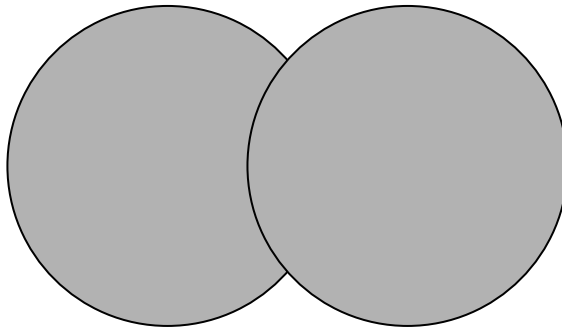


Introduction to Oracle9i : SQL

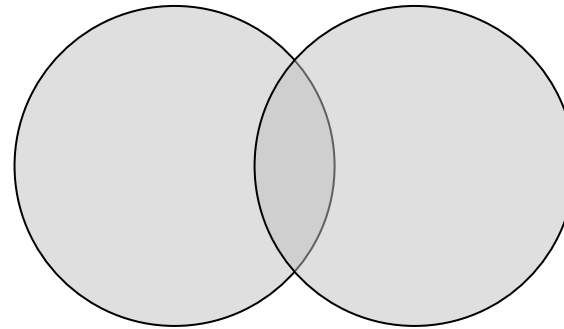
Chapter11. SET 연산자 및 Datetime 함수

- ▶ 여러 SELECT문의 결과를 SET연산자를 사용해 연결한다.
- ▶ Oracle9i에서 제공하는 날짜관련 함수를 익힌다.

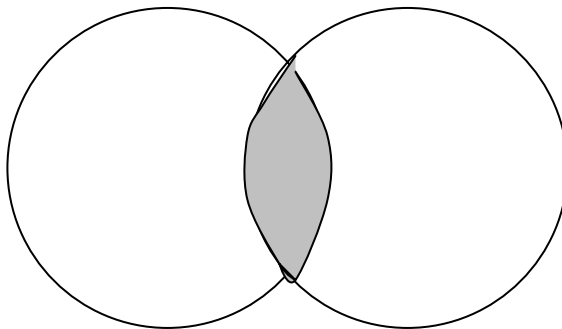
1. SET 연산자



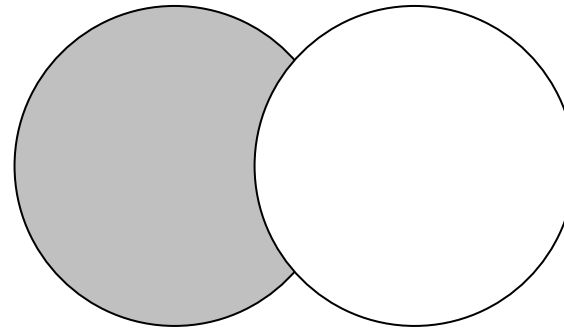
UNION



UNION ALL



INTERSECT



MINUS

※ 이 장에서 사용할 테이블

EMPLOYEES

컬럼이름
EMPLOYEE_ID
FIRST_NAME
LAST_NAME
EMAIL
PHONE_NUMBER
HIRE_DATE
JOB_ID
SALARY
COMMISSION_PCT
MANAGER_ID
DEPARTMENT_ID
DEPARTMENT_NAME

JOB_HISTORY

컬럼이름
EMPLOYEE_ID
START_DATE
END_DATE
JOB_ID
DEPARTMENT_ID

1. SET 연산자

(1) UNION 연산자

1) 정의

- 양쪽 질의문의 결과를 함께 출력하되 중복되는 행은 한번만 출력

2) 사용 시 고려사항

- 양쪽 질의문에서 검색하는 컬럼 수 및 데이터 타입 동일할 것
- 중복 체크 시 NULL값에 대해서도 검사
- IN 연산자의 우선순위가 높음
- 첫번째 질의문의 첫번째 컬럼값을 기준으로 오름차순 정렬

1. SET 연산자

(1) UNION 연산자

3) 사용

현재 직원 및 과거 직원의 직종 정보를 검색. 같은 직원에 대한 중복 행은 한번만 출력

```
SELECT employee_id, job_id  
FROM employees  
UNION  
SELECT employee_id, job_id  
FROM job_history ;
```

1. SET 연산자

(2) UNION ALL 연산자

1) 정의

- 양쪽 질의문의 결과를 함께 출력하되 중복되는 행은 중복 횟수만큼 출력

2) 사용 시 고려사항

- 결과에 대해 정렬 기능 비수행
- DISTINCT 키워드 사용 불가

1. SET 연산자

(2) UNION ALL 연산자

3) 사용

현재 직원 및 과거 직원의 직종 정보를 검색

```
SELECT employee_id, job_id  
FROM employees  
UNION ALL  
SELECT employee_id, job_id  
FROM job_history  
ORDER BY employee_id ;
```


1. SET 연산자

(3) INTERSECT 연산자

1) 정의

- 양쪽 질의문의 결과 중 중첩되는 값만 출력. 교집합

2) 사용 시 고려사항

- NULL값도 중첩 체크 시 반영

1. SET 연산자

(3) INTERSECT 연산자

3) 사용

현재 종사중인 직종에 과거에도 종사한 경험이 있는 직원 검색

```
SELECT employee_id, job_id  
FROM employees  
INTERSECT  
SELECT employee_id, job_id  
FROM job_history ;
```

1. SET 연산자

(4) MINUS 연산자

1) 정의

- 앞의 질의문의 결과에서 뒤의 질의문 결과를 뺀 값을 출력

2) 사용 시 고려사항

- WHERE절에 등장한 컬럼은 반드시 SELECT 절에 명시

1. SET 연산자

(4) MINUS 연산자

3) 사용

직종을 한 번도 변경하지 않은 직원 검색

```
SELECT employee_id, job_id
FROM   employees
MINUS
SELECT employee_id, job_id
FROM   job_history ;
```

1. SET 연산자


(5) SET 연산자 사용 지침

- 양쪽 SELECT문에 등장하는 컬럼의 수와 데이터 타입 일치
- 실행 순서의 변경을 위해 괄호 사용
- ORDER BY절은 마지막 문장의 마지막 절에 위치시키며, 컬럼명, 컬럼의 별칭등을 사용해서 정렬


1. SET 연산자

(6) SELECT 구문의 매칭

```
SELECT department_id, TO_NUMBER (null) location, hire_date
FROM employees
UNION
SELECT department_id, location_id, TO_DATE (null)
FROM departments ;
```



```
SELECT employee_id, job_id, salary
FROM employees
UNION
SELECT employee_id, job_id, 0
FROM job_history ;
```



1. SET 연산자

(7) 행의 순서 제어

```
COLUMN a_dummy NOPRINT
SELECT 'sing' AS "My dream", 3 a_dummy
FROM dual
UNION
SELECT 'I'd like to teach' AS "My dream", 1
FROM dual
UNION
SELECT 'the world to' AS "My dream", 2
FROM dual
ORDER BY 2 ;
```

My dream
I'd like to teach
the world to
sing

2. DATETIME 함수

(1) Oracle 9i 지원 날짜 관련 데이터 타입

데이터타입	설 명
TIMESTAMP	DATE 타입에서 저장하는 날짜 및 시간정보 외에 Fractional Second 정보도 저장
TIMESTAMP WITH TIME ZONE	TIMESTAMP의 변형. 날짜 시간 정보 및 시간 차 정보도 저장
TIMESTAMP WITH LOCAL TIME ZONE	데이터베이스의 시간대로 환산된 표준화 시간값 저장
INTERVAL YEAR TO MONTH	YEAR 및 MONTH로 환산하여 시간 구간값 저장
INTERVAL DAY TO SECOND	날짜, 시, 분, 초 단위로 시간 구간값 저장

2. DATETIME 함수

(2) DATETIME 함수의 종류

1) TZ_OFFSET

- 명령이 수행된 지역과 UTC 사이의 시간 차 정보
- V\$TIMEZONE_NAMES를 통해 지역명 확인

```
SELECT TZ_OFFSET ('US/Eastern') FROM dual ;
```

TZ_OFFS
-04:00

2. DATETIME 함수

(2) DATETIME 함수의 종류

2) CURRENT_DATE

- 세션을 일으키고 있는 time zone의 현재 날짜와 시간을 리턴
- 리턴값은 Gregorian 달력상의 날짜

```
ALTER SESSION  
SET NLS_DATE_FORMAT = 'DD-MON-YYYY HH24:MI:SS' ;
```

```
ALTER SESSION SET TIME_ZONE = '-5, 0' ;  
SELECT SESSIONTIMEZONE, CURRENT_DATE FROM DUAL ;
```

SESSIONTIMEZONE	CURRENT_DATE
-05:00	03-OCT-2001 19:37:06

2. DATETIME 함수

(2) DATETIME 함수의 종류

3) CURRENT_TIMESTAMP

- 세션을 일으키고 있는 time zone의 현재 날짜와 시간을 fractional second 정보와 함께 리턴
- 리턴값은 TIMESTAMP WITH TIME ZONE 데이터타입

```
ALTER SESSION SET TIME_ZONE = '-5, 0' ;  
SELECT SESSIONTIMEZONE, CURRENT_TIMESTAMP FROM DUAL ;
```

SESSIONTIMEZONE	CURRENT_TIMESTAMP
-05:00	03-OCT-01 09.40.59.000000 AM -05:00

2. DATETIME 함수

(2) DATETIME 함수의 종류

4) LOCALTIMESTAMP

- 세션을 일으키고 있는 time zone의 현재 날짜와 시간을 리턴
- 리턴값은 TIMESTAMP 데이터타입

```
ALTER SESSION SET TIME_ZONE = '-5, 0' ;  
SELECT CURRENT_TIMESTAMP, LOCALTIMESTAMP  
FROM DUAL ;
```

CURRENT_TIMESTAMP	LOCALTIMESTAMP
03-OCT-01 09.40.59.000000 AM -05:00	03-OCT-01 09.40.59.000000

2. DATETIME 함수

(2) DATETIME 함수의 종류

5) DBTIMEZONE

- 데이터베이스 time zone의 값을 리턴

```
SELECT DBTIMEZONE FROM DUAL ;
```

DBTIMEZONE
-05:00

6) SESSIONTIMEZONE

- 세션 time zone의 값을 리턴

```
SELECT SESSIONTIMEZONE FROM DUAL ;
```

DBTIMEZONE
-08:00

2. DATETIME 함수

(2) DATETIME 함수의 종류

7) EXTRACT

- 보고자 하는 날짜 요소만 추출하여 리턴

```
SELECT EXTRACT (YEAR FROM SYSDATE) FROM DUAL ;
```

DBTIMEZONE	
	2004

```
SELECT last_name, EXTRACT (MONTH FROM HIRE_DATE)  
FROM employees  
WHERE manager_id = 100 ;
```

2. DATETIME 함수

(2) DATETIME 함수의 종류

8) FROM_TZ

- **TIMESTAMP 값을 TIMESTAMP WITH TIME ZONE 값으로 리턴**

```
SELECT FROM_TZ (TIMESTAMP '2004-03-28 08:00:00', '3:00')  
FROM DUAL ;
```

```
FROM_TZ (TIMESTAMP '2004-03-28 08:00:00', '3:00')
```

```
28-MAR-04 08.00.00.000000 AM +03:00
```

2. DATETIME 함수

(2) DATETIME 함수의 종류

9) TO_TIMESTAMP

- 문자를 TIMESTAMP 타입으로 변환하여 리턴

```
SELECT TO_TIMESTAMP ('2004-03-28 08:00:00', 'YYYY-MM-DD HH:MI:SS' )  
FROM DUAL ;
```

TO_TIMESTAMP ('2004-03-28 08:00:00', 'YYYY-MM-DD HH:MI:SS')
--

28-MAR-04 08.00.00.000000 AM

2. DATETIME 함수

(2) DATETIME 함수의 종류

10) TO_TIMESTAMP_TZ

- 문자를 TIMESTAMP WITH TIME ZONE 타입으로 변환하여 리턴

```
SELECT TO_TIMESTAMP_TZ ('2004-03-28 08:00:00 -8:00 ',  
                        'YYYY-MM-DD HH:MI:SS TZH:TZM' )  
FROM DUAL ;
```

TO_TIMESTAMP_TZ('2004-03-28 08:00:00 -8:00','YYYY-MM-DD HH:MI:SS TZH:TZM')
28-MAR-04 08.00.00.000000 AM

2. DATETIME 함수

(2) DATETIME 함수의 종류

11) TO_YMINTERVAL

- 문자를 INTERVAL YEAR TO MONTH 타입으로 변환하여 리턴

```
SELECT hire_date, hire_date + TO_YMINTERVAL ('01-02')  
FROM   employees  
WHERE  department_id = 90 ;
```