

## 8. 상호작용과 대체 변수

컴퓨터공학부

김은경

## [학습 목표]

1. & 대체 변수의 용도를 설명할 수 있다.
2. && 대체 변수의 용도를 설명할 수 있다.
3. 대체 변수의 활용 방법을 나열할 수 있다.
4. DEFINE 문으로 사용자 변수를 정의할 수 있다.
5. UNDEFINE 문으로 사용자 변수의 정의를 해지시킬 수 있다.

# 1. 상호작용과 대체 변수

## ★ 상호작용(Interaction)이란?

- 사용자에게 일방적으로 결과만 표시하는 것이 아니라, 사용자로부터 값을 받아들이어서 질의문을 실행시키는 것을 의미함
- SQL 문에 대체 변수를 포함시켜 상호작용이 가능한 질의문을 작성할 수 있음

## ★ 대체 변수(Substitution Variable)란?

- 사용자가 입력한 값으로 치환될 수 있는 변수로서, 치환 변수라고도 칭함
- 대체 변수의 용도
  - iSQL\*Plus 서버에 값을 임시로 저장함
  - SQL 문장 간에 변수 값을 전달함
- 대체 변수의 종류
  - ① &(Single Ampersand) 대체 변수
  - ② &&(Double Ampersand) 대체 변수
  - ③ DEFINE과 ACCEPT 명령으로 정의한 사용자 변수

## 2. & 대체 변수

### ★ & 대체 변수의 용도

- ① 사용자로부터 값을 받아들이거나 이미 할당된 값을 활용하기 위한 것
- ② SQL 문장에서 변수를 가리킴
- ③ 사용자로부터 정보를 받아들이م

### ★ & 대체 변수의 역할

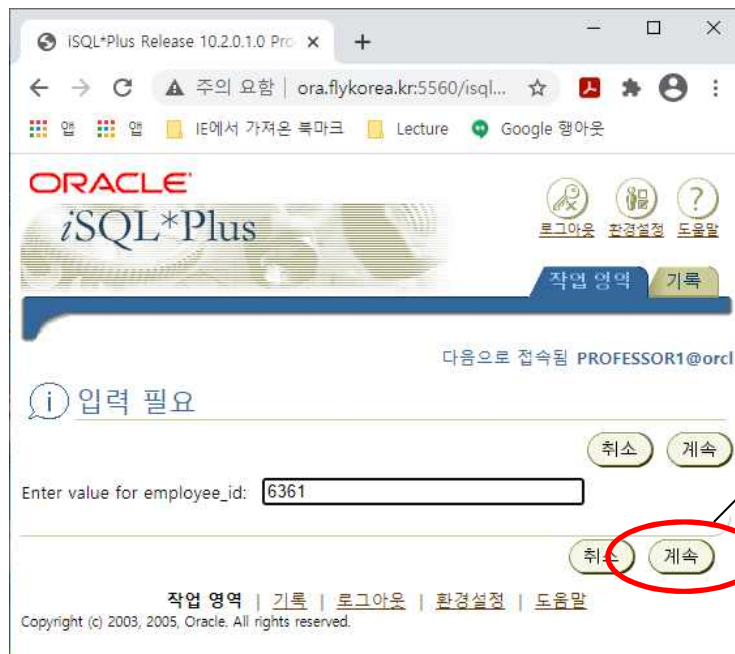
- ① SQL 문에 포함된 대체 변수에 할당된 값이 없으면, 이 대체 변수를 포함한 SQL 문장이 실행될 때마다 iSQL\*Plus가 매번 사용자에게 새로 요청함
- ② 만약 DEFINE 명령에 의해 변수에 값이 할당되어 있는 경우, 사용자에게 요청하지 않고 미리 할당된 값을 사용함
- ③ SQL 문 안에 재사용 가능한 대체 변수가 지정된 경우, 값을 받아들이지 않고 재사용함

## 2. & 대체 변수

### ✓ & 대체 변수 활용 사례(1)

- 사용자가 입력한 사원 번호에 해당하는 사원의 사원번호와 이름, 급여, 그리고 부서번호를 검색하시오.

```
SELECT emp_id, emp_name, sal, dept_id
FROM emp
WHERE emp_id = &employee_id ;
```



## 2. & 대체 변수

### ✓ & 대체 변수 활용 사례(2)

- 300번 부서에 근무하는 사원의 번호와 이름, 사용자가 입력한 열 이름을 출력하시오.

```
SELECT emp_id, emp_name, &column_name
FROM emp
WHERE emp_id = 300;
```

i 입력 필요

취소

계속

Enter value for column\_name:

old 1: SELECT emp\_id, emp\_name, &column\_name  
new 1: SELECT emp\_id, emp\_name, hiredate

EMP_ID	EMP_NAME	HIREDATE
6351	진대준	20/05/31
7910	이영희	20/09/01
7920	김마리아	20/12/03

### 3. SET VERIFY 명령어

#### ★ SET VERIFY 명령어의 용도

- 대체 변수를 사용자가 입력한 값으로 대체하는 내용을 화면에 표시하도록, 시스템 변수인 VERIFY를 ON(설정) 또는 OFF(해지) 시키는 SQL\*Plus 명령어

SET VERIFY **ON | OFF**

#### ★ iSQL\*Plus에서의 설정

- iSQL\*Plus에서는 디폴트로 대체하는 내용을 화면에 표시하도록 VERIFY가 ON되어 있음

old 1: SELECT emp\_id, emp\_name, &column\_name

new 1: SELECT emp\_id, emp\_name, hiredate

EMP_ID	EMP_NAME	HIREDATE
6351	진대준	20/05/31
7910	이영희	20/09/01
7920	김마리아	20/12/03

VERIFY가 ON 된 상태이므로 & 대체 변수가 대체되는 내용을 표시함

## 4. 문자 및 날짜 대체 변수

### ★ 문자 및 날짜 대체 변수란?

- 문자 및 날짜 값을 대체하는 변수

### ★ 문자 및 날짜 대체 변수 사용 방법

- ① 반드시 **단일 인용부호**로 묶어서 표시해야 함
- ② 문자 대체 변수의 경우, UPPER, LOWER, INITCAP과 같은 문자 조작 함수와 함께 사용할 수 있음



## 4. 문자 및 날짜 대체 변수

### ✓ 대소문자 구분이 필요 없는 문자 대체 변수 사용 사례

- emp 테이블에서 사용자가 입력한 업무를 수행하는 모든 사원의 사원번호와 급여, 입사일을 검색하는 질의문을 실행시켜서 업무로 '세일즈' 입력하시오.

```
SELECT emp_id, sal, hiredate
FROM emp
WHERE job = '&job_title' ;
```

i 입력 필요

취소

계속

Enter value for job\_title: 세일즈

old 3: WHERE job = '&job\_title'

new 3: WHERE job = '세일즈'

EMP_ID	SAL	HIREDATE
7489	3700	19/02/20
7522	4520	18/02/22
7854	3500	20/09/08

## 4. 문자 및 날짜 대체 변수

### ✓ 대소문자 구분이 필요한 문자 대체 변수 사용 사례

- foreign\_emp 테이블에서 사용자가 영문자로 입력한 업무를 수행하는 모든 사원의 사원 번호와 업무, 급여를 검색하는 질의문을 실행시켜서, 업무로 'engineer'와 'Engineer'를 각각 입력하시오.

```
SELECT emp_id, job, sal
FROM foreign_emp
WHERE job = '&job_title' ;
```

i 입력 필요

Enter value for job\_title:

old 3: WHERE job = '&job\_title'  
new 3: WHERE job = 'engineer'  
no rows selected

i 입력 필요

Enter value for job\_title:

old 3: WHERE job = '&job\_title'  
new 3: WHERE job = 'Engineer'

EMP_ID	JOB	SAL
8211	Engineer	3200
8212	Engineer	3000
8213	Engineer	2900

## 4. 문자 및 날짜 대체 변수

### ✓ 날짜 대체 변수 사용 사례

- 사용자가 입력한 날짜에 입사한 모든 사원의 이름과 업무, 입사일을 검색하는 질의문을 입사일로 '2021년 1월 23일' 입력하시오.

```
SELECT emp_id, job, hiredate
FROM emp
WHERE hiredate = '&hired_date' ;
```

 입력 필요

Enter value for hired\_date:

old 3: WHERE hiredate = '&hired\_date'

new 3: WHERE hiredate = '21/01/23'

EMP_ID	JOB	HIREDATE
7933	사무직	21/01/23

## 5. && 대체 변수

### ★ && 대체 변수의 용도

- ① SQL문 안에서 대체 변수를 재사용하기 위한 것
- ② 동일한 질의문에서 같은 값이 두 번 사용될 때, 사용자에게 매번 요구하지 않고 이미 받아들인 변수 값을 재사용하기 위해서 사용함

## 5. && 대체 변수

### ✓ && 대체 변수 사용 예제

- 200번 부서의 사원 이름과 업무 및 사용자가 입력한 열을 출력하되, 사용자가 입력한 열을 기준으로 내림차순으로 정렬해서 출력하시오.

```
SELECT emp_name, job, &&column_name
FROM emp
WHERE dept_id = 200
ORDER BY &column_name DESC;
```

&&column\_name에 대체된 값이  
같은 이름을 갖는 대체 변수인  
&column\_name에서 재사용됨

i 입력 필요

취소

계속

Enter value for column\_name:

old 1: SELECT emp\_name, job, &&column\_name

new 1: SELECT emp\_name, job, hiredate

old 4: ORDER BY &column\_name DESC

new 4: ORDER BY hiredate DESC

EMP_NAME	JOB	HIREDATE
김철수	사무직	21/01/23
백기수	연구직	20/12/09
김철수	부장	20/06/09
박철수	대표이사	17/12/17

## 6. 대체 변수의 활용

### ★ 대체 변수 활용 방법

- 사용자와 상호작용 할 수 있는 SQL 문을 작성하기 위해서 다음과 같이 활용함
  - ① SELECT 절의 열 이름 대체
  - ② WHERE 절의 검색 제한 조건 대체
  - ③ FROM 절의 테이블 이름 대체
  - ④ ORDER BY 절의 열 이름 대체
  - ⑤ 전체 SELECT 문장에서 여러 개의 대체 변수를 사용할 수 있음

## 6. 대체 변수의 활용

### ✓ SELECT 절의 열 이름 대체 예제

- 300번 부서에서 사원번호와 함께 사용자가 입력한 특정 열 검색하시오.

```
SELECT emp_id, &컬럼명
FROM emp
WHERE dept_id = 300;
```

 입력 필요

Enter value for 컬럼명:

old 1: SELECT emp\_id, &컬럼명  
new 1: SELECT emp\_id, sal


EMP_ID	SAL
6351	5850
7910	3200
7920	2650

## 6. 대체 변수의 활용

### ✓ SELECT 절에서 2개의 열 이름 대체 예제

- 200번 부서에서 사원이름과 함께 사용자가 입력한 2개의 열 검색하시오.

```
SELECT emp_id, &column_1, &column_2
FROM emp
WHERE dept_id = 200;
```

 입력 필요

Enter value for column\_1:

Enter value for column\_2:

old 1: SELECT emp\_id, &column\_1, &column\_2  
new 1: SELECT emp\_id, sal, hiredate

EMP_ID	SAL	HIREDATE
6200	9000	17/12/17
6361	6500	20/06/09
7878	5000	20/12/09
7933	2500	21/01/23



## 6. 대체 변수의 활용

### ✓ WHERE 절의 조건 대체 예제

- 사용자가 입력한 조건을 만족하는 사원의 번호와 이름, 입사일을 출력하는 질의문을 실행시켜서, '부서번호가 100'이라는 조건과 '업무가 연구직'이라는 조건을 각각 입력하시오.

```
SELECT emp_id, emp_name, job
FROM emp
WHERE &condition;
```

i 입력 필요

Enter value for condition: dept\_id = 100

old 3: WHERE &condition  
new 3: WHERE dept\_id = 100

EMP_ID	EMP_NAME	JOB
6311	김대정	부장
7872	이문정	사무직

## 6. 대체 변수의 활용

### ✓ WHERE 절의 조건 대체 예제

- 사용자가 입력한 조건을 만족하는 사원의 번호와 이름, 입사일을 출력하는 질의문을 실행시켜서, '부서번호가 100'이라는 조건과 '업무가 연구직'이라는 조건을 각각 입력하시오.

```
SELECT emp_id, emp_name, job
FROM emp
WHERE &condition;
```

i 입력 필요

Enter value for condition: job = '연구직'

old 3: WHERE &condition  
new 3: WHERE job = '연구직'

EMP_ID	EMP_NAME	JOB
7878	백기수	연구직
7901	정동길	연구직

## 6. 대체 변수의 활용

### ✓ FROM 절의 테이블 이름 대체 예제

- 사용자가 입력한 테이블에서 사원의 번호와 이름, 업무를 출력하는 질의문을 실행해서, 테이블 이름으로 foreign emp 입력하시오.

```
SELECT  *  
FROM    &table_name;
```

 입력 필요

Enter value for table\_name:

old 2: FROM &table\_name

new 2: FROM dept

DEPT_ID	DEPT_NAME	DEPT_LOC
100	관리부	서울
200	연구부	대전
300	총무부	서울
400	영업부	인천
500	개발부	천안

## 6. 대체 변수의 활용

### ✓ ORDER BY 절의 열 이름 대체 예제

- 400번 부서 사원의 이름과 급여, 입사일을 출력하되, 사용자가 입력한 열 이름을 기준으로 오름차순으로 정렬해서 출력하는 질의문을 실행해서, 정렬의 기준이 되는 열 이름으로 입사일을 입력하기

```
SELECT    emp_name, sal, hiredate
FROM      emp
WHERE     dept_id = 400
ORDER BY  &order_column;
```

 입력 필요

Enter value for order\_column:

old 4: ORDER BY &order\_column

new 4: ORDER BY hiredate

EMP_NAME	SAL	HIREDATE
정종철	4520	18/02/22
민동규	3700	19/02/20
이종길	7000	19/04/02
진영진	3500	20/09/08

## 7. 사용자 변수

### ★ 사용자 변수(User Variable)란?

- 사용자가 임의로 정의해서 사용하는 변수를 의미하며, 대체 변수의 값으로 대체될 수 있음

### ★ 사용자 변수 정의 방법

- ① **DEFINE** : CHAR 데이터 형의 사용자 변수를 생성하여 값을 할당하거나, 이미 생성된 사용자 변수의 값을 확인함
- ② **ACCEPT** : 사용자의 입력을 읽어서 사용자 변수에 저장함

### ★ 사용자 변수 해지 방법

- **UNDEFINE** : 사용자가 생성한 변수의 정의를 해지함

## 7. 사용자 변수

### ★ DEFINE 명령어

- CHAR 형의 사용자 변수를 정의하거나, 정의된 변수를 확인하는 SQL\*Plus 명령어임
- 정의하는 변수가 없으면 새로 변수를 생성하고, 변수가 있으면 재정의함

### ★ DEFINE 명령 사용 방법

유형	기능
DEFINE 변수이름 = 값	. CHAR 데이터형 변수를 생성한 다음, 값을 할당함 . 값에 공백이 포함되면, 값을 단일 인용 부호로 묶어야 함
DEFINE 변수이름	. 변수이름의 변수 값과 변수의 데이터 형을 출력함
DEFINE	. 값과 데이터 형을 가진 모든 변수를 출력함

## 7. 사용자 변수

### ✓ 사용자 변수의 정의 및 활용 예제

- 사원이름을 저장할 사용자 변수인 empname에 "Tom"이란 값을 할당하고 확인한 후, foreign\_emp 테이블에서 사원 정보를 검색할 때 그 변수를 활용하시오.

```
DEFINE empname = Tom  
DEFINE empname
```

```
DEFINE EMPNAME = "Tom" (CHAR)
```

```
SELECT emp_name, job, sal  
FROM foreign_emp  
WHERE emp_name = '&empname';
```

사용자 입력을 받지 않고 앞에서  
정의한 empname 변수의 값을  
그대로 이용함

old 3: WHERE emp\_name = '&empname'

new 3: WHERE emp\_name = 'Tom'

EMP_NAME	JOB	SAL
Tom	Engineer	3200

## 7. 사용자 변수

### ★ UNDEFINE 명령어

- DEFINE이나 ACCEPT 명령으로 정의된 변수의 정의를 해지시키는 SQL\*Plus 명령어임

UNDEFINE 변수이름

### ★ DEFINE 명령으로 정의된 변수가 해지되는 경우

- ① UNDEFINE 명령으로 정의를 해지한 경우
- ② DEFINE 명령으로 같은 이름의 변수를 재정의한 경우
- ③ iSQL\* Plus를 종료시킨 경우



## 7. 사용자 변수

### ✓ 사용자 변수의 정의 및 활용 예제

- 앞에서 정의한 empname 변수의 정의를 확인해 본 다음, 그 변수의 정의를 해지한 후, 다시 그 변수의 정의를 확인해 보시오.

```
DEFINE empname
```

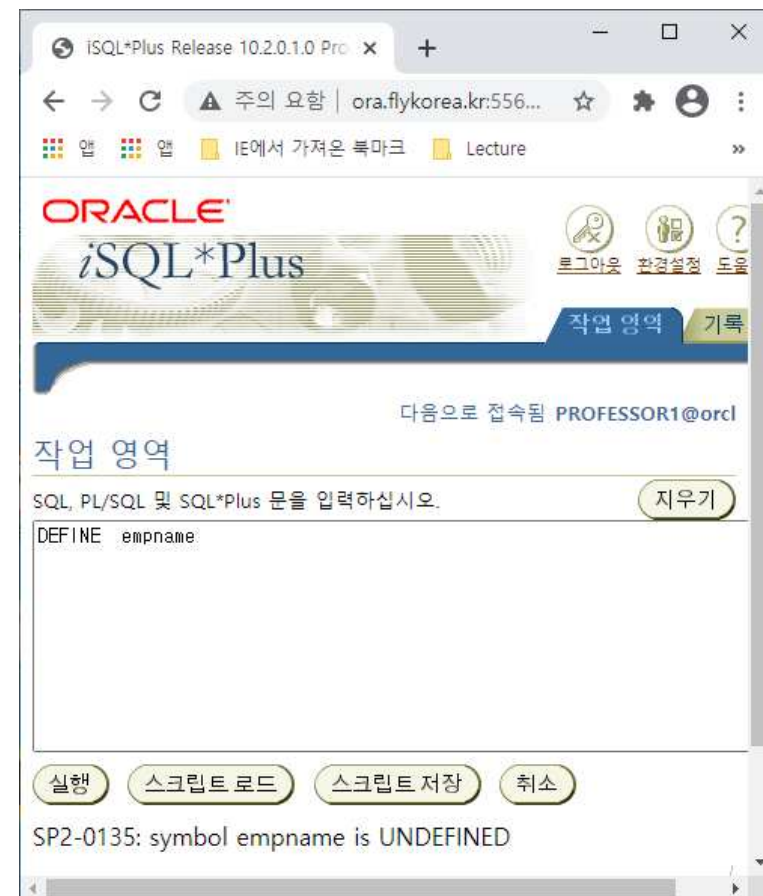
```
DEFINE EMPNAME = "Tom" (CHAR)
```

```
UNDEFINE empname
```

SP2-0863: iSQL\*Plus 프로세싱이 완료되었습니다.

```
DEFINE empname
```

SP2-0135: symbol empname is UNDEFINED



## 7. 사용자 변수

### ★ ACCEPT 명령어

- 사용자 변수를 정의함과 동시에, 사용자 입력을 받아들이기 위한 추가적인 기능을 제공하는 SQL\*Plus 명령어임

**ACCEPT 변수이름 [데이터 형] [FORMAT 포맷]  
[PROMPT '텍스트'] {HIDE}**

- . 데이터 형 : 디폴트로 CHAR 형의 변수를 정의하며, NUMBER와 DATE 형도 지정할 수 있음
- . FORMAT 포맷 : A와 9 등과 같은 포맷 요소를 이용해서 입력을 받아들일 형식을 정의함  
예) A10 : 문자 10자리, 9.99 : 소수점 이하가 두 자리인 숫자
- . PROMPT 텍스트 : 디폴트 프롬프트 대신 텍스트의 내용을 새로운 프롬프트로 정의함
- . HIDE : 사용자가 입력하는 내용을 숨기고, 입력한 문자 수만큼 '\*'로 표시함

## 7. 사용자 변수

### ★ ACCEPT 명령의 특징

- ① 사용자 입력을 받을 때, 디폴트 프롬프트 대신 사용자에게 표시할 프롬프트를 새로 정의할 수 있음
- ② NUMBER 또는 DATE 형 변수를 명시적으로 정의할 수 있음
- ③ 보안을 위해 사용자 입력을 숨길 수 있음

## 7. 사용자 변수

### ✓ ACCEPT 명령 사용 예시

```
ACCEPT 부서번호 PROMPT '부서 번호를 입력하시오 : '
```

```
SELECT *  
FROM emp  
WHERE dept_id = &부서번호
```

사용자가 새로운 프롬프트를 정의함

i 입력 필요

부서 번호를 입력하시오 :

디폴트 프롬프트 대신  
사용자가 정의한 프롬프트를 표시함

old 3: WHERE dept\_id = &부서번호

new 3: WHERE dept\_id = 100

EMP_ID	EMP_NAME	JOB	DEPT_ID	SAL	BONUS	MGR_ID	HIREDATE
6311	김대정	부장	100	6200		6200	18/12/17
7872	이문정	사무직	100	2500		6311	21/01/12

## 7. 사용자 변수

### ✓ ACCEPT 명령에서 HIDE 옵션을 추가한 경우

```
ACCEPT 사원이름 PROMPT '사원 이름을 입력하시오 : ' HIDE
SELECT *
FROM emp
WHERE emp_name = '&사원이름';
```

i 입력 필요

사원 이름을 입력하시오 : .....

HIDE 옵션을 지정했으므로  
사용자가 입력한 문자 수  
만큼 '\*'를 표시함

old 3: WHERE emp\_name = '&사원이름'

new 3: WHERE emp\_name = '김철수'

EMP_ID	EMP_NAME	JOB	DEPT_ID	SAL	BONUS	MGR_ID	HIREDATE
6361	김철수	부장	200	6500		6200	20/06/09
7933	김철수	사무직	200	2500		6361	21/01/23



**Q & A**