

[5주차] Node JS to TS

Ch7.4 ~ 7.5 / 배운선



목차

- 데이터베이스란?

1

- 데이터베이스 및 테이블 생성하기

데이터베이스 생성하기

테이블 생성하기

기타(컬럼 자료형 / 옵션 / 테이블 설정)

2

- CRUD 작업하기

CREATE

READ

UPDATE

DELETE

3

데이터베이스란?

관련성을 가지며 중복이 없는 데이터들의 집합



MySQL

대표적인 관계형 DBMS



MongoDB

대표적인 NoSQL

데이터베이스 및 테이블 생성하기

데이터베이스 생성하기

형식 : **CREATE SCHEMA [데이터베이스명]**

```
baeys — mysql -u root — 80x24
Last login: Sat Nov  8 19:47:54 on ttys003
baeys@Baeysui-MacBookAir ~ % mysql -u root
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 9.5.0 Homebrew

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE SCHEMA `nodejs`DEFAULT CHARACTER SET utf8mb4 DEFAULT COLLATE utf8mb4_general_ci;
Query OK, 1 row affected (0.003 sec)

mysql> use nodejs
Database changed
```

📌 MySQL 8버전의 기본 COLLATE는 utf8mb4_0900_ai_ci이지만 한글문제가 있음 → **utf8mb4_general_ci** 사용

📌 SQL구문 마지막엔 세미콜론(;) 작성

📌 예약어는 가급적 대문자로 작성

테이블 생성하기

형식 : **CREATE TABLE [데이터베이스명.테이블명];**

```
baeys — mysql -u root — 80x24
-- UNIQUE INDEX name_UNIQUE (name ASC))
-- COMMENT = '사용자 정보'
-- ENGINE = InnoDB;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'VARVH
AR(20) NOT NULL,
age INT UNSIGNED NOT NULL,
married TINTINT NOT NULL,
comme' at line 3
mysql> CREATE TABLE nodejs.user(
    -- id INT NOT NULL AUTO_INCREMENT,
    -- name VARCHAR(20) NOT NULL,
    -- age INT UNSIGNED NOT NULL,
    -- married TINYINT NOT NULL,
    -- comment TEXT NULL,
    -- created_at DATETIME NOT NULL DEFAULT now(),
    -- PRIMARY KEY(id),
    -- UNIQUE INDEX name_UNIQUE (name ASC)
    -- )
    -- COMMENT = '사용자 정보'
    -- ENGINE = InnoDB;
Query OK, 0 rows affected (0.010 sec)
```

📌 테이블 : 데이터가 들어갈 수 있는 틀을 의미.
테이블에 맞는 데이터만 들어갈 수 있다...

📌 서로 콤마(,)로 구분함

📌 만들어두지 않은 컬럼의 정보는 저장할 수 없음

컬럼과 로우

컬럼 : 세로줄(세로 필드의 집합)

로우 : 가로줄(가로 필드의 집합)

필드 : 컬럼과 로우가 교차하는 칸 하나

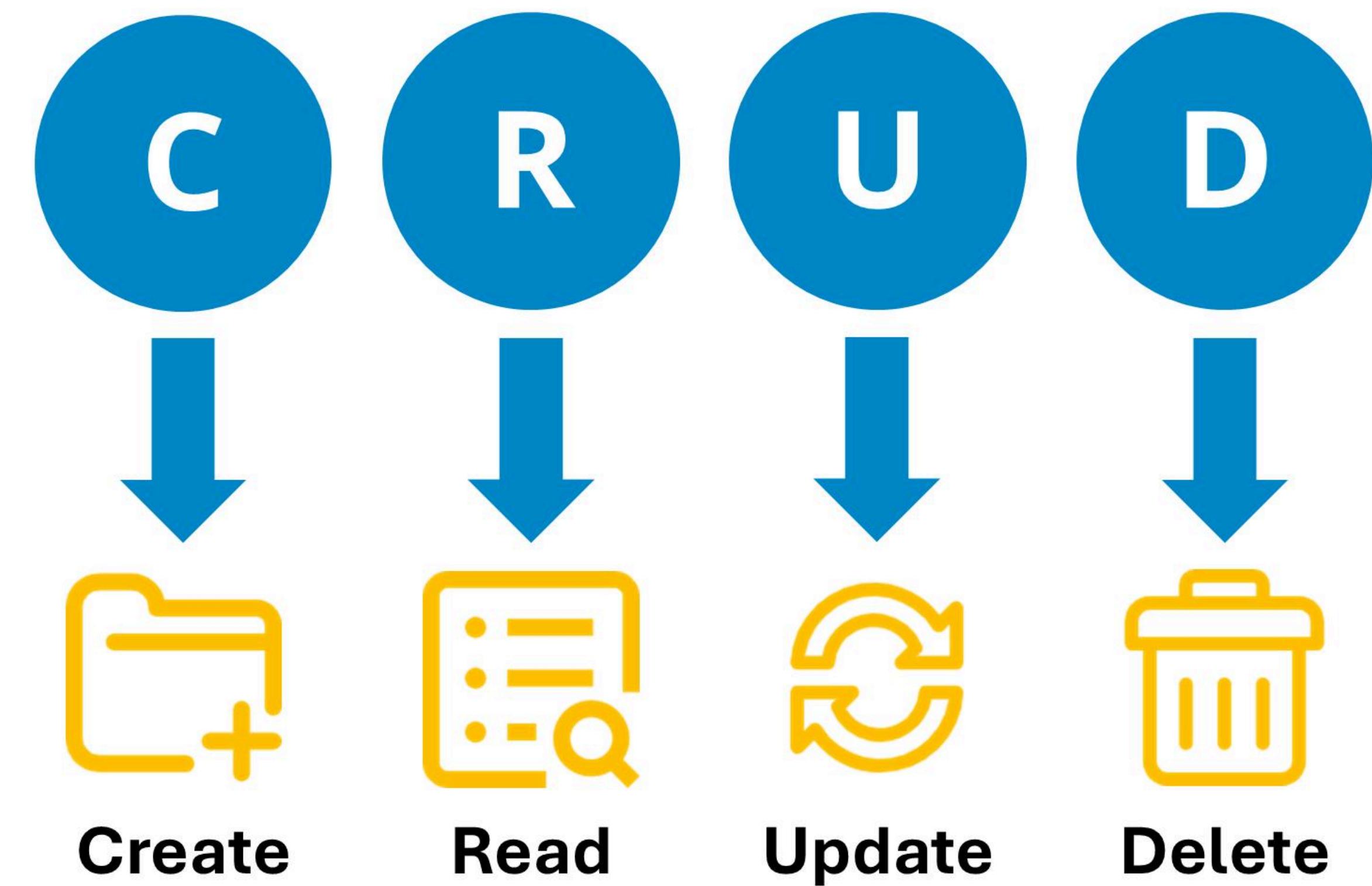
컬럼의 옵션 / 자료형

| 자료형 | 설명 |
|----------|--|
| INT | 정수. 소수까지 저장하려면 FLOAT나 DOUBLE 사용 |
| VARCHAR | 가변 길이 문자열. VARCHAR(10)이면 길이 0~10인 문자열 저장 가능 |
| CHAR | 고정 길이 문자열. CHAR(10)이면 반드시 길이 10인 문자열만 입력. 짧은 문자열 입력 시 남은 공간은 스페이스로 채워짐 |
| TEXT | 긴 글 저장용. 수백자 이내는 VARCHAR 사용, 그 이상은 TEXT 사용 |
| TINYINT | -128 ~ 127까지의 정수 저장. 1 또는 0 만 저장하면 Bool값처럼 사용 |
| DATETIME | 날짜/시간 정보 저장. DATE(날짜만), TIME(시간만) 자료형도 있음 |

| 옵션 | 설명 |
|-----------------|---|
| NULL / NOT NULL | 빈칸 허용 여부. NOT NULL이면 로우 생성 시 반드시 데이터 입력 필요 |
| AUTO_INCREMENT | 숫자를 자동으로 올림 |
| UNSIGNED | 숫자 자료형에 적용. 0부터 저장 가능(음수 무시) |
| ZEROFILL | 고정 자릿수에서 빈 자리에 0 삽입. INT(4)에 1 입력 시 0001로 저장 |
| DEFAULT now() | DB 저장 시 값이 없으면 MySQL이 기본값 자동 삽입 |
| PRIMARY KEY | 로우를 대표하는 키. 로우를 구별하는 고유 식별자 |
| UNIQUE INDEX | 값이 고유해야 함. DB가 별도로 컬럼 관리해서 조회 속도 빠름. PK는 자동으로 UNIQUE INDEX 포함 |

CRUD 생성하기

- 관련성을 Create, Read, Update, Delete의 첫글자를 모은 두문자어
- 데이터베이스에서 많이 수행하는 4가지 작업을 말함
- 중복이 없는 데이터들의 집합



CREATE

- 데이터를 생성해서 데이터베이스에 넣는 작업을 말함
- 형식: `INSERT INTO [테이블명] ([컬럼1], [컬럼2], ... [컬럼n]);`

```
mysql> INSERT INTO nodejs.user (name, age, married, comment) VALUES ('zero', 24, 0, '자기소개 1');
ERROR 1054 (42S22): Unknown column 'maage' in 'field list'
mysql> INSERT INTO nodejs.user (name, age, married, comment) VALUES ('zero', 24, 0, '자기소개 1');
Query OK, 1 row affected (0.001 sec)

mysql> INSERT INTO nodejs.user (name, age, married, comment) VALUES ('nero', 32, 1, '자기소개 2');
Query OK, 1 row affected (0.002 sec)

mysql> INSERT INTO nodejs.comments (commenter, comment) VALUES (1, '안녕하세요. zero의 댓글입니다.');
Query OK, 1 row affected (0.003 sec)
```

READ

- 데이터에 있는 데이터를 조회하는 작업

1. 모든 데이터를 조회하는 명령어 `SELECT * FROM [테이블명];`

2. 특정 칼럼만 조회하는 명령어 `SELECT [컬럼1], [컬럼2] ... FROM [테이블명];`

```
[mysql]> SELECT name, age FROM nodejs.user WHERE married = 1 AND age>30;
+-----+-----+
| name | age |
+-----+-----+
| nero | 32 |
+-----+-----+
1 row in set (0.001 sec)
```

UPDATE

- 데이터베이스에 있는 데이터를 수정하는 작업
- 형식 UPDATE [테이블명] SET [컬럼명 = 바꿀 값] WHERE [조건];

```
mysql> UPDATE nodejs.user SET comment = '바꿀 내용' WHERE id = 3;
Query OK, 0 rows affected (0.001 sec)
Rows matched: 0    Changed: 0    Warnings: 0
```

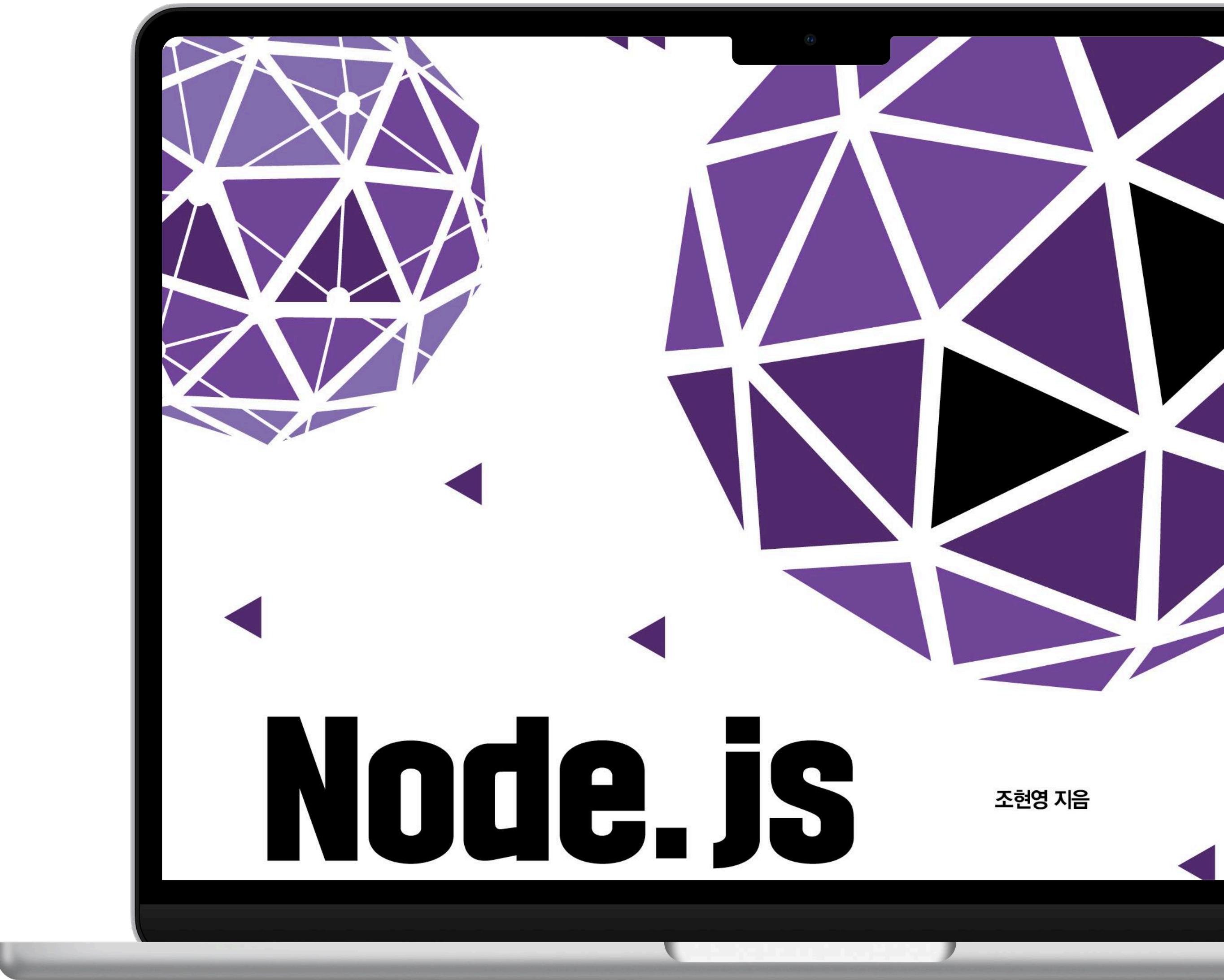
DELETE

- 데이터베이스에 있는 데이터를 삭제하는 작업
- 형식 `DELETE FROM [테이블명] WHERE [조건];`

```
[mysql]> DELETE FROM nodejs.user WHERE id = 3;  
Query OK, 0 rows affected (0.001 sec)
```

QnA

자유롭게 질문해주세요.



조현영 지음