

위의 정리한 내용을 토대로 트리거를 만들어줘

아래는 앞서 설계한 내용을 토대로 MSSQL 환경에서 바로 사용할 수 있는 트리거 예제입니다.
실제 MMS_LOG_YYYYMM 테이블명과 인덱스 상황에 맞게 적절히 수정하시고, 트리거 실행시 성능 영향을 꼭 검증해 주세요.

```
CREATE TRIGGER TRG_MMS_MSG_DUP_CHECK
ON MMS_MSG
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    -- 새로 삽입된 각 레코드에 대해 중복 여부 체크 및 상태 변경 수행
    UPDATE M
    SET status = 3
    FROM MMS_MSG M
    INNER JOIN inserted I ON M.msg_id = I.msg_id
    WHERE
        -- MMS_MSG 내 동일 phone/msg 중복 존재: 자신 제외
        EXISTS (
            SELECT 1
            FROM MMS_MSG
            WHERE phone = I.phone
            AND msg = I.msg
            AND msg_id <> I.msg_id
        )
    OR
        -- MMS_LOG 최근 2개월 테이블에서 동일 phone/msg 존재
        EXISTS (
            SELECT 1 FROM MMS_LOG_202506 L WHERE L.phone = I.phone AND L.msg = I.msg
        )
    OR
        EXISTS (
            SELECT 1 FROM MMS_LOG_202507 L WHERE L.phone = I.phone AND L.msg = I.msg
        );
END
```

주요 참고 사항

- `inserted` 테이블은 트리거가 처리하는 새로 들어온 레코드를 의미합니다.
- `MMS_LOG` 테이블 명(여기선 202506, 202507)은 현재 월 기준으로 1~2개월 전 테이블명으로 변경해야 하며, 운영시에는 동적으로 변경하거나 월별로 트리거를 수정하여 관리해야 합니다.
- `msg_id`는 `MMS_MSG` 테이블의 기본키(또는 고유 식별자) 컬럼명입니다. 실제 컬럼명에 따라 변경해주세요.
- 중복 판단 기준인 `phone`, `msg` 컬럼에 인덱스가 있으면 성능이 개선됩니다.

- 트리거 안에서 너무 많은 데이터를 조회하는 경우 성능 저하가 발생할 수 있으니, 사용 전에 충분한 테스트와 인덱스 최적화를 권장합니다.

필요 시 Oracle, MySQL 등 다른 DBMS용 트리거 예제도 만들어 드릴 수 있으니 알려주세요!