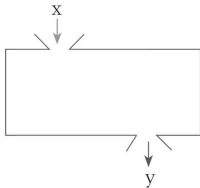


## Chap06. 데이터 처리와 가공을 위한 오라클 함수

## 06-1 오라클 함수

---

### “ 함수



### “ 오라클 함수의 종류

“ 내장 함수(built-in function)

“ 사용자 정의 함수(user-defined function)

## 06-1 오라클 함수

### “ 내장 함수의 종류

#### “ 단일행 함수(single-row function)

열 1	열 2	...	열 N
행 1			
행 2			
...			
행 N			



열 1	열 2	...	열 N
행 1			
행 2			
...			
행 N			

열 1	열 2	...	열 N
행 1			
행 2			
...			
행 N			



열 1	열 2	...	열 N
행 1			

## 06-2 문자 데이터를 가공하는 문자 함수

---

### “ UPPER, LOWER, INITCAP: 대소문자 변환

- “ 실습 6-1: SELECT 절 기본 사용
- “ 실습 6-2: WHERE 절에서의 활용
- “ 실습 6-3: LIKE 문과 함께 활용

### “ LENGTH: 문자열 길이

- “ 실습 6-4: SELECT 절 기본 사용
- “ 실습 6-5: WHERE 절에서의 활용



## 06-2 문자 데이터를 가공하는 문자 함수

### “ SUBSTR: 문자열 일부 추출

함수	설명
SUBSTR(문자열 데이터, 시작 위치, 추출 길이)	문자열 데이터의 시작 위치부터 추출 길이만큼 추출합니다. 시작 위치가 음수일 경우에는 마지막 위치부터 거슬러 올라간 위치에서 시작합니다.
SUBSTR(문자열 데이터, 시작 위치)	문자열 데이터의 시작 위치부터 문자열 데이터 끝까지 추출합니다. 시작 위치가 음수일 경우에는 마지막 위치부터 거슬러 올라간 위치에서 끝까지 추출합니다.

### “ INSTR : 문자열 데이터 내 특정 문자 위치 찾기

### “ REPLACE : 특정 문자를 다른 문자로 대체



## 06-2 문자 데이터를 가공하는 문자 함수

---

- “ LPAD, RPAD : 데이터의 빈 공간 채우기
- “ CONCAT : 두 문자열 데이터를 합치기
- “ TRIM, LTRIM, RTRIM : 특정 문자 지우기



## 06-3 숫자 데이터를 연산하고 수치를 조정하는 숫자함수

---

“ **ROUND** : 반올림

“ **TRUNC** : 버림

“ **CEIL** : 지정된 숫자와 가장 가까운 큰 정수

“ **FLOOR** : 지정된 숫자와 가장 가까운 작은 정수

“ **MOD** : 숫자를 나눈 나머지

---



## 06-4 날짜 데이터를 다루는 날짜 함수

### “ 오라클의 DATE 데이터의 연산

연산	설명
날짜 데이터 + 숫자	날짜 데이터보다 숫자만큼 일수 이후의 날짜
날짜 데이터 - 숫자	날짜 데이터보다 숫자만큼 일수 이전의 날짜
날짜 데이터 - 날짜 데이터	두 날짜 데이터 간의 일수 차이
날짜 데이터 + 날짜 데이터	연산 불가, 지원하지 않음★

### “ **SYSDATE**: 현재 날짜와 시간





## 06-4 날짜 데이터를 다루는 날짜 함수

---

- “ **ADD\_MONTHS** : 몇 개월 이후 날짜
- “ **MONTHS\_BETWEEN** : 두 날짜 간의 개월수 차이
- “ **NEXT\_DAY** : 돌아오는 요일
- “ **LAST\_DAY** : 달의 마지막 날짜
- “ **ROUND, TRUNC** : 날짜 반올림, 버림



## 06-5 자료형을 반환하는 형 변환 함수

---

- “ TO\_CHAR: 숫자 또는 날짜 데이터를 문자 데이터로
- “ TO\_NUMBER: 문자 데이터를 숫자 데이터로
- “ TO\_DATE: 문자 데이터를 날짜 데이터로



## 06-6 NULL 처리 함수

---

- “ **NVL** : NULL이 아니면 그대로, NULL이면 지정한 값
- “ **NVL2** : NULL이 아닐때와 NULL일때 각각 지정한 값



## 06-7 상황에 따라 다른 데이터를 반환하는 DECODE 함수와 CASE문

---

### “ DECODE

- “ 기준 데이터를 지정
- “ 기준 데이터에 따라 반환할 데이터 지정

### “ CASE

- “ 기준 데이터를 지정하는 방식
- “ 기준 데이터를 지정하지 않는 방식



## Chap07. 다중행 함수와 데이터 그룹화

## 07-1 하나의 열에 출력 결과를 담는 다중행 함수

---

“ **SUM** : 합계

“ **COUNT** : 데이터 개수

“ **MAX** : 최댓값

“ **MIN** : 최솟값

“ **AVG** : 평균값

---



## 07-2 결과 값을 원하는 열로 묶어 출력하는 GROUP BY 절

---

### “ GROUP BY 절의 기본 사용법

```
SELECT    [조회할 열1 이름], [열2 이름], ..., [열N 이름]
FROM      [조회할 테이블 이름]
WHERE     [조회할 행을 선별하는 조건식]
GROUP BY  [그룹화할 열을 지정(여러 개 지정 가능)]
ORDER BY  [정렬하려는 열 지정]
```

### “ GROUP BY 절을 사용할 때 유의점

---



## 07-3 GROUP BY절에 조건을 줄 때 사용하는 HAVING절

---

### “ HAVING 절의 기본 사용법

```
SELECT    [조회할 열1 이름], [열2 이름], ..., [열N 이름]
FROM      [조회할 테이블 이름]
WHERE     [조회할 행을 선별하는 조건식]
GROUP BY  [그룹화할 열 지정(여러 개 지정 가능)]
HAVING    [출력 그룹을 제한하는 조건식]
ORDER BY  [정렬하려는 열 지정];
```

### “ HAVING절을 사용할 때 유의점

### “ WHERE절과 HAVING절의 차이점

---





## 07-4 그룹화와 관련된 여러 함수

---

- “ ROLLUP, CUBE : 그룹화 데이터의 합계
- “ GROUPING SETS : 지정한 각 열별 그룹화
- “ GROUPING : ROLLUP, CUBE와 함께 사용 (하나)
- “ GROUPING\_ID : ROLLUP, CUBE와 함께 사용 (여럿)



## 07-4 그룹화와 관련된 여러 함수

---

- “ LISTAGG : 그룹 데이터 가로 출력
- “ PIVOT, UNPIVOT : 행/열 바꾸어 출력



## Chap08. 여러 테이블을 하나의 테이블처럼 사용하는 조인

## 08-1 조인

---

“ 조인 : 가로 연결

“ 집합 연산자 : 세로 연결

“ **FROM** 절에 여러 테이블 지정

```
SELECT
```

```
FROM 테이블1, 테이블2, . . . , 테이블N
```



## 08-1 조인

“ WHERE절을 활용한 조인 조건 사용의 중요성

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1980/12/17	800		20
7499	ALLEN	SALESMAN	7698	1981/02/20	1600	300	30
7521	WARD	SALESMAN	7698	1981/02/22	1250	500	30
7566	JONES	MANAGER	7839	1981/04/02	2975		20
7654	MARTIN	SALESMAN	7698	1981/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	1981/05/01	2850		30
7782	CLARK	MANAGER	7839	1981/06/09	2450		10
7788	SCOTT	ANALYST	7566	1987/04/19	3000		20
7839	KING	PRESIDENT		1981/11/17	5000		10
7844	TURNER	SALESMAN	7698	1981/09/08	1500	0	30
7876	ADAMS	CLERK	7788	1987/05/23	1100		20
7900	JAMES	CLERK	7698	1981/12/03	950		30
7902	FORD	ANALYST	7566	1981/12/03	3000		20
7934	MILLER	CLERK	7782	1982/01/23	1300		10

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

“ 테이블의 별칭 설정

## 08-2 조인 종류

---

“ **등가 조인** : '=' 사용

“ **비등가 조인** : 등가 조인 외

“ **자체 조인** : 하나의 테이블을 여러 테이블처럼 사용

“ **외부 조인** : 조인 조건의 **NULL** 데이터도 출력

왼쪽 외부 조인(Left Outer Join)	WHERE TABLE1.COL1 = TABLE2.COL1(+)
오른쪽 외부 조인(Right Outer Join)	WHERE TABLE1.COL1(+) = TABLE2.COL1



## 08-3 SQL-99 표준 문법으로 배우는 조인

---

- “ NATURAL JOIN : 두 테이블의 같은 열 기준
- “ JOIN ~ USING : 조인에 사용할 기준열 명시
- “ JOIN ~ ON : 조인 조건 직접 명시



## 08-3 SQL-99 표준 문법으로 배우는 조인

### “ OUTER JOIN : 외부 조인

왼쪽 외부 조인 (Left Outer Join)	기존	WHERE TABLE1.COL1 = TABLE2.COL1(+)
	SQL-99	FROM TABLE1 LEFT OUTER JOIN TABLE2 ON (조인 조건식)
오른쪽 외부 조인 (Right Outer Join)	기존	WHERE TABLE1.COL1(+) = TABLE2.COL1
	SQL-99	FROM TABLE1 RIGHT OUTER JOIN TABLE2 ON (조인 조건식)
전체 외부 조인 (Full Outer Join)	기존	기본 문법은 없음 (UNION 집합 연산자를 활용)
	SQL-99	FROM TABLE1 FULL OUTER JOIN TABLE2 ON (조인 조건식)

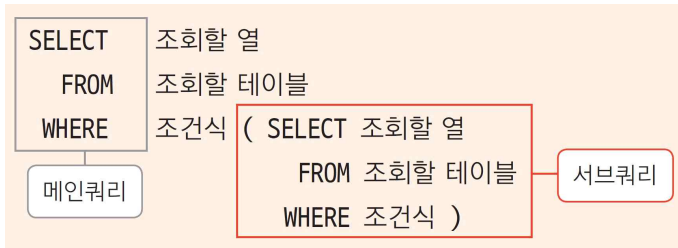




## Chap09. SQL문 속 또 다른 SQL문, 서브쿼리

## 09-1 서브쿼리

“ 서브쿼리란?



## 09-2 실행 결과가 하나인 단일행 서브쿼리

---

“ 단일행 서브쿼리

“ 실행 결과가 단 하나의 행으로 나오는 서브쿼리

“ 단일행 연산자 사용

단일행 연산자							
>	>=	=	<=	<	<>	^=	!=
초과	이상	같음	이하	미만	같지 않음		



## 09-3 실행 결과가 여러 개인 다중행 서브쿼리

---

### “ 다중행 서브쿼리

“ 실행 결과가 여러 개의 행으로 나오는 서브쿼리

“ 다중행 연산자 사용

다중행 연산자	설명
IN	메인쿼리의 데이터가 서브쿼리의 결과 중 하나라도 일치한 데이터가 있다면 true
ANY, SOME	메인쿼리의 조건식을 만족하는 서브쿼리의 결과가 하나 이상이면 true
ALL	메인쿼리의 조건식을 서브쿼리의 결과 모두가 만족하면 true
EXISTS	서브쿼리의 결과가 존재하면(즉, 행이 1개 이상일 경우) true



## 09-3 실행 결과가 여러 개인 다중행 서브쿼리

---

“ **IN** 연산자

“ 서브쿼리의 결과와 일치하는 값을 가진 행을 출력

“ **WHERE**절에 서브쿼리 없을 때의 **IN** 연산자와 동일한  
효과



## 09-3 실행 결과가 여러 개인 다중행 서브쿼리

---

“ ANY, SOME 연산자

“ 서브쿼리의 여러 결과 값에 메인쿼리의 조건식을 대입

“ 조건식을 대입한 결과가 하나라도 **true**일 경우,  
메인쿼리의 조건식이 **true**



## 09-3 실행 결과가 여러 개인 다중행 서브쿼리

---

“ ALL 연산자

“ 서브쿼리의 여러 결과 값에 메인쿼리의 조건식을 대입

“ 조건식을 대입한 결과가 모두가 **true**일 경우, 메인쿼리의  
조건식이 **true**



## 09-3 실행 결과가 여러 개인 다중행 서브쿼리

---

“ EXIT 연산자

“ 서브쿼리의 결과가 존재할 경우, true





## 09-4 비교할 열이 여러 개인 다중열 서브쿼리

---

“ 다중열 서브쿼리

“ 서브쿼리에 여러 열 지정

“ 메인쿼리의 비교 열은 동일한 개수, 동일한 자료형을  
괄호로 묶어서 지정



## 09-5 FROM절에 사용하는 서브쿼리와 WITH절

---

“ **FROM** 절에 사용하는 서브쿼리

“ 별칭을 지정하여 테이블처럼 사용

“ **WITH** 절

“ 메인쿼리 사용 전 먼저 이름과 함께 작성



## 09-6 SELECT절에 사용하는 서브쿼리

---

- “ 하나의 열 영역으로 사용
- “ 하나의 결과를 반환

