

SQL Pogramming

- *Day 9* -

2023. 04

목차

Day 1. 데이터베이스와 SQL
Day 2. 테이블 / 인덱스
Day 3. DDL / DML / DCL / TCL
Day 4. SELECT 기본문형 익히기1
Day 5. SELECT 기본문형 익히기2
Day 6. 서브쿼리 / 스칼라쿼리
Day 7. 뷰 / 인라인뷰
Day 8. 내장함수 일반
Day 9. 내장함수 CASE
Day 10. 조인 기본
Day 11. 조인 활용1
Day 12. 조인 활용2

Day 13. 데이터 압축하기1
Day 14. 데이터 압축하기2
Day 15. 데이터 늘리기1
Day 16. 데이터 늘리기2
Day 17. 인덱스 이해하기
Day 18. SELECT 중요성
Day 19. 분석함수1
Day 20. 분석함수2
Day 21. 분석함수3
Day 22. 실전연습1
Day 23. 실전연습2
Day 24. 프로시저 만들기1
Day 25. 프로시저 만들기2
Day 26. SQL 리뷰하기

Day 9. 내장함수 CASE

■ CASE 구문

- ▶ IF-THEN-ELSE 논리와 유사한 방식으로 표현식을 작성해서 SQL의 비교 연산 기능을 보완하는 역할을 함
- ▶ 앞 장에서 설명한 단일행 내장함수에 포함됨
- ▶ 오라클 초창기 버전에서 사용했던 DECODE와 유사한 기능을 가지고 있으며 오히려 풍부한 비교연산이 가능함
→ 최근에는 오라클에서도 CASE문을 권장하고 있음

CASE 구문	설명
CASE SIMPLE_CASE_EXPRESSION 조건 ELSE 표현절 END	SIMPLE_CASE_EXPRESSION 조건이 맞으면 SIMPLE_CASE_EXPRESSION 조건 내의 THEN 절을 수행하고 조건이 맞지 않으면 ELSE 절을 수행한다
CASE SEARCHED_CASE_EXPRESSION 조건 ELSE 표현절 END	SEARCHED_CASE_EXPRESSION 조건이 맞으면 SEARCHED_CASE_EXPRESSION 조건 내의 THEN 절을 수행하고 조건이 맞지 않으면 ELSE 절을 수행한다

Day 9. 내장함수 CASE

■ SIMPLE_CASE_EXPRESSION

- ▶ CASE 다음에 바로 조건에 사용되는 컬럼이나 표현식을 표시하고, 다음 WHEN 절에서 앞에서 정의한 컬럼이나 표현식과 같은지 아닌지를 판단하는 문장으로 EQUI(=)조건만 사용한다면 SEARCHED_CASE_EXPRESSION보다 간단히 사용할 수 있는 장점이 있음 (DECODE와 기능면에서 동일)

《 1 》.

```
SELECT INVOICE_NO
       ,OUT_TYPE_DIV
       ,CASE SUBSTR(OUT_TYPE_DIV, 1, 2)
           WHEN 'M1' THEN '상온'
           WHEN 'M2' THEN '냉장'
           ELSE '기타'
       END AS TEMP
FROM LO_OUT_M
ORDER BY CASE SUBSTR(OUT_TYPE_DIV, 1, 2)
           WHEN 'M1' THEN '상온'
           WHEN 'M2' THEN '냉장'
           ELSE '기타'
END;
```

■ SEARCHED_CASE_EXPRESSION

- ▶ CASE 다음에는 컬럼이나 표현식을 표시하지 않고 WHEN 절에서 EQUI(=) 조건을 포함한 여러 조건을 이용한 조건절을 사용할 수 있기 때문에 SIMPLE_CASE_EXPRESSION보다 훨씬 다양한 조건을 적용할 수 있는 장점이 있음

《 2 》.

```
SELECT ITEM_CD
       ,ORDER_QTY
       ,CASE WHEN ORDER_QTY = 0 THEN '없음'
           WHEN ORDER_QTY BETWEEN 1 AND 10 THEN '소량'
           WHEN ORDER_QTY <= 99 THEN '보통'
           WHEN ITEM_WEIGHT = 1566 THEN '스페셜'
           ELSE '대량'
       END AS AMOUNT
FROM LO_OUT_D;
```

Day 9. 내장함수 CASE

《 3 》.

```
SELECT CASE SUBSTR(OUT_TYPE_DIV, 1, 2)
        WHEN 'M1' THEN '상온'
        WHEN 'M2' THEN '냉장'
        ELSE '기타'
END AS TEMP
,COUNT(*) AS CNT
FROM LO_OUT_M
GROUP BY CASE SUBSTR(OUT_TYPE_DIV, 1, 2)
        WHEN 'M1' THEN '상온'
        WHEN 'M2' THEN '냉장'
        ELSE '기타'
END
ORDER BY COUNT(*);
```

Day 9. 내장함수 CASE

《 4 》.

```
SELECT CASE WHEN ORDER_QTY = 0          THEN '없음'
          WHEN ORDER_QTY BETWEEN 1 AND 10 THEN '소량'
          WHEN ORDER_QTY <= 99          THEN '보통'
          WHEN ITEM_WEIGHT = 1566       THEN '스패셜'
          ELSE '대량'
        END AS AMOUNT
      ,COUNT(DISTINCT ITEM_CD) AS ITEM_CNT
FROM LO_OUT_D
GROUP BY CASE WHEN ORDER_QTY = 0          THEN '없음'
          WHEN ORDER_QTY BETWEEN 1 AND 10 THEN '소량'
          WHEN ORDER_QTY <= 99          THEN '보통'
          WHEN ITEM_WEIGHT = 1566       THEN '스패셜'
          ELSE '대량'
        END;
```

Day 9. 내장함수 CASE

《 5 》.

```
SELECT CASE TO_CHAR(OUTBOUND_DATE, 'MM') WHEN '01' THEN SET_QTY END AS M01
        ,CASE TO_CHAR(OUTBOUND_DATE, 'MM') WHEN '02' THEN SET_QTY END AS M02
        ,CASE TO_CHAR(OUTBOUND_DATE, 'MM') WHEN '03' THEN SET_QTY END AS M03
        ,CASE TO_CHAR(OUTBOUND_DATE, 'MM') WHEN '04' THEN SET_QTY END AS M04
        ,CASE TO_CHAR(OUTBOUND_DATE, 'MM') WHEN '05' THEN SET_QTY END AS M05
        ,CASE TO_CHAR(OUTBOUND_DATE, 'MM') WHEN '06' THEN SET_QTY END AS M06
        ,CASE TO_CHAR(OUTBOUND_DATE, 'MM') WHEN '07' THEN SET_QTY END AS M07
        ,CASE TO_CHAR(OUTBOUND_DATE, 'MM') WHEN '08' THEN SET_QTY END AS M08
        ,CASE TO_CHAR(OUTBOUND_DATE, 'MM') WHEN '09' THEN SET_QTY END AS M09
        ,CASE TO_CHAR(OUTBOUND_DATE, 'MM') WHEN '10' THEN SET_QTY END AS M10
        ,CASE TO_CHAR(OUTBOUND_DATE, 'MM') WHEN '11' THEN SET_QTY END AS M11
        ,CASE TO_CHAR(OUTBOUND_DATE, 'MM') WHEN '12' THEN SET_QTY END AS M12
FROM LO_OUT_M
WHERE OUTBOUND_DATE BETWEEN TO_DATE('20190101', 'YYYY-MM-DD') AND TO_DATE('20191231', 'YYYY-MM-DD');
```

주문 마스터 정보 (A_OUT_M)

BRAND_CD	INVOICE_NO	OUTBOUND_DATE	OUT_TYPE_DIV	ORDER_NM
1001	#01	2023-01-03	M11	윤현수
	#02	2023-01-03	M11	전정훈
	#03	2023-01-04	M12	고선주
	#04	2023-01-05	M12	최재원
	#05	2023-01-05	M21	권민재
2001	#01	2023-01-03	M11	강민규
	#07	2023-01-04	M21	김민기
	#08	2023-01-04	M22	김민기
	#09	2023-01-04	M22	조승완
	#10	2023-01-05	M22	진효인

상품 마스터 정보 (A_ITEM)

BRAND_CD	ITEM_CD	ITEM_NM	QTY_IN_BOX
1001	A	상품A	2
	B	상품B	2
	C	상품C	2
	D	상품D	3
	E	상품E	3
2001	A	상품A	2
	B	상품B	2
	C	상품C	2
	D	상품D	3
	E	상품E	3

주문 디테일 정보 (A_OUT_D)

BRAND_CD	INVOICE_NO	LINE_NO	ITEM_CD	ORDER_QTY
1001	#01	1	A	1
	#02	1	B	1
		2	C	3
	#03	1	B	2
	#04	1	A	1
		2	D	1
		3	E	2
	#05	1	C	5
	#01	1	A	1
		2	B	2
2001	#07	1	E	1
	#08	1	C	1
	#09	1	B	3
		2	D	1
	#10	1	E	1

- 출고유형이 M1로 시작하면 상온, M2로 시작하면 저온이라고 표시해 줘! (OUTBOUND_DATE, DY[요일], INVOICE_NO, EVENODD[홀/짝], ORDER_NM, TEMP[상온/저온])
- 주문수량이 1~2이면 '하', 3~4이면 '중', 5이상이면 '상'으로 표시해 줘! (BRAND_CD, INVOICE_NO, LINE_NO, ITEM_CD, ORDER_QTY, GRADE[상/중/하])

Day 9. 내장함수 CASE

★SQL문형 익히기 - 5

CASE문 적용 GROUP BY ▶ SELECT CASE문's, 집계함수 FROM WHERE GROUP BY CASE문's

주문 마스터 정보 (A_OUT_M)

BRAND_CD	INVOICE_NO	OUTBOUND_DATE	OUT_TYPE_DIV	ORDER_NM
1001	#01	2023-01-03	M11	윤현수
	#02	2023-01-03	M11	전정훈
	#03	2023-01-04	M12	고선주
	#04	2023-01-05	M12	최재원
	#05	2023-01-05	M21	권민재
2001	#01	2023-01-03	M11	강민규
	#07	2023-01-04	M21	김민기
	#08	2023-01-04	M22	김민기
	#09	2023-01-04	M22	조승완
	#10	2023-01-05	M22	진효인

상품 마스터 정보 (A_ITEM)

BRAND_CD	ITEM_CD	ITEM_NM	QTY_IN_BOX
1001	A	상품A	2
	B	상품B	2
	C	상품C	2
	D	상품D	3
	E	상품E	3
2001	A	상품A	2
	B	상품B	2
	C	상품C	2
	D	상품D	3
	E	상품E	3

주문 디테일 정보 (A_OUT_D)

BRAND_CD	INVOICE_NO	LINE_NO	ITEM_CD	ORDER_QTY
1001	#01	1	A	1
	#02	1	B	1
		2	C	3
	#03	1	B	2
	#04	1	A	1
		2	D	1
		3	E	2
#05	1	C	5	
2001	#01	1	A	1
		2	B	2
	#07	1	E	1
	#08	1	C	1
	#09	1	B	3
		2	D	1
	#10	1	E	1

- [브랜드] & [상온/저온]별로 몇 개의 인보이스를 처리했는지 가르쳐 줘!
- [브랜드] & [상/중/하(인보이스 단위의 합계)]별로 몇 개의 인보이스를 처리했는지 가르쳐 줘!
- [상품]별로 주문수량의 합계를 구한 다음, TOP2까지는 그대로 표시하고, 나머지 상품들은 etc로 묶어서 표시해 줘!

주문 마스터 정보 (A_OUT_M)

BRAND_CD	INVOICE_NO	OUTBOUND_DATE	OUT_TYPE_DIV	ORDER_NM
1001	#01	2023-01-03	M11	윤현수
	#02	2023-01-03	M11	전정훈
	#03	2023-01-04	M12	고선주
	#04	2023-01-05	M12	최재원
	#05	2023-01-05	M21	권민재
2001	#01	2023-01-03	M11	강민규
	#07	2023-01-04	M21	김민기
	#08	2023-01-04	M22	김민기
	#09	2023-01-04	M22	조승완
	#10	2023-01-05	M22	진효인

상품 마스터 정보 (A_ITEM)

BRAND_CD	ITEM_CD	ITEM_NM	QTY_IN_BOX
1001	A	상품A	2
	B	상품B	2
	C	상품C	2
	D	상품D	3
	E	상품E	3
2001	A	상품A	2
	B	상품B	2
	C	상품C	2
	D	상품D	3
	E	상품E	3

주문 디테일 정보 (A_OUT_D)

BRAND_CD	INVOICE_NO	LINE_NO	ITEM_CD	ORDER_QTY
1001	#01	1	A	1
	#02	1	B	1
		2	C	3
	#03	1	B	2
	#04	1	A	1
		2	D	1
		3	E	2
	#05	1	C	5
	#01	1	A	1
		2	B	2
2001	#07	1	E	1
	#08	1	C	1
	#09	1	B	3
		2	D	1
	#10	1	E	1
		1	E	1

- 1001 브랜드의 주문내역을 표시하되, C 상품을 가장 먼저 표시하고, 나머지 상품은 상품코드 순으로 나열해 줘!
(BRAND_CD, INVOICE_NO, LINE_NO, ITEM_CD, ORDER_QTY, GRADE[상/중/하])
단, 동일한 상품에 대해서는 주문수량이 많은 것부터 나열해 줘!

Day 9. 내장함수 CASE

실전문제① ► WHERE절, SELECT절에 CASE문 적용하기

《테이블》	■ LO_OUT_M(출고주문)	■		
《조건》	■ OUTBOUND_DATE(출고일자) ■ :COND	► 2019년 9월 19일 ► 0 또는 1 또는 2		
《정렬》	■			
《특징》	■ 조건으로 주어진 출고일자의 해당 월 1일 부터 조건으로 주어진 출고일자 까지를 실제 조건으로 사용하기 ■ :COND 값에 따라 OUT_BOX_DIV(출고박스유형)의 검색조건이 달라짐 → [0] 전체, [1] 'D'로 시작하는 모든 레코드, [2] 'F'로 시작하는 모든 레코드, [그 외] 조회되지 않음 ■ M1_CNT → 출고유형구분(OUT_TYPE_DIV)가 'M1'로 시작하는 레코드의 카운트 ■ M2_CNT → 출고유형구분(OUT_TYPE_DIV)가 'M2'로 시작하는 레코드의 카운트			

결과 ▼ 총 건수 : 13건

OUTBOUND_DATE	INVOICE_CNT	M1_CNT	M2_CNT
2019/09/01	554	550	4
2019/09/02	9103	9011	85
2019/09/03	4727	4654	73
2019/09/04	7238	7204	34
2019/09/05	5461	5439	22
2019/09/06	3951	3919	32
2019/09/08	4433	4407	26

Day 9. 내장함수 CASE

실전문제② ▶ 이상한 데이터 찾아내기

《테이블》	■ LO_OUT_D(출고주문상세)	■	■
《조건》	■	▶	
《정렬》	■ 중복되는 상품코드(ITEM_CD)의 개수의 내림차순으로 정렬하되, 상품바코드(ITEM_BAR_CD)가 NULL인 것은 맨 마지막에 표시함		
《특징》	■ 상품코드(ITEM_CD)는 서로 다른데, 상품바코드(ITEM_BAR_CD)가 동일하여 바코드 스캔 시 상품이 2개 이상 선택될 수 있는 상품바코드(ITEM_BAR_CD)와 중복되는 상품코드(ITEM_CD)의 개수를 구함		

결과 ▼ 총 건수 : 119건

ITEM_BAR_CD	ITEM_CNT
8801047161578	4
8801047317012	3
8801155737511	3
3073781087055	3
8801047315493	2
8801043049207	2
8801005305839	2
8801128503174	2
	669

Day 9. 내장함수 CASE

실전문제③ ▶ 전형적인 SUM(CASE()) 구문 사용하기

《테이블》	■ LO_OUT_M(출고주문)	■	■
《조건》	■ OUTBOUND_DATE(출고일자)	▶ 2019년 전체	
《정렬》	■		
《특징》	■ SEQ_QTY의 Summary값을 해당 월에 해당하는 컬럼에 표시하시오. ■ 1월부터 12월을 옆으로 나열함		

결과 ▼ 총 건수 : 1건

M01	M02	M03	M04	M05	M06	M07	M08	M09	M10	M11	M12
					50973	36740	50194	43727			

Day 9. 내장함수 CASE

실전문제④ ▶ 일자별 SUM(CASE()) 구문 사용하기

《테이블》	■ LO_OUT_M(출고주문)	■	■
《조건》	■ OUTBOUND_DATE(출고일자)	▶ 2019년 전체	
《정렬》	■		
《특징》	■ 출고일자 단위로 SET_QTY의 Summary값을 해당 월에 해당하는 컬럼에 표시하시오. ■ 1월부터 12월을 옆으로 나열함		

결과 ▼ 총 건수 : 85건

OUTBOUND_DATE	M01	M02	M03	M04	M05	M06	M07	M08	M09	M10	M11	M12
2019/06/03						2748						
2019/06/04						3033						
2019/06/05						3291						
2019/06/07						2486						
2019/06/08						25						
2019/06/10						8227						
2019/06/11						2891						
2019/06/12						2369						

Day 9. 내장함수 CASE

실전문제⑤ ▶ 연도별 SUM(CASE()) 구문 사용하기

《테이블》	■ LO_OUT_M(출고주문)	■	■
《조건》	■	▶	
《정렬》	■		
《특징》	■ 출고연도 단위로 SEQ_QTY의 Summary값을 해당 월에 해당하는 컬럼에 표시함 ■ 1월부터 12월을 옆으로 나열함		

결과 ▼ 총 건수 : 2건

YYYY	M01	M02	M03	M04	M05	M06	M07	M08	M09	M10	M11	M12
2018						152919	110220	150582	131181			
2019						50973	36740	50194	43727			

Day 9. 내장함수 CASE

실전문제⑥ ▶ 조건 값(변수)에 따른 다른 정렬기준 적용하기

《테이블》	■ LO_OUT_D(출고주문상세)	■	■
《조건》	■ INVOICE_NO(송장번호)	▶ 346724706214	
《정렬》	■		
《특징》	■ SORT_TYPE 변수가 1이면 LINE_NO 컬럼으로 정렬하고 ■ SORT_TYPE 변수가 2이면 먼저 ORDER_QTY 컬럼으로 정렬하고 값이 동일하면 LINE_NO 컬럼으로 정렬하기		

결과 ▼ 총 건수 : 3건

INVOICE_NO	LINE_NO	ITEM_CD	ITEM_NM	ORDER_QTY
346724706214	1	64209	뉴트리플랜 건강프로젝트-장 160g	36
346724706214	2	64212	뉴트리플랜 건강프로젝트-항산화 160g	8
346724706214	4	64210	뉴트리플랜 건강프로젝트-피부모질 160g	36

INVOICE_NO	LINE_NO	ITEM_CD	ITEM_NM	ORDER_QTY
346724706214	2	64212	뉴트리플랜 건강프로젝트-항산화 160g	8
346724706214	1	64209	뉴트리플랜 건강프로젝트-장 160g	36
346724706214	4	64210	뉴트리플랜 건강프로젝트-피부모질 160g	36

Thank you !

ASETEC Location <http://www.asetec.co.kr>

본사. 경기도 성남시 분당구 성남대로 331번길 8, 킨스타워 2201호 TEL.031.609.7000 FAX.031.609.7009
부산. 부산광역시 해운대구 센텀동로 99 TEL.051.506.6352 FAX.051.504.8794