입사한 사원부터 출력하시오.

1 Sql 주요함수(숫자함수)

숫자함수

구 분	설 명
ABS	절대값을 구한다.
COS	COSINE 값을 반환한다.
EXP	$e(2.71828183\cdots)$ 의 n 승을 반환한다.
FLOOR	소수점 아래를 잘라낸다.(버림)
LOG	LOG값을 반환한다.
POWER	POWER(m, n) m 의 n 승을 반환한다.
SIGN	SIGN (n) $n < 0$ 이면 -1 , $n = 0$ 이면 0 , $n > 0$ 이면 1 을 반환한다.
SIN	SINE값을 반환한다.
TAN	TANGENT값을 반환한다.
ROUND	특정 자릿수에서 반올림한다.
TRUNC	특정 자릿수에서 잘라낸다. (버림)
MOD	입력 받은 수를 나눈 나머지 값을 반환한다.

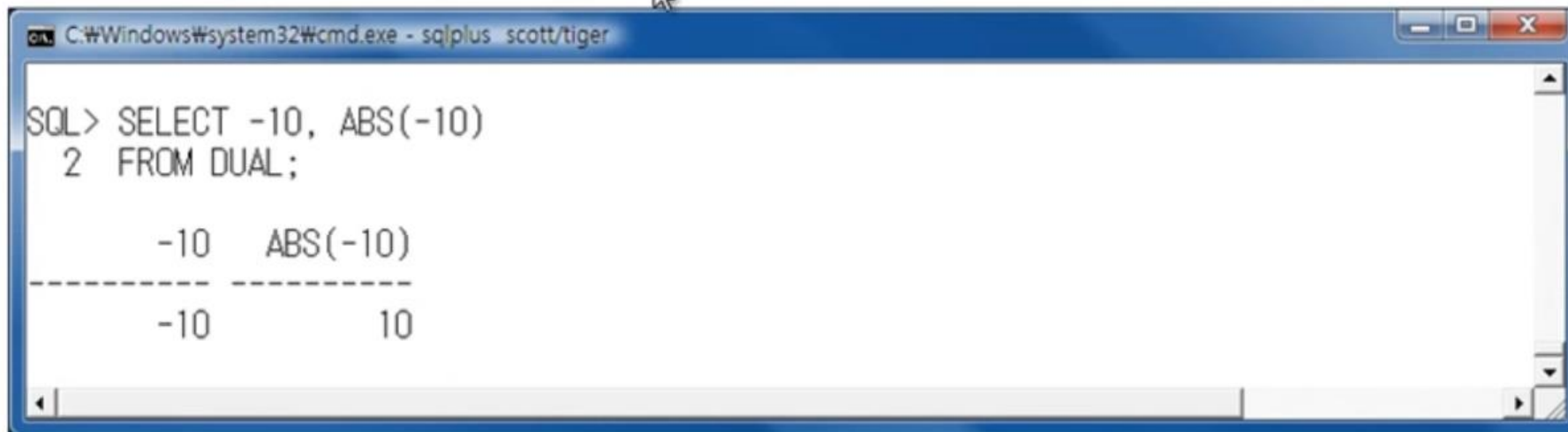
1 Sql 주요함수(숫자함수)

ABS

- ❖ ABS 함수는 절대값을 구합니다. 절대값은 방향은 없고 크기만 있는 것으로서 주어진 데이터가 음수일 경우 양수로 표현합니다.
- ❖ -10 에 대한 절대값을 구해 봅시다.

예

```
SELECT -10, ABS(-10)  
FROM DUAL;
```

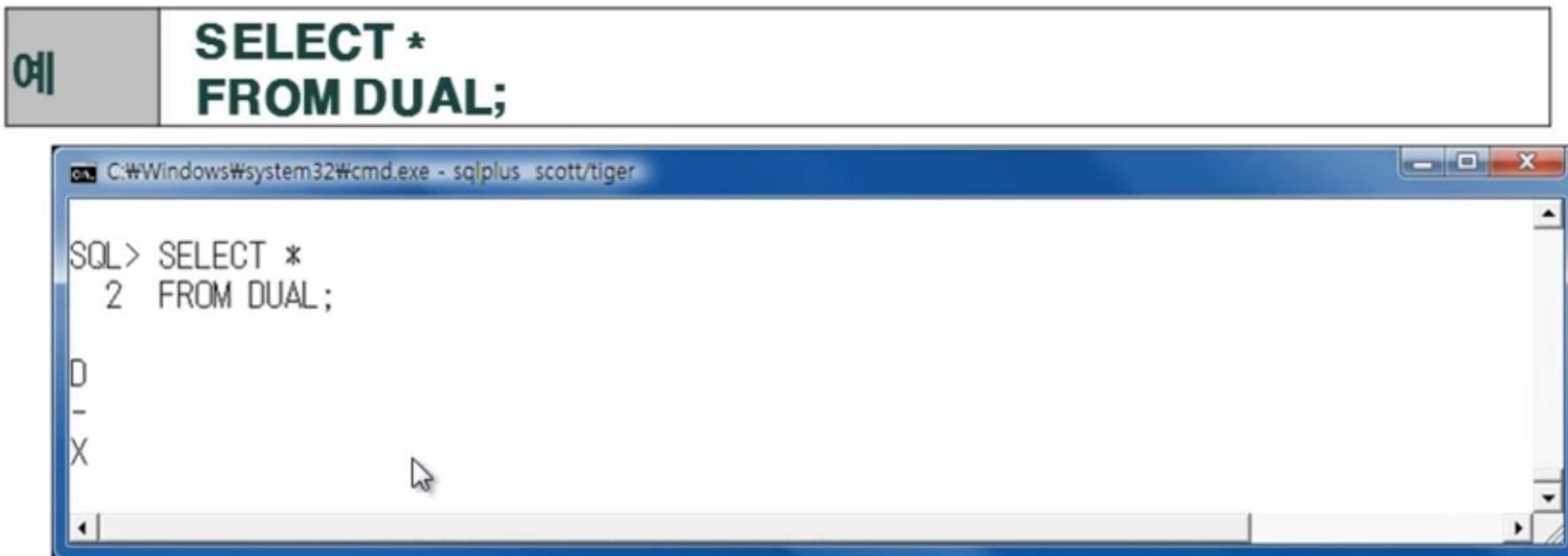


The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The prompt is "SQL>". The user has entered the command "SELECT -10, ABS(-10)" on the first line and "2 FROM DUAL;" on the second line. The output is displayed as a table with two columns: "-10" and "ABS(-10)". The first row of data shows "-10" and "10".

```
SQL> SELECT -10, ABS(-10)  
2  FROM DUAL;  
  
-10  ABS(-10)  
-----  
-10      10
```

1 Sql 주요함수(숫자함수)

DUAL



- ❖ DUAL 테이블은 DUMMY라는 단 하나의 컬럼에 X라는 단 하나의 로우만을 저장하고 있으나 이 값은 아무런 의미가 없습니다.
- ❖ 쿼리문의 수행 결과가 하나의 로우로 출력되도록 하기 위해서 단 하나의 로우를 구성하고 있을 뿐입니다.

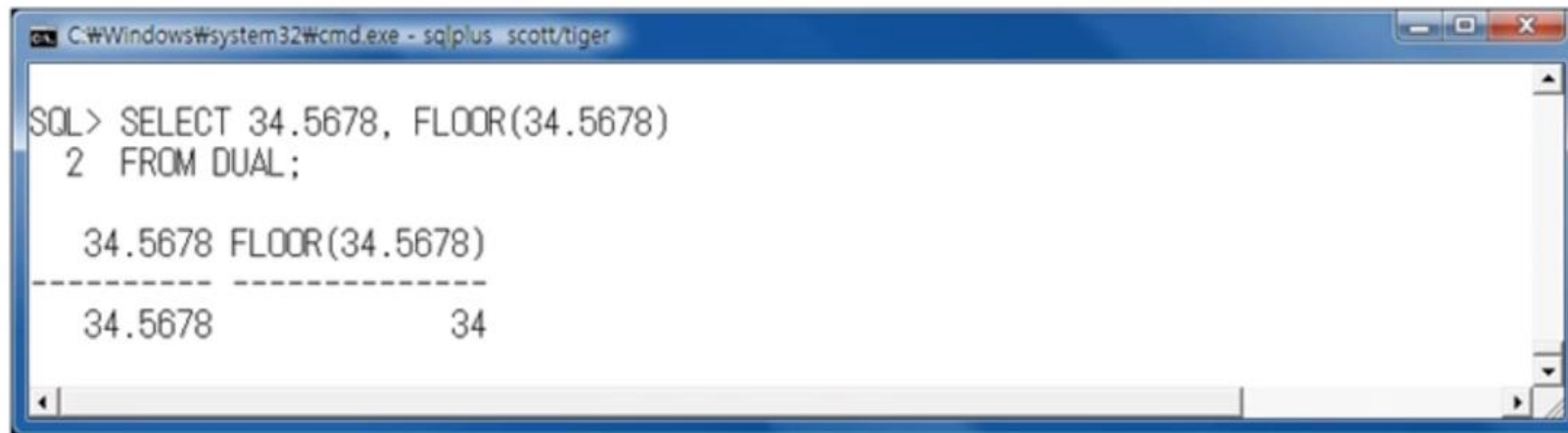
1 Sql 주요함수(숫자함수)

FLOOR - 버림

- ❖ **FLOOR 함수는 소수점 아래를 버립니다. 34.5678를 FLOOR 함수에 적용하면 34가 구해집니다.**
- ❖ **-10 에 대한 절대값을 구해 봅시다.**

예

```
SELECT 34.5678, FLOOR(34.5678)  
FROM DUAL;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The prompt is "SQL>". The user has entered the command "SELECT 34.5678, FLOOR(34.5678) FROM DUAL;". The output is displayed as follows:

```
SQL> SELECT 34.5678, FLOOR(34.5678)  
2  FROM DUAL;  
  
34.5678 FLOOR(34.5678)  
-----  
34.5678          34
```

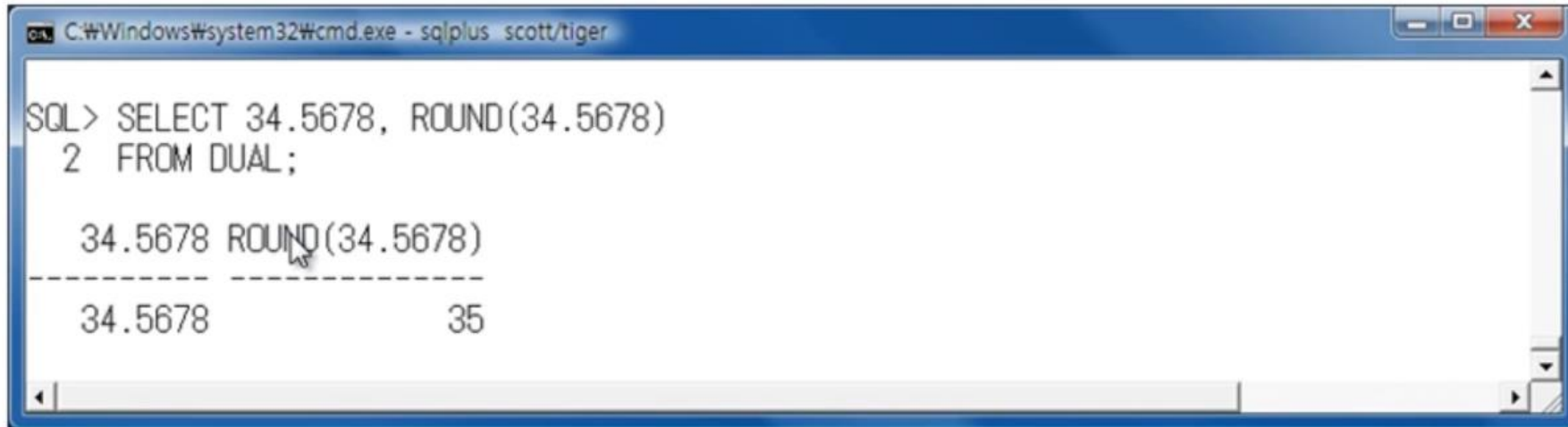
1 Sql 주요함수(숫자함수)

ROUND - 반올림

- ❖ 34.5678를 반올림하면 35입니다. 이와 같이 반올림한 결과를 구하기 위한 함수로 오라클에서는 ROUND가 제공됩니다.
- ❖ -10 에 대한 절대값을 구해 봅시다.

예

```
SELECT 34.5678, ROUND(34.5678)  
FROM DUAL;
```



```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
SQL> SELECT 34.5678, ROUND(34.5678)  
2  FROM DUAL;  
  
34.5678 ROUND(34.5678)  
-----  
34.5678                35
```

1 Sql 주요함수(숫자함수)

ROUND - 반올림

- ❖ ROUND 함수는 지정한 자릿수에서 반올림을 할 수 있도록 자릿수를 지정할 수 있습니다.

형식	ROUND(대상, 자릿수)
----	----------------

- ❖ ROUND 함수의 두 번째 인자 값이 2이면 소수점 이하 세 번째 자리에서 반올림하여 소수점 이하 두 번째 자리까지 표시합니다. 즉, ROUND(34.5678, 2)의 결과는 34.57이 됩니다.

예	SELECT 34.5678, ROUND(34.5678, 2) FROM DUAL;
---	---

- ❖ 자릿수에 음수를 지정할 수 있는데 이럴 경우 소수점 이하가 아니라 반대쪽인 일단위, 십단위, 백단위 순으로 거슬러 올라가게 되며, 인자 값이 양수일 때와는 달리 해당 자리에서 반올림이 일어나게 됩니다.

예	SELECT 34.5678, ROUND(34.5678, -1) FROM DUAL;
---	--

- ❖ ROUND(34.5678, -1)의 결과는 30이 됩니다.

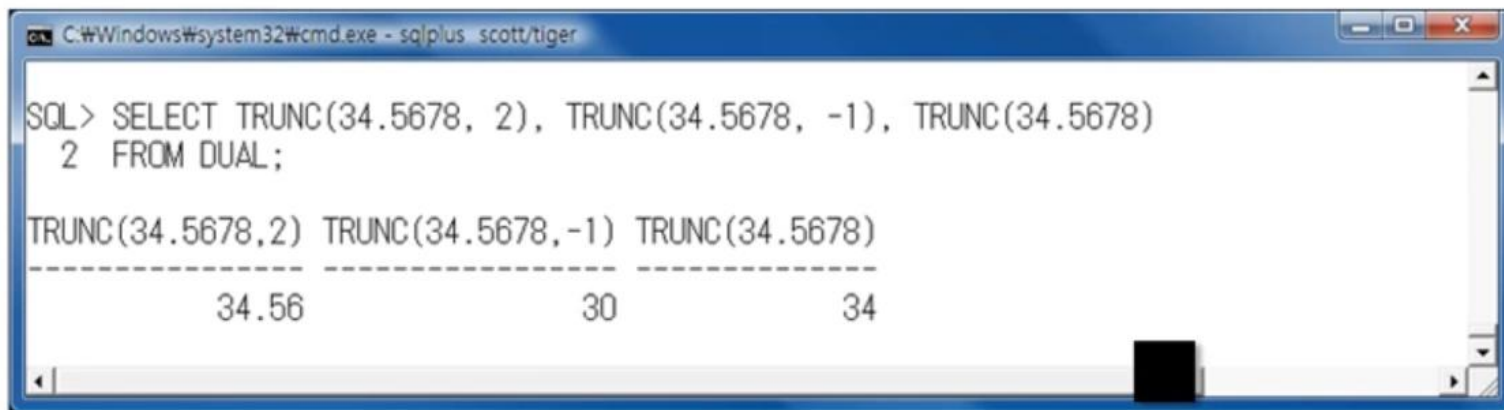
1 Sql 주요함수(숫자함수)

TRUNC - 지정자리 버림

❖ **TRUNC 함수는 지정한 자리 수 이하를 버린 결과를 구해주는 함수입니다.**

예

```
SELECT TRUNC(34.5678, 2), TRUNC(34.5678, -1), TRUNC(34.5678)
FROM DUAL;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The prompt is "SQL>". The user has entered the command: "SELECT TRUNC(34.5678, 2), TRUNC(34.5678, -1), TRUNC(34.5678) FROM DUAL;". The output is displayed in a table format with three columns: "TRUNC(34.5678,2)", "TRUNC(34.5678,-1)", and "TRUNC(34.5678)". The values are "34.56", "30", and "34" respectively. The columns are separated by dashed lines.

TRUNC(34.5678,2)	TRUNC(34.5678,-1)	TRUNC(34.5678)
34.56	30	34

- **TRUNC 함수의 두 번째 인자 값이 20이면 소수점 이하 세 번째 자리에서 버림 연산을 하여 소수점 이하 두 번째 자리까지 표시합니다.**
- **두 번째 인자 값이 0인 경우에는 소수점자리에서 버림 연산을 하고 -1인 경우는 일의 자리에서 버림 연산을 하며 두 번째 인자가 주어지지 않은 경우 0으로 간주 되어 소수점자리에서 버림 연산을 수행합니다.**

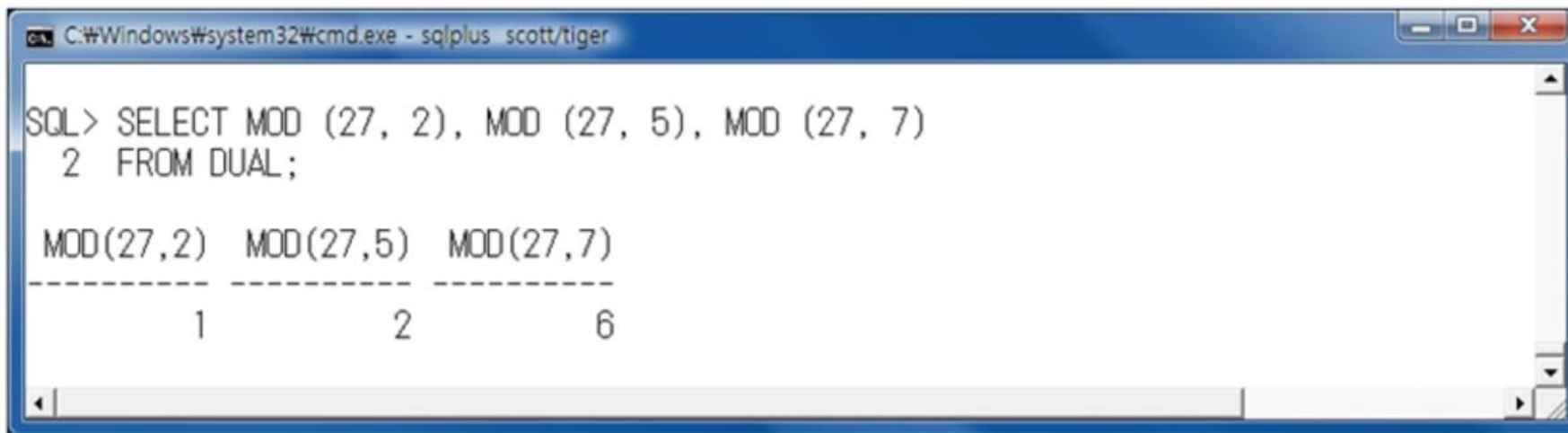
1 Sql 주요함수(숫자함수)

MOD - 나머지 결과값

❖ MOD 함수는 나누기 연산을 한 후에 구한 몫이 아닌 나머지를 결과로 되돌려주는 함수입니다.

예

```
SELECT MOD (27, 2), MOD (27, 5), MOD (27, 7)
FROM DUAL;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". Inside the window, the following SQL query is entered and executed:

```
SQL> SELECT MOD (27, 2), MOD (27, 5), MOD (27, 7)
2 FROM DUAL;
```

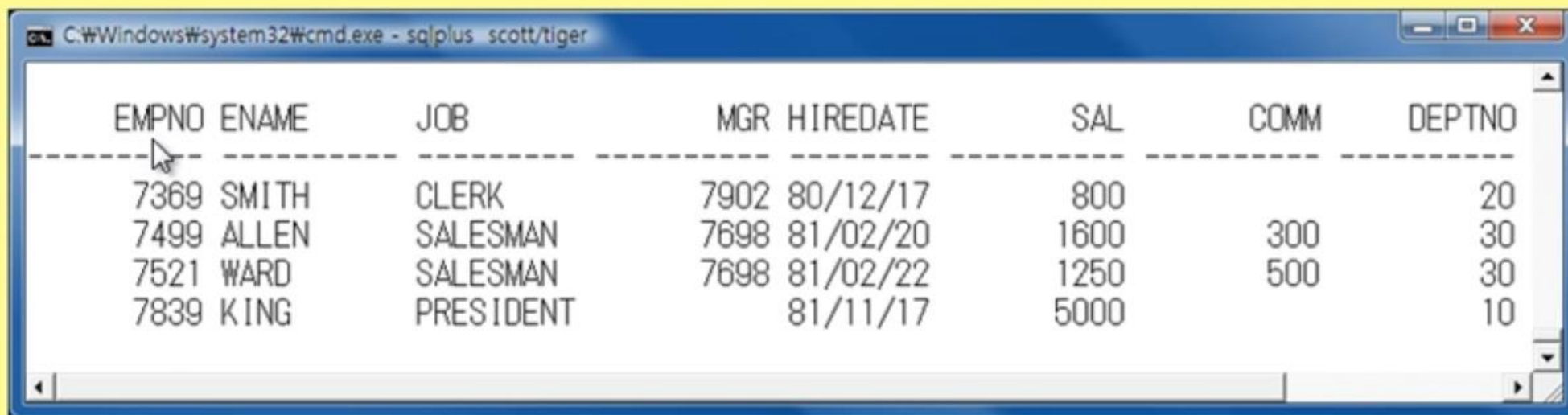
The output of the query is displayed as follows:

MOD(27,2)	MOD(27,5)	MOD(27,7)
1	2	6

1 실습.

아래와 같이 출력하시오.

1. 사번이 홀수인 사람들을 검색해 보십시오.



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7839	KING	PRESIDENT		81/11/17	5000		10

```
select empno, ename from emp where mod( empno, 2) = 1;
```


THANK YOU