

DML의 이후

LIKE : 검색에 특화되어 있다


- 비교하려는 값이 특정한 패턴을 만족 시키면 조회하는 연산자

[작성법]
WHERE 컬럼명 LIKE '패턴'


-LIKE의 패턴(와일드 카드): ' % ' ' _ '

- '%' 예시
- 1) 'A%' : 문자열이 A로 시작하는 모든 컬럼 값
 - 2) '%A' : 문자열이 A로 끝나는 모든 컬럼 값
 - 3) '%A%' : 문자열에 A가 포함되어 있는 모든 컬럼 값
- 1) 'A_': A 뒤에 아무거나 한 글자
- 2) '____A' : A 앞에 아무거나 3세글자
(4글자 문자열이면서 A로 끝나야됨)


'전' 시 조회



'하' 가 포함된 직원



010으로 시작하는 사원 / _로 찾아보는 방법 / NOT 으로 부정해보는 연산.





AND연산과 OR연산의 우선순위가 다르다 (AND가 더 높다)
괄호로 우선 연산을 시켜줘야 한다.

연산자 우선순위

-- 연산자 우선순위

- 1. 산술연산자
- 2. 연결연산자
- 3. 비교연산자
- 4. IS NULL / IS NOT NULL, LIKE, IN / NOT IN
- 5. BETWEEN AND / NOT BETWEEN AND
- 6. NOT(논리연산자)

7. AND(논리연산자)

8. OR(논리연산자)

/* IN 연산자

비교하려는 값과 목록에 작성된 값 중 일치하는 것이 있으면 조회하는 연산자.
OR 연산을 연달아 작성하는 효과를 가져온다

[작성법]

컬럼명 IN(값1, 값2, 값3)

- IN괄호 안이 목록

```
--EMPLOYEE 테이블에서
--부서코드가 D1, D6,D9인 사원의 사번 이름 부서코드 조회
SELECT EMP_ID,EMP_NAME,DEPT_CODE
FROM EMPLOYEE
WHERE DEPT_CODE IN('D1','D6','D9');

--사번이 200 , 205 ,210 인 사원의 사번 이름
SELECT EMP_ID,EMP_NAME
FROM EMPLOYEE
WHERE EMP_ID IN (200,205,210);

--사번이 200,205,210이 아닌 사원
SELECT EMP_ID,EMP_NAME
FROM EMPLOYEE
WHERE EMP_ID NOT IN (200,205,210);
```

연결 연산자



NULL 처리 연산자

- JAVA에서 NULL 참조하는 객체가 없다.
- DB에서 NULL : 컬럼 값이 없다.

- 1) IS NULL : 컬럼 값이 NULL인 경우 조회
- 2) IS NOT NUL : 컬럼 값이 NULL 아닌 경우 조회



LIKE %_의 의미
AND OR 연산 순서
IN연산자
연결 연산자
ISNULL / ISNOTNULL

ORDER BY 절

[작성법]

- 3 - SELECT 컬럼명 AS 별칭, 컬럼명, 컬럼명
- 1 - FROM 테이블명

- 2 - WHERE 조건식
- 4 - ORDER BY 컬럼 | 별칭 | 컬럼순서 [정렬방식(오름/내림)] [NULL FIRST \ LAST]

하나의 SELECT문에서 가장 마지막에 해석된다.

ASC : 오름차순 (점점 값이 커지는)



ASC를 생략해도 기본값이 오름차순이다.

DESC : 내림차순(점점 값이 작아지는)

```
--정렬은 숫자, 문자, 날짜 모두 사용 가능하다.
SELECT EMP_NAME, SALARY
FROM EMPLOYEE
ORDER BY NAME /*ASC*/;

SELECT EMP_NAME, HIRE_DATE AS 입사일
FROM EMPLOYEE
ORDER BY 입사일 DESC;
--별칭

SELECT EMP_NAME, HIRE_DATE
FROM EMPLOYEE
ORDER BY 2 DESC;
--컬럼 순서로도 조회 가능
```

ORDER BY같은 경우 마지막에 해석되기 때문에 별칭이 허용되는 것이다.

```
--보너스 정렬
SELECT EMP_NAME, BONUS
FROM EMPLOYEE
--ORDER BY BONUS DESC NULLS LAST;
ORDER BY BONUS NULLS FIRST;
--오름차순 정렬시 NULLS LAST가 기본값.
--내림차순 정렬시 NULLS FIRST가 기본값.

/*정렬 중첩*/
--EMPLOYEE 테이블에서 부서코드 오름차순 정렬 후
--부서별 급여 내림차순
SELECT EMP_NAME, DEPT_CODE , SALARY
FROM EMPLOYEE
ORDER BY DEPT_CODE, SALARY DESC;
--DEPT_CODE를 오름차순 정렬 후 , 급여를 내림차순 정렬을 진행한다.
```

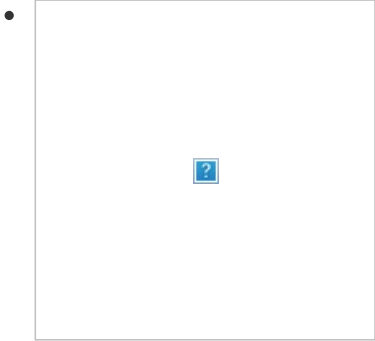
함수

02_함수(FUNCTION).pdf

752 kB

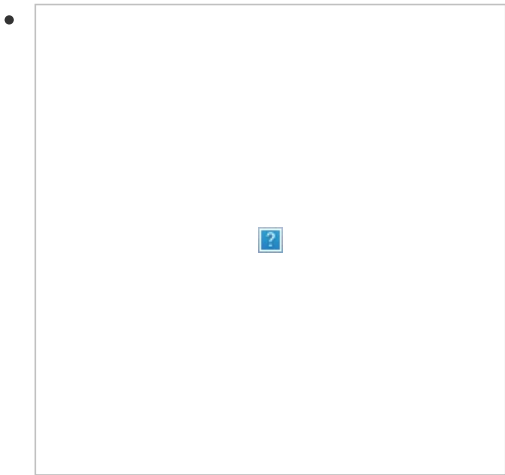
함수 : 컬럼의 값을 읽어서 연산한 결과를 반환한다.
여러 종류의 함수 문법을 배웠다.
함수의 문법 종류
LENGTH : 길이 값을 구함.

- LENGTH(문자열 | 컬럼) : 문자열 길이 반환



INSTR:조건에 맞는 문자의 시작 위치 반환

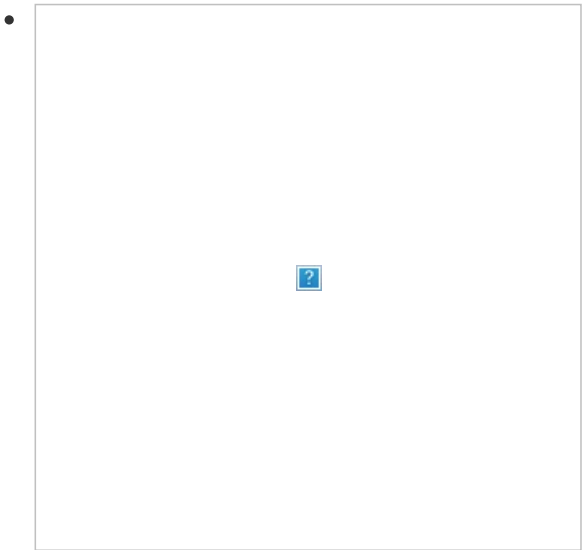
- INSTR('문자열' | 컬럼명, '찾을 문자', [찾을 위치 시작 위치, [순번]])
- 지정한 위치부터 지정한 순 번째로 검색되는 문자의 시작 위치를 반환



- EMAL에 '@'의 위치 조회

SUBSTR: 범위를 지정해 문자열을 잘라내어 반환한다

- 컬럼이나 문자열에서 지정한 위치부터 지정된 길이만큼 문자열을 잘라내서 반환
- - 잘라낼 길이 생략 시 끝까지 잘라냄



TRIM : 문자열의 앞,뒤,양쪽에 있는 지정된 문자들을 제거한다.



- '-' 문자들을 지운다.
- BOTH 또는 생략 시 : 양쪽의 '-' 제거
- LEADING : 앞쪽만 제거
- TRAILING : 뒤쪽만 제거.

숫자 관련 함수

ABS : 절대값
MOD : 나머지 반환(%는 사용 불가 함)

ROUND : 지정 소수점 위치 기준으로 '반올림'

CEIL : 올림
FLOOR : 내림

TRUNC : 특정 위치 아래를 <버림>

날짜 관련 함수

SYSDATE : 시스템에 현재 시간 (년/월/일/시/분/초)를 반환한다. 표기 설정 값을 변경 할 수 있다.
SYSTIMESTAMP : SYSDATE + MS 단위도 추가한다.
MONTHS_BETWEEN (날짜, 날짜) : 두 날짜의 개월 수 차이를 반환한다

- EX) ROUND(MONTHS_BETWEEN(SYSDATE,'2022/01/01')) = 3
 - ROUND로 반올림해서 1월1일부터 오늘까지의 개월 수를 측정했다

ADD_MONTH (날짜,숫자) : 날짜에 숫자만큼 개월 수를 더함.
LAST_DAT (날짜) : 해당 달의 마지막 날짜를 구함.

EXTRACT : 년, 월, 일 정보를 추출하여 리턴

EXTRACT(YEAR FROM 날짜) : 년도만 추출
EXTRACT(MONTH FROM 날짜) : 월만 추출
EXTRACT(DAY FROM 날짜) : 일만 추출

형변환 함수

문자열(CHAR)

숫자(NUMBER)
날짜 (DATE)

3 종류간에 형 변환이 가능하다.

```
TO_CHAR(날짜, [포맷]) : 날짜형 데이터를 문자형 데이터로 변경
TO_CHAR(숫자, [포맷]) : 숫자형 데이터를 문자형 데이터로 변경

TO_DATE(문자형 데이터, [포맷]) : 문자형 데이터를 날짜로 변경
TO_DATE(숫자형 데이터, [포맷]) : 숫자형 데이터를 날짜로 변경
지정된 포맷으로 날짜를 인식함.

TO_NUMBER(문자데이터, [포맷]) : 문자형데이터를 숫자 데이터로 변경
```

TO_CHAR

패턴

- 9: 숫자 한 칸을 의미한다 여러 개 작성 시 오른쪽 정렬
- 0: 9와 같은 역할에 '0'을 추가한다.

```
SELECT TO_CHAR(1234, '99999') FROM DUAL; --숫자 5칸 오른쪽 정렬.
SELECT TO_CHAR(1234, '00000') FROM DUAL; --숫자 5칸 . 오른쪽 정렬 빈칸 0 추가

SELECT TO_CHAR(1000000, '9,999,999') FROM DUAL;
SELECT TO_CHAR(1000000, '$9,999,999') FROM DUAL;
```

```
--날짜에 TO_CHAR 적용
--YYYY : 년도 / YY :년(짧게)
--RRRR : 년도 / RR : 년도 짧게
--MM : 월
--DD : 일수
--AM/PM : 오전 /오후를 표기하겠다
--HH : 시간 (24 불일경우 24시간 표기법)
--MI : 분
--SS : 초
--DAY : 요일 전체 / DY : 요일명만 표기
```

```
SELECT EMP_NAME , TO_CHAR(HIRE_DATE, 'RRRR"년" MM"월" DD"일" (DY)') FROM EMPLOYEE;
```


- 날짜 형을 RRRR년 MM월 DD일 포맷을 가지고 문자 형으로 변경한다.
 - 이 때 년,월,일은 오라클에 등록된 날짜 표기 패턴이 아니라 ""(쌍따옴표)로 새로운 패턴을 정의해 줘야 한다.
 - ' '(따옴표)로 모든 문을 감싼 후 ""문자를 쌍따옴표로 감싼다.

```
SELECT TO_DATE(20100110) FROM DUAL;
```

패턴에 맞게 하면 정수로 해도 날짜 정보로 형변환이 가능하다.

```
SELECT TO_CHAR(TO_DATE('041030 143000','YYMMDD HH24MISS'),
                  'YYYY/MM/DD HH24"시"MI"분"') FROM DUAL;
```

날짜 형으로 변환한다

- TO_DATE('041030 143000','YYMMDD HH24MISS')
 - '041030 143000'이라는 값을 'YYMMDD(년월일) HH24MISS(24시간,분 ,초)'포맷을 만들었다
- 만들어진 날짜형 데이터를 TO_CHAR으로 형변환 한다.
 - 이때 YY값인 04는 2004로 변하고 각각 형식에 맞는 포맷에 들어간다
 - 

```
---- TO_NUMBER(문자데이터, [포맷]) : 문자형데이터를 숫자 데이터로 변경
SELECT TO_NUMBER('1,000,000','9,999,999') + 10 FROM DUAL;
```

숫자형 변환은 뭐 별거 없다.

NULL처리 함수

NVL(컬럼명 ,컬럼값이 NULL일 경우 바꿀 값)

```
SELECT EMP_NAME, SALARY, NVL (BONUS, 0) , NVL (BONUS, 0) *SALARY
FROM EMPLOYEE;
```

보너스의 값이 NULL일 경우 0 대입
보너스의 값이 NULL일 경우 0을 대입 '후' SALARY와 연산

NVL2(컬럼명, 바꿀값1, 바꿀값2)

해당 컬럼의 값이 있으면 바꿀값1로 변경,
해당 컬럼이 NULL이면 바꿀값2로 변경

```
SELECT EMP_NAME, BONUS, NVL2 (BONUS, 0.8, 0.3)
FROM EMPLOYEE;
```

BONUS에 값이 있을 경우 0.8
값이 NULL일 경우 0.3