

DML(Data Manipulation Language) 개념서

이 문서는 SQLD 시험을 준비하는 데 필요한 DML(Data Manipulation Language)의 핵심 개념과 사용법을 다룹니다. DML의 기본 개념부터 INSERT, UPDATE, DELETE, MERGE 문의 상세한 설명, 주의사항, 성능 고려사항, 그리고 트랜잭션 관리까지 포괄적으로 다룹니다. 또한 SQLD 시험 대비를 위한 팁과 학습 전략을 제공합니다.

1. DML의 기본 개념

1.1 정의

- 데이터베이스 내의 데이터를 조작하는 언어
- 데이터의 검색, 삽입, 수정, 삭제 기능 제공
- SELECT, INSERT, UPDATE, DELETE가 대표적

1.2 특징

- 데이터 조작의 기본 단위
 - 테이블의 행(ROW) 단위로 조작
 - 트랜잭션의 대상이 됨
- 실시간 조작
 - 즉시 데이터베이스에 영향을 미침
 - COMMIT/ROLLBACK으로 제어 가능

2. INSERT 문

2.1 기본 문법

```
-- 단일 행 삽입
INSERT INTO 테이블명 [(컬럼1, 컬럼2, ...)]
VALUES (값1, 값2, ...);
```

```
-- 다중 행 삽입
INSERT INTO 테이블명 [(컬럼1, 컬럼2, ...)]
SELECT 문;
```

2.2 주요 사용 패턴

```
-- 모든 컬럼에 데이터 입력
INSERT INTO 직원
VALUES (1001, '홍길동', '개발부', 5000);

-- 특정 컬럼만 데이터 입력
INSERT INTO 직원 (직원번호, 이름)
VALUES (1002, '김철수');
```

```
-- 서브쿼리를 이용한 다중 행 입력
INSERT INTO 신입사원 (직원번호, 이름, 부서)
SELECT 직원번호, 이름, 부서
FROM 지원자
WHERE 합격여부 = 'Y';
```

3. UPDATE 문

3.1 기본 문법

```
UPDATE 테이블명
SET 컬럼1 = 값1 [, 컬럼2 = 값2, ...]
[WHERE 조건식];
```

3.2 주요 사용 패턴

```
-- 단순 업데이트
UPDATE 직원
SET 급여 = 6000
WHERE 직원번호 = 1001;

-- 다중 컬럼 업데이트
UPDATE 직원
SET 급여 = 급여 * 1.1,
    부서 = '영업부'
WHERE 직급 = '과장';

-- 서브쿼리를 이용한 업데이트
UPDATE 직원
SET 급여 = (SELECT AVG(급여) FROM 직원)
WHERE 부서 = '개발부';
```

4. DELETE 문

4.1 기본 문법

```
DELETE [FROM] 테이블명
[WHERE 조건식];
```

4.2 주요 사용 패턴

```
-- 조건부 삭제
DELETE FROM 직원
WHERE 부서 = '영업부';

-- 전체 데이터 삭제
DELETE FROM 직원;

-- 서브쿼리를 이용한 삭제
DELETE FROM 직원
WHERE 부서 IN (SELECT 부서코드
                FROM 부서
                WHERE 폐지여부 = 'Y');
```

5. MERGE 문

5.1 기본 문법

```
MERGE INTO 대상테이블
USING 원본테이블
ON (조건식)
WHEN MATCHED THEN
    UPDATE SET 컬럼1 = 값1, ...
WHEN NOT MATCHED THEN
    INSERT (컬럼1, 컬럼2, ...)
VALUES (값1, 값2, ...);
```

5.2 활용 예제

```
-- 기존 데이터는 갱신, 신규는 삽입
MERGE INTO 실제고객 T
USING 잠재고객 S
ON (T.고객번호 = S.고객번호)
WHEN MATCHED THEN
    UPDATE SET T.연락처 = S.연락처, T.주소 = S.주소
WHEN NOT MATCHED THEN
    INSERT (고객번호, 이름, 연락처, 주소)
VALUES (S.고객번호, S.이름, S.연락처, S.주소);
```

6. DML 관련 주의사항

6.1 제약조건 고려

- PRIMARY KEY**
 - 중복된 값 입력 불가
 - NULL 값 입력 불가
- FOREIGN KEY**
 - 참조 테이블에 존재하는 값만 입력 가능
 - 참조되는 데이터 삭제 시 주의
- NOT NULL**
 - NULL 값 입력 불가
 - 반드시 값을 지정해야 함

6.2 암시적 형변환

```
-- 숫자형 컬럼에 문자형 데이터 입력
INSERT INTO 테이블 (숫자컬럼)
VALUES ('123'); -- 자동 형변환

-- 날짜형 컬럼에 문자형 데이터 입력
UPDATE 테이블
SET 날짜컬럼 = '2024-01-01'; -- 자동 형변환
```

7. 성능 고려사항

7.1 대량 데이터 처리

배치 처리

```
-- 다중 행 입력 활용
INSERT INTO 테이블
SELECT * FROM 소스테이블
WHERE 조건식;
```

커밋 주기

- 적절한 단위로 커밋
- 메모리 사용량 고려

7.2 인덱스 영향

- INSERT**
 - 인덱스가 많을수록 성능 저하
 - 인덱스 정렬 필요
- UPDATE**
 - 인덱스 컬럼 수정 시 성능 저하
 - 인덱스 재구성 발생
- DELETE**
 - 인덱스 엔트리 삭제 필요
 - 대량 삭제 시 TRUNCATE 고려

8. 트랜잭션 관리

8.1 기본 개념

```
-- 트랜잭션 시작
START TRANSACTION;

-- DML 문장들
INSERT INTO 테이블1 VALUES (...);
UPDATE 테이블2 SET ...;
```

```
-- 트랜잭션 종료
COMMIT; -- 변경사항 저장
ROLLBACK; -- 변경사항 취소
```

8.2 주의사항

- 트랜잭션 범위**
 - 최소한의 범위 설정
 - 길어지면 락 경쟁 발생
- 정합성 보장**
 - 관련 DML을 하나의 트랜잭션으로 처리
 - 데이터 일관성 유지

SQLD 시험 대비 TIP

주요 출제 포인트

- DML 문법**
 - 기본 문법 숙지
 - 다양한 사용 패턴
- 제약조건**
 - 제약조건과 DML 관계
 - 예러 발생 상황
- 성능**
 - 대량 데이터 처리
 - 인덱스 영향

학습 전략

- 기본 문법 완벽 숙지
- 다양한 예제 실습
- 제약조건 영향 이해
- 성능 관련 내용 학습

실전 문제 유형

- 구문 오류 찾기
- 실행 결과 예측
- 제약조건 위배 상황
- 성능 개선 방안