

# SQL Pogramming

## - *Day 3* -

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

## ■ DDL (Data Definition Language) - 데이터 정의어

▶ 테이블과 컬럼을 정의하는 명령어로 생성, 수정, 삭제 등의

데이터 전체 골격을 결정하는 역할을 담당

▶ 명령어를 입력하는 순간, 작업이 즉시 반영되기 때문에 주의하여 실행해야 함

## ■ DDL의 종류

- ▶ CREATE → 테이블을 생성하는 명령어
- ▶ ALTER → 테이블의 구조를 수정하는 명령어
- ▶ DROP → 테이블 자체를 삭제하는 명령어 (데이터 포함)
- ▶ RENAME → 테이블의 이름을 변경하는 명령어
- ▶ TRUNCATE → 테이블의 데이터를 초기화하는 명령어

## ■ 테이블 생성

- ▶ CREATE TABLE 테이블명  
(컬럼명1 데이터타입 DEFAULT 디폴트값 NULL/NOT NULL, ...  
컬럼명n 데이터타입 DEFAULT 디폴트값 NULL/NOT NULL);  
→ CREATE TABLE 테이블명 AS SELECT구문 가능
- ▶ DEFAULT값과 NULL(NOT NULL) 지정은 옵션 항목임
- ▶ 기타 옵션 부분은 이번 교육에서 언급하지 않음

## ■ 인덱스 생성

- ▶ CREATE UNIQUE INDEX 인덱스명 ON 테이블명  
(컬럼명1, ..., 컬럼명n);
- ▶ UNIQUE는 옵션 항목임 (레코드 중복 불허)
- ▶ 기타 옵션 부분은 이번 교육에서 언급하지 않음

## ■ 기본키 (Primary Key) 생성

- ▶ ALTER TABLE 테이블명 ADD CONSTRAINT PK명  
PRIMARY KEY (컬럼명1, ..., 컬럼명n)

## ■ 코멘트 (COMMENT) 생성

- ▶ 테이블 코멘트 생성
  - COMMENT ON TABLE 테이블명 IS '테이블명 설명';
- ▶ 컬럼 코멘트 생성
  - COMMENT ON COLUMN 테이블명.컬럼명 IS '컬럼 설명';

## ■ 기타

- ▶ ALTER, DROP, RENAME, TRUNCATE 명령은 추후 교육 (참고서적 참조)

## ■ DML (Data Manipulation Language) - 데이터 조작어

- ▶ 데이터를 생성, 수정, 삭제 및 조회를 위한 명령어
- ▶ INSERT / UPDATE / DELETE / MERGE / SELECT

### ■ INSERT 구문

- ▶ 테이블에 데이터를 생성하는 명령어
- ▶ 데이터를 1건 씩 입력하는 방법
  - ① **INSERT INTO** 테이블명  
**VALUES** (전체 컬럼들의 값을 기재);
  - ② **INSERT INTO** 테이블명 (값을 입력할 컬럼들을 기재)  
**VALUES** (값을 입력할 컬럼들의 값을 기재);
- ▶ 데이터를 1건 이상 입력하는 방법
  - ③ **INSERT INTO** 테이블명  
**SELECT** 구문;
  - ④ **INSERT INTO** 테이블명 (값을 입력할 컬럼들을 지정)  
**SELECT** 구문;
- ▶ INSERT 구문은 문법적으로 간단함
- ▶ 데이터를 1건 생성하는 구문의 실행은 최소시간 소요

- ▶ 데이터를 수백 건 이상 생성하기 위해 데이터를 1건 생성하는

INSERT 구문을 FOR문을 통해 수백 번 실행하는 행위는 시스템을 병들게 하는 지름길

- ▶ INSERT가 개발 측면에서 가장 중요한 부분은 INSERT와 SELECT 구문을 조합하여 복수 개의 레코드를 추출하고 동시에 지정된 테이블에 데이터를 생성하는 능력
- ▶ SELECT 구문의 중요성을 첫 번째로 강조합니다.

## ■ UPDATE 구문

- ▶ 이미 생성된 데이터를 컬럼 단위로 변경하는 명령어
- ▶ UPDATE 구문에 의해 변경되는 데이터는 1건 이상
- ▶ 기본적인 UPDATE 구문

### ○ UPDATE 테이블명

SET 컬럼명1 = 변경할 값1  
    , 컬럼명2 = 변경할 값2

...

WHERE <SELECT 구문의 WHERE절과 동일>;

- ▶ 업그레이드된 UPDATE 구문

### ○ UPDATE 테이블명

SET (컬럼명1, 컬럼명2, ...) =  
    (SELECT 구문)

WHERE <SELECT 구문의 WHERE절과 동일>;

- ▶ WHERE절은 옵션이며, 기술하지 않을 시, 해당 테이블의 모든 데이터를 수정하게 됨
- ▶ UPDATE 구문의 실행에서 레코드의 값을 변경하는 행위

는  
생산성에 큰 영향을 미치지 않지만, 변경할 대상을 찾는  
검색  
시간이 생산성에 가장 큰 영향을 미침

- ▶ UPDATE가 개발 측면에서 가장 중요한 부분은 UPDATE와

SELECT 구문을 조합하여 복수 개의 레코드를 추출하여  
추출된 레코드의 데이터를 변경하는 능력

- ▶ SELECT 구문의 중요성을 두 번째로 강조합니다.

## ■ DELETE 구문

- ▶ 이미 생성된 데이터 중에서 조건에 부합하는 레코드들만 삭제하는 명령어
- ▶ DELETE [FROM] 테이블명  
    WHERE <SELECT 구문의 WHERE절과 동일>;
- ▶ WHERE절은 옵션이며, 기술하지 않을 시, 해당 테이블의 모든 데이터를 삭제하게 됨
- ▶ DELETE 구문의 실행에서 레코드를 삭제하는 행위는 생산성에 큰 영향을 미치지 않지만, 삭제할 대상을 찾는 검색 시간이 생산성에 가장 큰 영향을 미침

## ■ MERGE 구문

- ▶ 데이터를 INSERT or UPDATE하는 명령어 -참고서적 참조

## ■ SELECT 구문

- DML 구문들 중에서 가장 기본이 되는 구문이며  
SELECT 자체로도 상당히 중요한 위치에 있지만, 다른  
구문들을 자유자재로 구사하기 위해서는 SELECT 구문  
을 집중적으로 공부하는 것이 가장 좋은 방법
- ▶ SELECT 구문을 거의 완벽하게 구사할 수 있다면  
INSERT / UPDATE / DELETE / MERGE 구문도 덩달아  
완벽하게 구사할 수 있는 토대가 됨
  - ▶ 이 과목의 교육에서 다룰 대부분의 내용이 SELECT 구문  
의 구사 능력을 향상시키기 위한 것

## ■ SELECT 유형1

```
SELECT *  
FROM 테이블;
```

## ■ SELECT 유형2

```
SELECT [DISTINCT] 컬럼1, ..., 컬럼n  
FROM 테이블;
```

## ■ SELECT 유형3

```
SELECT [DISTINCT] 컬럼1, ..., 컬럼n  
FROM 테이블  
WHERE 컬럼1 = 조건값1  
AND 컬럼2 = 조건값2;
```

## ■ SELECT 유형4

```
SELECT [DISTINCT] 컬럼1, ..., 컬럼n  
FROM 테이블  
WHERE 컬럼1 = 조건값1  
OR 컬럼2 = 조건값2;
```

## ■ SELECT 유형5

```
SELECT [DISTINCT] 컬럼1, ..., 컬럼n  
FROM 테이블  
WHERE 컬럼1 = 조건값1  
AND 컬럼2 IN (조건값2-1, ..., 조건값2-n)  
AND 컬럼3 BETWEEN 조건값3-1 AND 조건값3-2  
AND 컬럼4 LIKE 조건값4 || '%' ;
```

## ■ SELECT 유형6

```
SELECT [DISTINCT] 별칭1.컬럼1, ..., 별칭1.컬럼n
        , 별칭2.컬럼1, ..., 별칭2.컬럼n
FROM 테이블1 별칭1
    JOIN 테이블2 별칭2 ON 별칭2.컬럼1 = 별칭1.컬럼1
                        AND 별칭2.컬럼2 = 별칭1.컬럼2
WHERE 별칭1.컬럼1 = 조건값1
    AND 별칭1.컬럼2 IN (조건값1, ..., 조건값n)
ORDER BY 별칭1.컬럼3, 별칭2.컬럼5;
```

## ■ SELECT 유형7

```
SELECT 별칭1.컬럼1, 별칭2.컬럼3, SUM(별칭2.컬럼5)
FROM 테이블1 별칭1
    JOIN 테이블2 별칭2 ON 별칭2.컬럼1 = 별칭1.컬럼1
                        AND 별칭2.컬럼2 = 별칭1.컬럼2
WHERE 별칭1.컬럼1 = 조건값1
    AND 별칭1.컬럼2 IN (조건값1, ..., 조건값n)
GROUP BY 별칭1.컬럼1, 별칭2.컬럼3
ORDER BY 별칭1.컬럼1, 별칭2.컬럼3;
```

## ■ SELECT 유형8

```
SELECT 별칭1.컬럼1, 별칭2.컬럼3, SUM(별칭2.컬럼5)
        , (SELECT 구문)
FROM (SELECT 구문) 별칭1
    JOIN 테이블2 별칭2 ON 별칭2.컬럼1 = 별칭1.컬럼1
                        AND 별칭2.컬럼2 = 별칭1.컬럼2
WHERE 별칭1.컬럼1 = 조건값1
    AND 별칭1.컬럼2 IN (조건값1, ..., 조건값n)
    AND 별칭1.컬럼4 IN (SELECT 구문)
GROUP BY 별칭1.컬럼1, 별칭2.컬럼3
ORDER BY 별칭1.컬럼1, 별칭2.컬럼3;
```

## ■ 기억해 두어야 할 중요한 포인트

- ▶ *Reserved Word(예약어) 이외의 컬럼, 조건값 등이 사용되는 모든 위치에는 내장함수(특히 CASE문)를 적용하여 가공 컬럼의 형태로 활용할 수 있음*

## ■ DCL (Data Control Language) – 데이터 제어어

- ▶ GRANT(권한주기), REVOKE(권한빼기)

## ■ TCL (Transaction Control Language) – 트랜잭션 제어어

- ▶ COMMIT, ROLLBACK



# Thank you !

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

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

FAX.031.609.7009