

Chapter 03. SQLD 이론

주요개념 Wrap up 3 – SQL기본

NULL 정리해 봅시다 – 사칙연산.

- NULL 은 아직 정의되지 않은 값으로 0 혹은 공백이 아니다. 즉 Null 은 Null 이다.
- NULL과의 수치연산의 결과는 NULL이다.

사칙연산	결과
NULL + 1	NULL
NULL - 1	
NULL * 1	
NULL / 1	
<div><div><div>* 큰 수를 연산해도 결과는 NULL 이다.</div><div>SELECT NULL + 100,000,000</div><div>SELECT NULL - 100,000,000</div><div>SELECT NULL * 100,000,000</div><div>SELECT NULL / 100,000,000</div></div><div>➡</div><div>NULL</div></div>	

NULL 정리해 봅시다 – 집계함수.

학점

학생ID	점수
1	100
2	80
3	85
4	NULL
5	50
6	NULL
7	75
8	NULL

* NULL은 집계 함수 대상에 제외 된다!

SELECT AVG(점수) FROM 학점 → 78

=

SELECT SUM(점수) FROM 학점 → 390

SELECT COUNT(점수) FROM 학점 → 5

NULL 정리해 봅시다 – 조회 조건 .

- NULL 은 오직 IS NULL , IS NOT NULL 로만 검색할 수 있다. 즉 < , > , = 등은 사용할 수 없다.

TABLE	SQL	결과 건수					
<div>TAB1</div> <table><tr><th>COL1</th></tr><tr><td>100</td></tr><tr><td>80</td></tr><tr><td>85</td></tr><tr><td>NULL</td></tr></table>	COL1	100	80	85	NULL	<div>SELET * FROM WHERE COL1 = NULL</div> <div>SELET * FROM WHERE COL1 <> NULL</div> <div>SELET * FROM WHERE COL1 IS NULL</div> <div>SELET * FROM WHERE COL1 IS NOT NULL</div>	<div>0건</div> <div>1건</div> <div>3건</div>
COL1							
100							
80							
85							
NULL							

NULL 정리해 봅시다 – 나올 수 있는 문제들..

TAB1

COL1	COL2	COL3	COL4
40	NULL	NULL	0
NULL	10	10	30
50	NULL	20	NULL

출제 가능 문제들

1. SELECT COL1 + COL4 FROM TAB1
2. SELECT SUM(COL1) + SUM(COL3) FROM TAB1
3. SELECT SUM(COL2 + COL3) FROM TAB1
4. SELECT AVG(COL3) FROM TAB1 VS SELECT AVG(COL4) FROM TAB1
5. SELECT AVG(COL4) FROM TAB1 WHERE COL4 IS NOT NULL VS SELECT AVG(COL4) FROM TAB1
6. SELECT COUNT(COL1 + COL2 + COL3 + COL4) FROM TAB1 VS SELECT COUNT(COL1) + COUNT(COL2) + COUNT(COL3) + COUNT(COL4) FROM TAB1

NULL 정리해 봅시다 – 나올 수 있는 문제 1

TAB1

COL1	COL2	COL3	COL4
40	NULL	NULL	0
NULL	10	10	30
50	NULL	20	NULL

1. SELECT COL1 + COL4 FROM TAB1

COL1		COL4	
40	+	0	= 40
NULL	+	30	= NULL
50	+	NULL	= NULL

NULL 정리해 봅시다 – 나올 수 있는 문제 2

TAB1

COL1	COL2	COL3	COL4
40	NULL	NULL	0
NULL	10	10	30
50	NULL	20	NULL

2. SELECT SUM(COL1) + SUM(COL3) FROM TAB1

COL1	COL3
40	NULL
SUM NULL	SUM 10
50	20

90 + 30 = 120

NULL 정리해 봅시다 – 나올 수 있는 문제 3

TAB1

COL1	COL2	COL3	COL4
40	NULL	NULL	0
NULL	10	10	30
50	NULL	20	NULL

3. SELECT SUM(COL2 + COL3) FROM TAB1

COL2	COL3
NULL	NULL
10	10
NULL	20

= NULL

= 20

= NULL

20

SUM

NULL 정리해 봅시다 – 나올 수 있는 문제 4

TAB1

COL1	COL2	COL3	COL4
40	NULL	NULL	0
NULL	10	10	30
50	NULL	20	NULL

4. SELECT AVG(COL3) FROM TAB1 VS SELECT AVG(COL4) FROM TAB1

COL3
NULL
AVG 10
20

$30 / 2 = 15$

COL4
0
AVG 30
NULL

$30 / 2 = 15$

NULL 정리해 봅시다 – 나올 수 있는 문제 5

TAB1

COL1	COL2	COL3	COL4
40	NULL	NULL	0
NULL	10	10	30
50	NULL	20	NULL

5. SELECT AVG(COL4) FROM TAB1 VS SELECT AVG(COL4) FROM TAB1
WHERE COL4 IS NOT NULL

COL4
0
AVG 30
NULL

$30 / 2 = 15$

COL4
0
AVG 30
NULL

$30 / 2 = 15$

NULL 정리해 봅시다 – 나올 수 있는 문제 6

SELECT COUNT(COL1 + COL2 + COL3 + COL4) FROM TA1

TAB1

COL1	COL2	COL3	COL4
40	NULL	NULL	0
NULL	10	10	30
50	NULL	20	NULL

COUNT
= NULL
= NULL
= NULL
0

VS

SELECT COUNT(COL1) + COUNT(COL2) + COUNT (COL3) + COUNT (COL4) FROM TA1

TAB1

COL1	COL2	COL3	COL4
COUNT 40	COUNT NULL	COUNT NULL	COUNT 0
NULL	10	10	30
50	NULL	20	NULL

2 + 1 + 2 + 2 = 7

SQL의 종류

명령어의 종류	명령어	설명
데이터 조작어 (DML: Data Manipulation Language)	SELECT	데이터베이스에 들어 있는 데이터를 조회하거나 검색하기 위한 명령어를 말하는 것으로 RETRIEVE 라고도 한다.
	INSERT UPDATE DELETE	데이터베이스의 테이블에 들어 있는 데이터에 변형을 가하는 종류의 명령어들을 말한다. 예를 들어 데이터를 테이블에 새로운 행을 집어 넣거나, 원하지 않는 데이터를 삭제하거나 수정하는 것들의 명령어들을 DML이라고 부른다.
데이터 정의어 (DDL: Data Definition Language)	CREATE ALTER DROP RENAME	테이블과 같은 데이터 구조를 정의하는데 사용되는 명령어들로 그러한 구조를 생성하거나 변경하거나 삭제하거나 이름을 바꾸는 데이터 구조와 관련된 명령어들을 DDL이라고 부른다.
데이터 제어어 (DCL: Data Control Language)	GRANT REVOKE	데이터베이스에 접근하고 객체들을 사용하도록 권한을 주고 회수하는 명령어를 DCL이라고 부른다.
트랜잭션 제어어 (TCL: Transaction Control Language)	COMMIT ROLLBACK	논리적인 작업의 단위를 묶어서 DML에 의해 조작된 결과를 작업단위(트랜잭션) 별로 제어하는 명령어를 말한다.

SELECT 문장 실행 순서

FROM → WHERE → GROUP BY → HAVING → SELECT → ORDER BY

5	SELECT 칼럼명 [ALIAS명]	SELECT JOB , COUNT(*) CNT
1	FROM 테이블명	FROM EMP
2	WHERE 조건식	WHERE SAL >= 4000
3	GROUP BY 칼럼(Column)이나 표현식	GROUP BY JOB
4	HAVING 그룹조건식	HAVING COUNT(*) > 0
6	ORDER BY 칼럼(Column)이나 표현식	ORDER BY CNT

ORDER BY 의 표현 1

* 6개의 SQL 쿼리는 같은 순서로 데이터가 추출된다.

```
SELECT DNAME, LOC, DEPTNO
FROM DEPT
ORDER BY DNAME ASC, LOC ASC, DEPTNO DESC;
```

```
SELECT DNAME, LOC, DEPTNO
FROM DEPT
ORDER BY DNAME, LOC, DEPTNO DESC;
```

```
SELECT DNAME, LOC, DEPTNO
FROM DEPT
ORDER BY 1 ASC, 2 ASC, 3 DESC;
```

```
SELECT DNAME, LOC, DEPTNO
FROM DEPT
ORDER BY 1, 2 , 3 DESC;
```

```
SELECT DNAME as A , LOC as B, DEPTNO as C
FROM DEPT
ORDER BY A ASC, B ASC, C DESC;
```

```
SELECT DNAME as A , LOC as B, DEPTNO as C
FROM DEPT
ORDER BY A, B , C DESC;
```

ORDER BY 의 표현 2

* GROUP BY 절 사용시 SELECT 절에 정의되지 않은 집계 칼럼을 ORDER BY 절 사용 가능

```
SELECT JOB  
FROM EMP  
GROUP BY JOB  
HAVING COUNT(*) > 0  
ORDER BY MAX(EMPNO), MAX(MGR), SUM(SAL), COUNT(DEPTNO), MAX(HIREDATE);
```

```
SELECT JOB , MAX(EMPNO), MAX(MGR), SUM(SAL), COUNT(DEPTNO), MAX(HIREDATE)  
FROM EMP  
GROUP BY JOB  
HAVING COUNT(*) > 0  
ORDER BY MAX(EMPNO), MAX(MGR), SUM(SAL), COUNT(DEPTNO), MAX(HIREDATE);
```

```
SELECT JOB , MAX(EMPNO), MAX(MGR), SUM(SAL), COUNT(DEPTNO), MAX(HIREDATE)  
FROM EMP  
GROUP BY JOB  
HAVING COUNT(*) > 0  
ORDER BY 2,3,4,5,6;
```

GROUP BY

* Group by 절에 선언된 컬럼을 기준으로 데이터 집합이 결정됨.

emp

empno	ename	deptno	job	mgr	hiredate	sal	comm
7934	MILLER	10	CLERK	7782	2000-01-21	1300	
7782	CLARK	10	MANAGER	7839	1993-05-14	2450	
7839	KING	10	PRESIDENT		1990-06-09	5000	0.00
7788	SCOTT	20	ANALYST	7566	1996-03-05	3000	
7902	FORD	20	ANALYST	7566	1997-12-05	3000	
7369	SMITH	20	CLERK	7902	1993-06-13	800	0.00
7876	ADAMS	20	CLERK	7788	1999-06-04	1100	
7566	JONES	20	MANAGER	7839	1995-10-31	2975	
7900	JAMES	30	CLERK	7698	2000-06-23	950	
7698	BLAKE	30	MANAGER	7839	1992-06-11	2850	
7499	ALLEN	30	SALESMAN	7698	1998-08-15	1600	300.00
7521	WARD	30	SALESMAN	7698	1996-03-26	1250	500.00
7844	TURNER	30	SALESMAN	7698	1995-06-04	1500	0.00
7654	MARTIN	30	SALESMAN	7698	1998-12-05	1250	1400.00

SELECT DEPTNO, COUNT(*) CNT
FROM EMP
GROUP BY DEPTNO;

deptno	cnt
10	3
20	5
30	6

SELECT DEPTNO, MAX(SAL) MAX_SAL
FROM EMP
GROUP BY DEPTNO;

deptno	max_sal
10	5000
20	2975
30	2850

GROUP BY (계속)

emp

empno	ename	deptno	job	mgr	hiredate	sal	comm
7934	MILLER	10	CLERK	7782	2000-01-21	1300	
7782	CLARK	10	MANAGER	7839	1993-05-14	2450	
7839	KING	10	PRESIDENT		1990-06-09	5000	0.00
7788	SCOTT	20	ANALYST	7566	1996-03-05	3000	
7902	FORD	20	ANALYST	7566	1997-12-05	3000	
7369	SMITH	20	CLERK	7902	1993-06-13	800	0.00
7876	ADAMS	20	CLERK	7788	1999-06-04	1100	
7566	JONES	20	MANAGER	7839	1995-10-31	2975	
7900	JAMES	30	CLERK	7698	2000-06-23	950	
7698	BLAKE	30	MANAGER	7839	1992-06-11	2850	
7499	ALLEN	30	SALESMAN	7698	1998-08-15	1600	300.00
7521	WARD	30	SALESMAN	7698	1996-03-26	1250	500.00
7844	TURNER	30	SALESMAN	7698	1995-06-04	1500	0.00
7654	MARTIN	30	SALESMAN	7698	1998-12-05	1250	1400.00

SELECT JOB, DEPTNO
FROM EMP
GROUP BY JOB, DEPTNO;

deptno	job
10	CLERK
10	MANAGER
10	PRESIDENT
20	ANALYST
20	CLERK
20	MANAGER
30	CLERK
30	MANAGER
30	SALESMAN

SELECT JOB, HIREDATE
FROM EMP
GROUP BY JOB;

→ ERROR 발생함, GROUP BY 절에 참여한 컬럼만
SELECT LIST에 있어야 함.

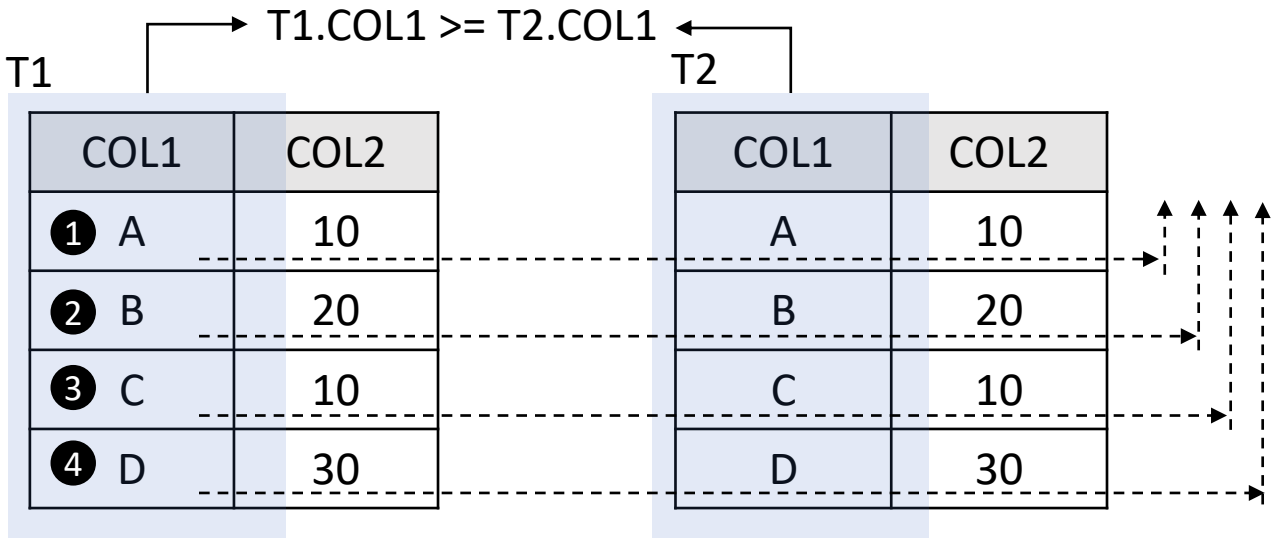
Non EQUI JOIN

"=" 연산자가 아닌 다른 (Between, >, >=, <, <= 등) 연산자들을 사용하여 JOIN을 수행함.

TAB

COL1	COL2
A	10
B	20
C	10
D	30

```
SELECT *  
FROM TAB T1, TAB T2  
WHERE T1.COL1 >= T2.COL1;
```



	T1		T2	
	COL1	COL2	COL1	COL2
1	A	10	A	10
2	B	20	A	10
	B	20	B	20
3	C	30	A	10
	C	30	B	20
	C	30	C	30
4	D	40	A	10
	D	40	B	20
	D	40	C	30
	D	40	D	40

수고하셨습니다.