

# SQL Pogramming

## *- Day 18 -*

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 18. SELECT 중요성

## ■ 왜 SELECT 구문을 잘해야 하는 것일까?

- ▶ SELECT는 말 그대로 데이터를 검색하는 구문임
- ▶ 빠른 검색은 SQL의 모든 DML 구문 들에서 핵심적인 요소임
- ▶ 테이블 생성
  - 테이블을 생성하는 것과 SELECT를 잘해야 하는 것과 무슨 상관 관계가 있는 것일까?
  - 일반적인 온라인 응용 프로그램을 개발하는 상황에서는 거의 발생하지 않지만, 특히 프로그램을 개발하는 초기에는 임시 테이블을 생성하여 활용하는 상황이 다수 발생함
  - CREATE TABLE ... AS SELECT ... 구문은 아주 유용함
  - 테이블명/컬럼명 이외에는 모두 DEFAULT(PK 지정 안됨)
- ▶ 데이터 생성
  - 일반적인 유저 인터페이스(화면)을 통해 데이터를 입력받는 경우에는 해당되지 않음. INSERT INTO... VALUES...
  - 그러나 어떤 이벤트에 의해, 다수의 테이블을 참조해 특정 테이블에 데이터를 생성하는 상황이 많이 발생함
  - INSERT INTO ... SELECT ...
  - 이 구문에서의 SELECT 역할은 상당히 높음

## ▶ 데이터 수정

- 데이터를 수정하는 UPDATE 구문에서도 SELECT는 여전히 많이 활용됨
- UPDATE 테이블명  
SET (컬럼3, 컬럼4) = (SELECT ...)  
WHERE (컬럼1, 컬럼2) IN (SELECT ...)
- 위의 구문과 같이 수정할 테이블에서 수정 대상이 되는 컬럼을 빠르게 검색해야 하고, 더구나 수정할 컬럼의 값을 찾는 것도 SELECT 구문을 이용하여 빠르게 검색해야 함

## ▶ 데이터 삭제

- 데이터를 수정하거나 삭제하는 행위의 수가 되는 수정/삭제는 대상이 되는 건수가 많다면 어쩔 수 없이 부하가 발생할 수 있지만, 그렇지 않다면 수정/삭제의 대상이 되는 건들을 빠르게 검색하는 것이 훨씬 중요한 일이 되기도 함
- DELETE FROM 테이블명  
WHERE (컬럼1, 컬럼2) IN (SELECT ...)

# Day 18. SELECT 중요성

## ■ CREATE TABLE ... AS SELECT ... 구문

```
CREATE TABLE Z_DAY17_1
AS
SELECT M1.OUTBOUND_DATE AS OUTBOUND_DATE
      ,SUM(M2.ORDER_QTY) AS SUM_QTY
      ,0                  AS UPDATE_COL1
      ,0                  AS UPDATE_COL2
FROM LO_OUT_M M1
     JOIN LO_OUT_D M2 ON M2.INVOICE_NO = M1.INVOICE_NO
WHERE OUTBOUND_DATE BETWEEN TO_DATE('20190901', 'YYYY-MM-DD') AND TO_DATE('20190930', 'YYYY-MM-DD')
GROUP BY M1.OUTBOUND_DATE
ORDER BY SUM(M2.ORDER_QTY) DESC
```

# Day 18. SELECT 중요성

## ■ INSERT ... SELECT ... 구문

```
INSERT INTO Z_DAY17_1
SELECT M1.OUTBOUND_DATE AS OUTBOUND_DATE
      ,SUM(M2.ORDER_QTY) AS SUM_QTY
      ,0                  AS UPDATE_COL1
      ,0                  AS UPDATE_COL2
FROM LO_OUT_M M1
      JOIN LO_OUT_D M2 ON M2.INVOICE_NO = M1.INVOICE_NO
WHERE OUTBOUND_DATE BETWEEN TO_DATE('20190801', 'YYYY-MM-DD') AND TO_DATE('20190831', 'YYYY-MM-DD')
GROUP BY M1.OUTBOUND_DATE
ORDER BY SUM(M2.ORDER_QTY) DESC
```

# Day 18. SELECT 중요성

## ■ UPDATE ... SELECT ... 구문

```
UPDATE Z_DAY17_1 U1
  SET (U1.UPDATE_COL1, U1.UPDATE_COL2) = (
    SELECT MAX(M2.ORDER_QTY)
           ,MIN(M2.ORDER_QTY)
    FROM LO_OUT_M M1
         JOIN LO_OUT_D M2 ON M2.INVOICE_NO = M1.INVOICE_NO
    WHERE M1.OUTBOUND_DATE = U1.OUTBOUND_DATE
  )

WHERE U1.OUTBOUND_DATE IN (
  SELECT M1.OUTBOUND_DATE
    FROM LO_OUT_M M1
         JOIN LO_OUT_D M2 ON M2.INVOICE_NO = M1.INVOICE_NO
    WHERE M1.OUTBOUND_DATE BETWEEN TO_DATE('20190801', 'YYYY-MM-DD') AND
                                   TO_DATE('20190930', 'YYYY-MM-DD')
    GROUP BY M1.OUTBOUND_DATE
    HAVING SUM(M2.ORDER_QTY) > 200000
  )
```

※ Updatable Join View

→ 여기서는 설명 Pass

# Day 18. SELECT 중요성

## 실전문제① ► UPDATE의 SET절에서 서브쿼리 사용하기

《테이블》	■ Z_DAY17_1	■	■
《조건》	■ OUTBOUND_DATE(출고일자)	▶ 2019년 8월	
《정렬》	■		
《특징》	■ UPDATE 컬럼 → UPDATE_COL2 ■ UPDATE 내용 → 2019년 8월 레코드를 대상으로 자신의 출고일자에 해당하는 요일과 일치하는 모든 출고일자의 SUM_QTY의 Summary 값으로 UPDATE (단, 자기자신의 출고일자는 계산에서 제외)		

결과 ▼ 총 건수 : 22건 updated

22 rows updated

OUTBOUND_DATE	SUM_QTY	UPDATE_COL1	UPDATE_COL2
2019/08/01	92575	0	449001
2019/08/02	93912	0	513176
2019/08/05	164048	0	562571
2019/08/06	93563	0	343010
2019/08/07	80502	0	380181
2019/08/08	81046	0	460530
2019/08/09	96467	0	510621

# Day 18. SELECT 중요성

## 실전문제② ► DELETE 조건절에서 서브쿼리 사용하기

《테이블》	■ Z_DAY17_1	■ LO_OUT_M(출고주문)	■ LO_OUT_D(출고주문상세)
《조건》	■ OUTBOUND_DATE(출고일자)	▶ 2019년 8월	
《정렬》	■		
《특징》	■ 삭제조건 → LO_OUT_M(출고주문), LO_OUT_D(출고주문상세) 테이블을 참조하여 SUM_QTY가 2019년 7월의 일평균 출고량보다 적은 레코드는 모두 삭제하기		

결과 ▼ 총 건수 : 11건 deleted

11 rows deleted

OUTBOUND_DATE	SUM_QTY	UPDATE_COL1	UPDATE_COL2
2019/08/05	164048	0	0
2019/08/12	193010	0	0
2019/08/13	141958	0	0
2019/08/16	160120	0	0
2019/08/19	164622	0	0
2019/08/21	153979	0	0
2019/08/22	174854	0	0
2019/08/26	204939	0	0
2019/08/28	155249	0	0
2019/08/29	193101	0	0
2019/08/30	136972	0	0



# 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