[Database] SQL select 문 사용법

노트북:	[TIL]
만든 날짜:	2020-07-15 오전 10:47
URL:	https://continuous-development.tistory.com/12?category=736682

Database

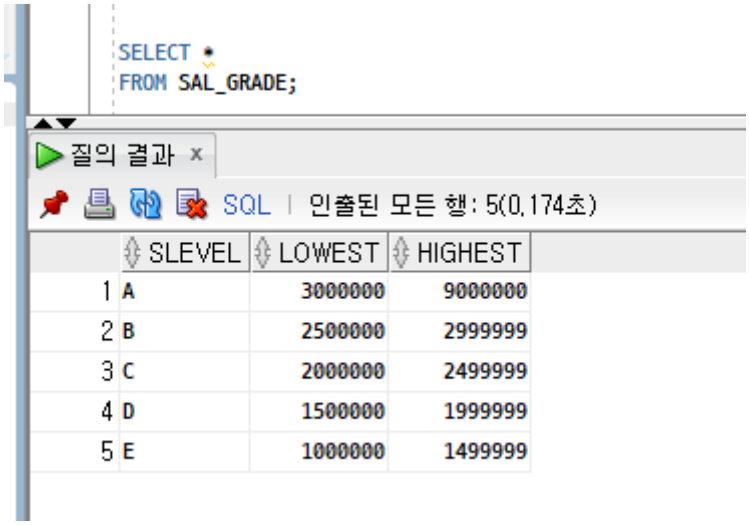
[Database] SQL select 문 사용법

꾸까꾸 2020. 7. 14. 12:53 수정 삭제

1.기본 select 절

SELECT 컬럼명
FROM 테이블명;

여기서 * 은 모든 컬럼을 뜻한다.



	SLEVEL	LOWEST	HIGHEST
1	A	3000000	9000000
2	B	2500000	2999999
3	C	2000000	2499999
4	D	1500000	1999999
5	E	1000000	1499999

이렇게 각각의 컬럼을 명시해도 된다. 보통 가독성을 위해 아래와 같이 많이 쓰여진다.

```
SELECT  SLEVEL,
        LOWEST,
        HIGHEST
FROM    SAL_GRADE;
```

질의 결과 x

SQL | 인출된 모든 행: 5(0.004초)

	SLEVEL	LOWEST	HIGHEST
1	A	3000000	9000000
2	B	2500000	2999999
3	C	2000000	2499999
4	D	1500000	1999999
5	E	1000000	1499999

2.컬럼 값에 대해 연산

SELECT 컬럼명 * 12
FROM 테이블명;

```
SELECT  *
FROM    EMPLOYEE;
```

질의 결과 x

SQL | 인출된 모든 행: 22(0.006초)

	EMP_ID	EMP_NAME	EMP_NO	EMAIL	PHONE	HIRE_DATE	JOB_ID	SALARY	BONUS_PCT	MARRIAGE	MGR_ID	DEPT_ID
1	100	한선기	621133-1483658	sg_ahn@vcc.com	0199949999	90/04/01	J1	9000000	0.2	Y	(null)	90
2	101	강중훈	621136-1006405	jh_park@vcc.com	0193334433	04/04/30	J2	5500000	(null)	Y	100	90
3	102	최만식	861011-1940062	ms_choi@vcc.com	0198879908	95/12/30	J2	3600000	(null)	Y	101	90
4	103	정도연	631127-2519077	sy_kang@vcc.com	0196654436	97/06/03	J4	2600000	(null)	Y	104	60
5	104	안석규	651031-1962810	sg_han@vcc.com	0192347654	98/07/01	J3	3500000	0.25	Y	100	60
6	107	조재영	721128-1732822	jh_jo@vcc.com	0193325548	98/11/23	J3	3800000	(null)	Y	104	60
7	124	정지현	641231-2269080	jih_jeon@vcc.com	01922976129	04/07/15	J7	1500000	(null)	N	141	50
8	141	김예수	651122-2592930	hs_kim@vcc.com	0194087600	01/03/20	J5	2100000	0.1	Y	100	50
9	143	나승원	871024-1945881	sw_cha@vcc.com	0197243979	01/03/20	J5	2300000	(null)	Y	141	50

해당 EMPLOYEE 컬럼을 봤을때 연봉을 계산 하기 위해서는 SALARY * 12가 필요하다.
이런식으로 해당 컬럼에 대해 산술 연산이 가능하다.

```
SELECT   EMP_NAME,
         SALARY*12,
         (SALARY+(SALARY*BONUS_PCT))*12
FROM     EMPLOYEE;
```

질의 결과

SQL | 인출된 모든 행: 22(0.006초)

	EMP_NAME	SALARY*12	(SALARY+(SALARY*BONUS_PCT))*12
1	한선기	108000000	129600000
2	강중훈	66000000	(null)
3	최만식	43200000	(null)
4	정도연	31200000	(null)
5	안석규	42000000	52500000
6	조재형	45600000	(null)
7	정지현	18000000	(null)
8	김예수	25200000	27720000
9	나승원	27600000	(null)
10	김순이	40800000	44880000

3.컬럼값에 대한 alias(별칭)

SELECT 컬럼명 AS 별칭
FROM 테이블명;

컬럼뒤 에 있는 as sql alias(별칭)로서 컬럼의 이름을 새로 지을수 있다. 그래서 컬럼 뒤에 AS 를 붙이고 내가 원하는 컬럼명으로 바꿔서 넣어준다. 여기서 실제 컬럼명이 바뀌는건 아니고 지금 당시 보일때만 컬럼명이 바껴서 나온다.
지금 현재 이름에는 " " 가 안들어가고 다른 컬럼에는 " " 가 들어가 있다. alias를 쓸때 주의할 점은 문자에 대해서는 그냥 사용해도 되지만 특수문자나(여기서는 ( )) 첫글자의 시작이 숫자인(1)경우에는 " " 이 쌍따옴표 안에 원하는 것을 넣어야 한다. 그리고 대소문자를 구분해야 하는 경우에도 쌍따옴표를 써야한다.

```
SELECT  EMP_NAME AS 이름,
        SALARY*12 AS "1년 급여",
        (SALARY+(SALARY*BONUS_PCT))*12 AS "총 소득(원)"
FROM    EMPLOYEE;
```

질의 결과 x

SQL | 인출된 모든 행: 22(0.006초)

	이름	1년 급여	총 소득(원)
1	한선기	108000000	129600000
2	강중훈	66000000	(null)
3	최만식	43200000	(null)
4	정도연	31200000	(null)
5	안석규	42000000	52500000
6	조재형	45600000	(null)

4.DISTINCT로 중복 값을 한번만 보기

SELECT DISTINCT 컬럼명
FROM 테이블명;

현재 아래 테이블을 검색했을때는 중복되는 값이 보인다. 90 이나 60,50등 이걸 중복을 뺀 값을 보기 위해서는

```
SELECT  DEPT_ID
FROM    EMPLOYEE
```

질의 결과 x

SQL | 인출된 모든 행: 22(

	DEPT_ID
1	90
2	90
3	90
4	60
5	60
6	60
7	50
8	50
9	50
10	50
11	50

DISTINCT를 사용해서 이런식으로 볼 수 있다.

```
SELECT  DISTINCT DEPT_ID
FROM    EMPLOYEE;
```

질의 결과

SQL | 인출된 모든 행: 7(0,0

	DEPT_ID
1	(null)
2	50
3	20
4	10
5	90
6	80
7	60

5.WHERE 절 사용

SELECT 컬럼명
FROM 테이블명
WHERE 컬럼 = 값;

WHERE 절 사용을 통해 컬럼에서 내가 원하는 값을 찾거나 범위등을 지정해서 내가 필요한 값들만 추출해 낼 수 있다.

```
SELECT  *
FROM    EMPLOYEE
WHERE   DEPT_ID = 90
```

질의 결과

SQL | 인출된 모든 행: 3(0.002초)

	AME	EMP_NO	EMAIL	PHONE	HIRE_DATE	JOB_ID	SALARY	BONUS_PCT	MARRIAGE	MGR_ID	DEPT_ID
1		621133-1483658	sg_ahn@vcc.com	0199949999	90/04/01	J1	9000000	0.2	Y	(null)	90
2		621136-1006405	jh_park@vcc.com	0193334433	04/04/30	J2	5500000	(null)	Y	100	90
3		861011-1940062	ms_choi@vcc.com	0198879908	95/12/30	J2	3600000	(null)	Y	101	90

위와 같은 경우에서 DEPT_ID 가 90일 때를 말한다.

6.연산자 사용

SELECT 컬럼명
FROM 테이블명
WHERE 컬럼 = 값 연산자 컬럼 = 값;

연산자의 종류

논리 연산자
AND : A 그리고 B - 둘다 조건에 맞아야 TRUE
OR : A 또는 B 중 하나의 조건만 맞아도 TRUE
NOT : 조건의 반대 값

비교 연산자
= - 같다
<,> - 크거나 작다
<=,>= - 크거나 같다 , 작거나 같다.
<>,!=,^= - 같지않다
BETWEEN AND - 특정범위에 포함되는걸 찾는다
LIKE / NOT LIKE - 문자 패턴비교
IS NULL / IS NOT NULL - NULL 여부 비교
IN - 비교 값 목록에 포함되

```
SELECT *
FROM   EMPLOYEE
WHERE  DEPT_ID = 90 AND SALARY > 5000000;
```

질의 결과

SQL | 인출된 모든 행: 2(0.005초)

	EMP_ID	EMP_NAME	EMP_NO	EMAIL	PHONE	HIRE_DATE	JOB_ID	SALARY	BONUS_PCT	MARRIAGE
1	100	한선기	621133-1483658	sg_ahn@vcc.com	0199949999	90/04/01	J1	9000000	0.2	Y
2	101	강중훈	621136-1006405	jh_park@vcc.com	0193334433	04/04/30	J2	5500000	(null)	Y

AND 조건

아래와 같은 경우에서 DEPT_ID 가 90 이고 SALARY 가 5000000 이상인 경우만 출력한다는 뜻이다. 이때

가운데 AND는 말 뜻대로 그리고의 뜻으로 두가지 조건을 모두 충족할 경우라고 보면 된다.

```
SELECT *
FROM   EMPLOYEE
WHERE  DEPT_ID = 90 OR SALARY > 5000000;
```

질의 결과

SQL | 인출된 모든 행: 3(0.004초)

	AME	EMP_NO	EMAIL	PHONE	HIRE_DATE	JOB_ID	SALARY	BONUS_PCT	MARRIAGE	MGR_ID	DEPT_ID
1		621133-1483658	sg_ahn@vcc.com	0199949999	90/04/01	J1	9000000	0.2	Y	(null)	90
2		621136-1006405	jh_park@vcc.com	0193334433	04/04/30	J2	5500000	(null)	Y	100	90
3		861011-1940062	ms_choi@vcc.com	0198879908	95/12/30	J2	3600000	(null)	Y	101	90

OR 조건

여기서는 DEPT_ID가 90 이거나 또는 SALARY가 500000 이상인 경우 모두 출력된다.

각 컬럼을 연결하여 값을 연결해서 나오게끔 한다.

```
SELECT  EMP_ID||EMP_NAME||SALARY
FROM    EMPLOYEE;
```

질의 결과

SQL | 인출된 모든 행: 22(0.004

	EMP_ID||EMP_NAME||SALARY
1	100한선기9000000
2	101강중훈5500000
3	102최만식3600000
4	103정도연2600000
5	104안석규3500000
6	107조재형3800000

BETWEEN 값 AND 값 을 넣어줘 사이에 있는 값을 찾는다.

```
SELECT  EMP_NAME,
        SALARY
FROM    EMPLOYEE
WHERE   SALARY BETWEEN 3500000 AND 5500000;
```

질의 결과

SQL | 인출된 모든 행: 4(0.003초)

	EMP_NAME	SALARY
1	강중훈	5500000
2	최만식	3600000
3	안석규	3500000
4	조재형	3800000

BETWEEN 사용

LIKE '원하는 값'으로 넣어준다. 여기서 김% 는 김이라는 문자열을 찾은 후 그 뒤에는 어떤 문자가 와도 상관 없다는 뜻이다. 김으로 시작되는 김XX 가능
%예% 라고 칠경우 가운데에 예라는 문자만 들어오면 앞뒤는 어떤 문자여도 상관 없다.

```
SELECT  EMP_NAME,
        SALARY
FROM    EMPLOYEE
WHERE   EMP_NAME LIKE '김%';
```

질의 결과

SQL | 인출된 모든 행: 3(0.004초)

	EMP_NAME	SALARY
1	김예수	2100000
2	김순이	3400000
3	김술오	2500000

LIKE % RUDDN

IS NULL 같은 경우에 값이 NULL 인 행을 찾는다.
IS NOT NULL 같은 경우에는 반대인 경우 값이 있는 행을 찾는다.

```
SELECT  EMP_NAME,
        MGR_ID,
        DEPT_ID
FROM    EMPLOYEE
WHERE   MGR_ID IS NULL
AND     DEPT_ID IS NULL;
```

질의 결과

SQL | 인출된 모든 행: 2(0.004초)

	EMP_NAME	MGR_ID	DEPT_ID
1	심하균	(null)	(null)
2	염정하	(null)	(null)

IS NULL

```
SELECT  EMP_NAME,
        MGR_ID,
        DEPT_ID
FROM    EMPLOYEE
WHERE   MGR_ID IS NOT NULL
```

질의 결과

SQL | 인출된 모든 행: 15(0.003초)

	EMP_NAME	MGR_ID	DEPT_ID
1	강중훈	100	90
2	최만식	101	90
3	정도연	104	60
4	안석규	100	60
5	조재형	104	60
6	정지현	141	50
7	김예수	100	50
8	나승원	141	50
9	김순이	141	50

IS NOT NULL

IN 을 통해서 여러개의 조건을 한번에 확인한다 여기서는 OR 비슷하다.
OR 로 했을경우

WHERE EMP_NAME = '심하균' OR EMP_NAME = '염정하'
이런 느낌이다.

```
SELECT  EMP_NAME,
        SALARY
FROM    EMPLOYEE
WHERE   EMP_NAME IN ('심하균','염정하');
```

질의 결과

SQL | 인출된 모든 행: 2(0.004초)

	EMP_NAME	SALARY
1	심하균	2300000
2	염정하	1500000

IN

'**Database**' 카테고리의 다른 글

[Database]LPAD/RPAD 함수 사용법 (0) 2020.07.14

[Database] INSTR, LENGT 함수 사용법 (0) 2020.07.14

[Database] SQL select 문 사용법 (0) 2020.07.14

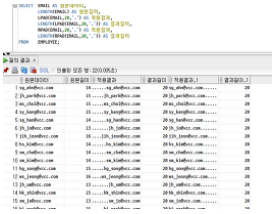
[Database] sql developer 설치하기 (0) 2020.07.13

[Database] 윈도우 에서 oracle 설치하기 (0) 2020.07.13

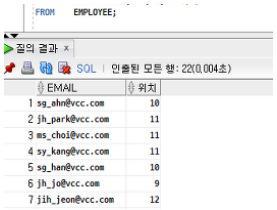
리눅스 mysql 설치 및 권한 설정 (0) 2019.11.22

태그 #select, #select option, #SELECT 사용, #SQL, #sql select, #셀렉

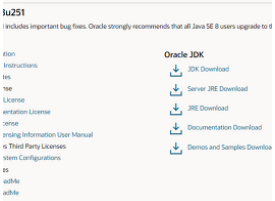
'Database' Related Articles



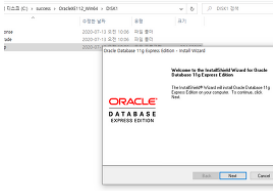
[Database]LPAD/R 함수 사용법



[Database] INSTR, LENGT ...



[Database] sql developer 설치...



[Database] 윈도우 에서 oracle ...