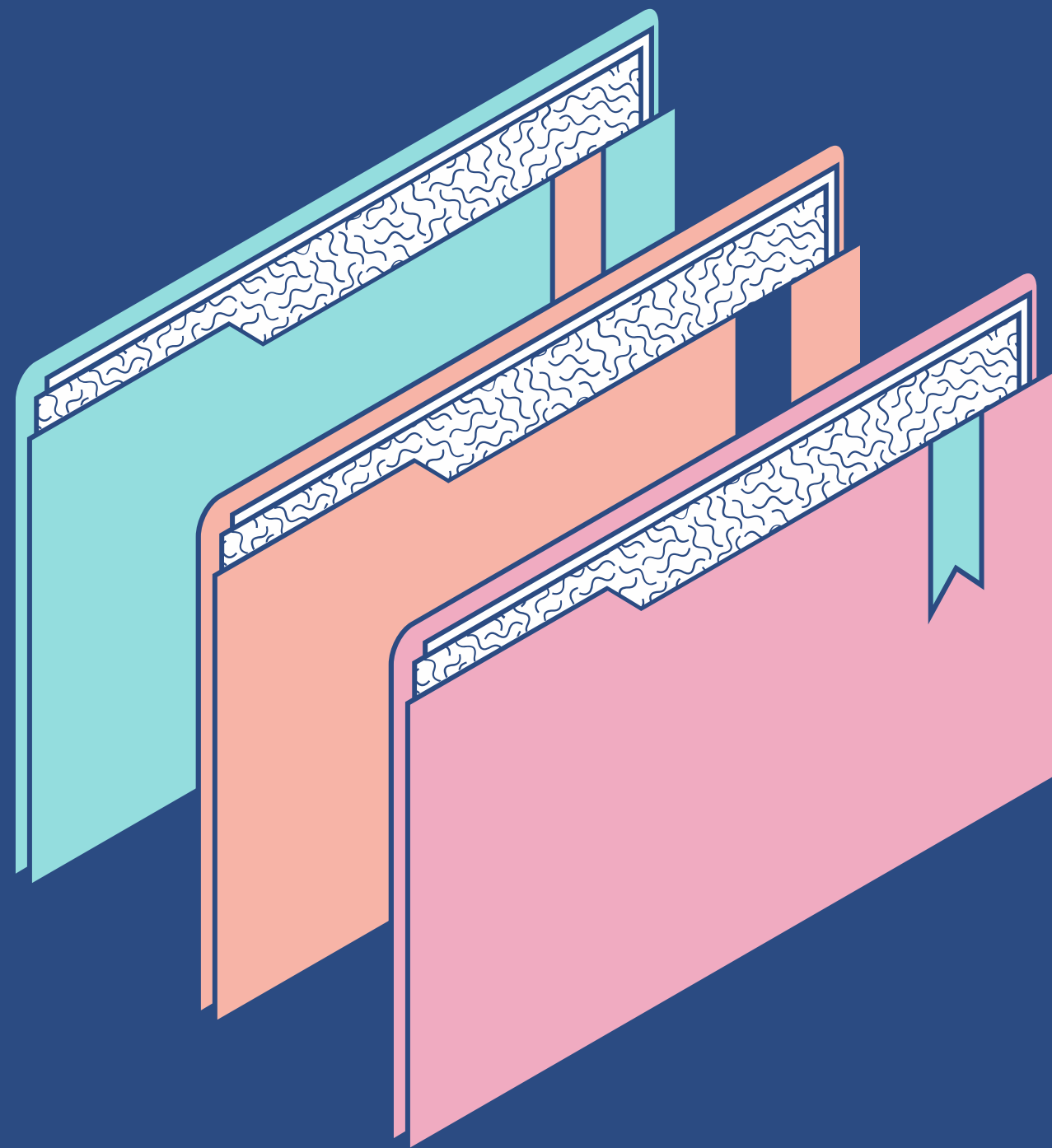




2023학년도 1학기 그룹스터디

웹프로그래밍_8주차

MaveWeb(한아림, 손효림, 이경화)

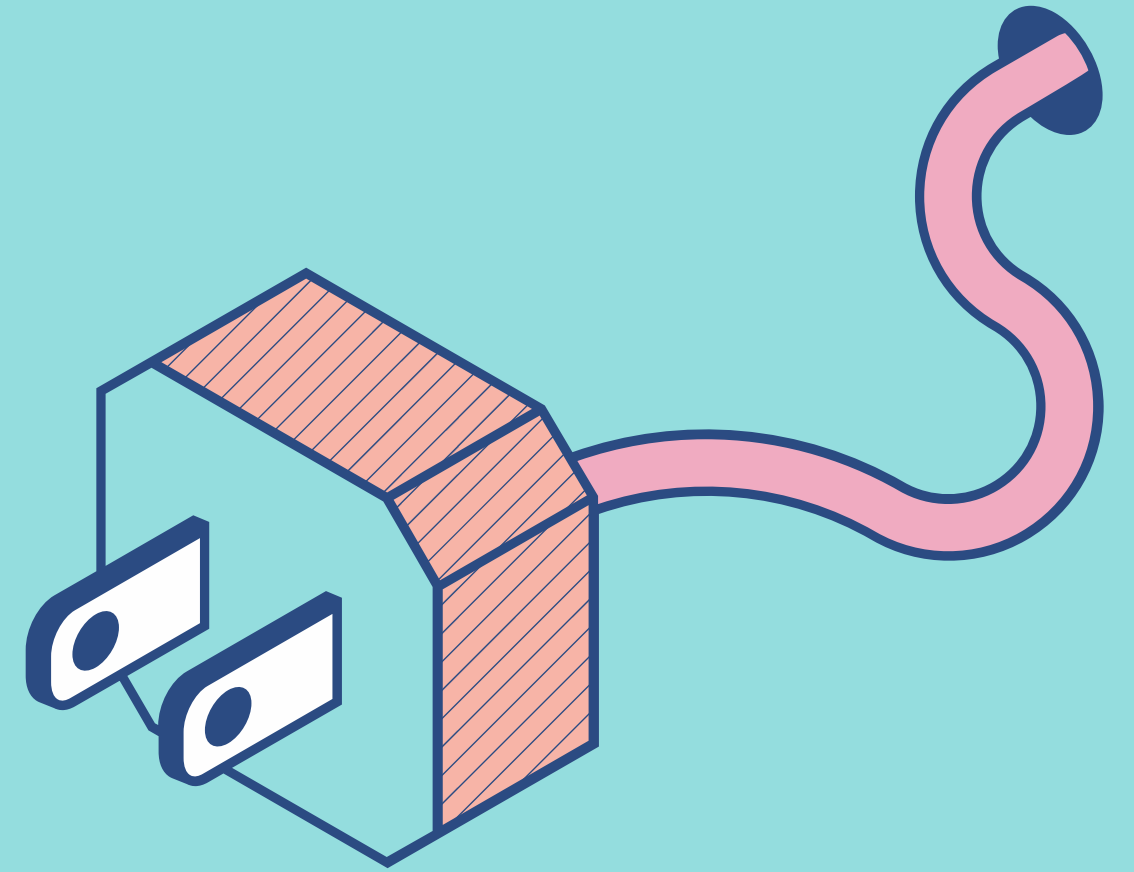


1.session 내부객체

- session 내부객체란
- session 내부객체의 메소드
- 세션 예제 살펴보기

session 내부객체란

- 클라이언트와 웹 서버 간의 상태를 지속적으로 유지하는 방법
- 클라이언트 요청에 대한 context 정보의 세션과 관련된 정보를 저장하고 관리하는 객체



session 내부객체의 메소드

<code>String getId()</code> 해당 세션의 세션id 반환	<code>int getMaxInactiveInterval()</code> 세션을 유지할 시간 반환 (기본값 30분)
<code>long getCreationTime()</code> 세션의 생성된 시간 반환	<code>boolean isNew()</code> 클라이언트 세션id를 할당하지 않은 경우 true 반환
<code>long getLastAccessedTime()</code> 클라이언트 요청이 마지막으로 시도된 시간 반환	<code>void invalidate()</code> 해당 세션 종료
<code>void setMaxInactiveInterval(time)</code> 세션을 유지할 시간을 초단위로 설정	...

세션 예제 살펴보기

세션을 이용한 아이디와
비밀번호를 입력하는 예제

form 태그 : ch05/session1.html

```
<form method=POST action="session1.jsp">
```

입력받은 아이디, 비밀번호 값을 session1.jsp로 보냄.

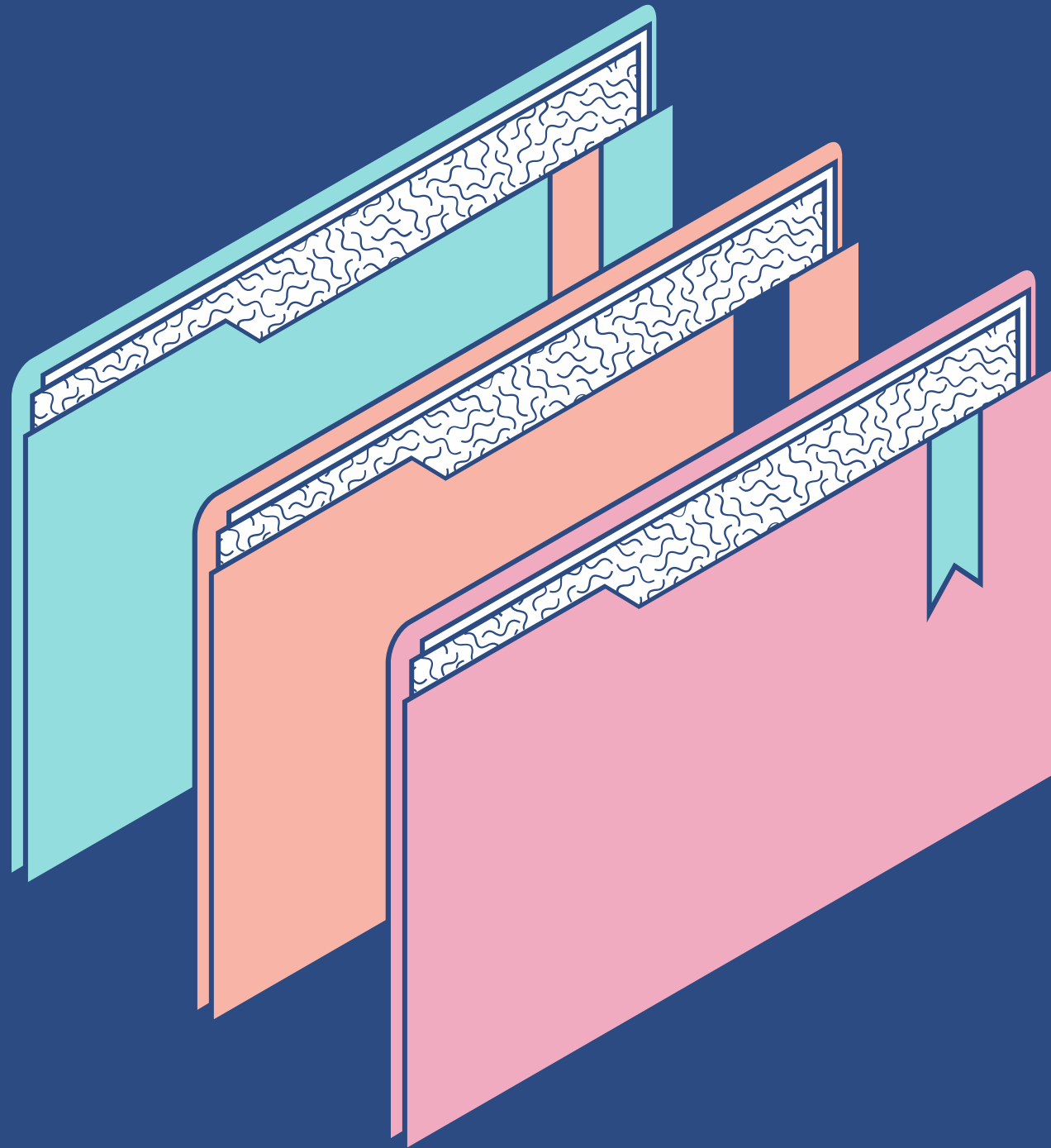
JSP 페이지 : ch05/session1.jsp

```
session.setAttribute("idKey",id);          idKey로 세션 객체 설정.  
session.setMaxInactiveInterval(60*5);      세션 유효시간 설정.  
<form method=POST action="session1_1.jsp">
```

form 태그 이용하여 session1_1.jsp로 action 보냄.

JSP 페이지 : ch05/session1_1.jsp

```
String id = (String)session.getAttribute("idKey");  
String sessionId = session.getId();  
int intervalTime = session.getMaxInactiveInterval();  
session id와 유효시간 가져옴.  
session.invalidate(); session 객체 종료.
```

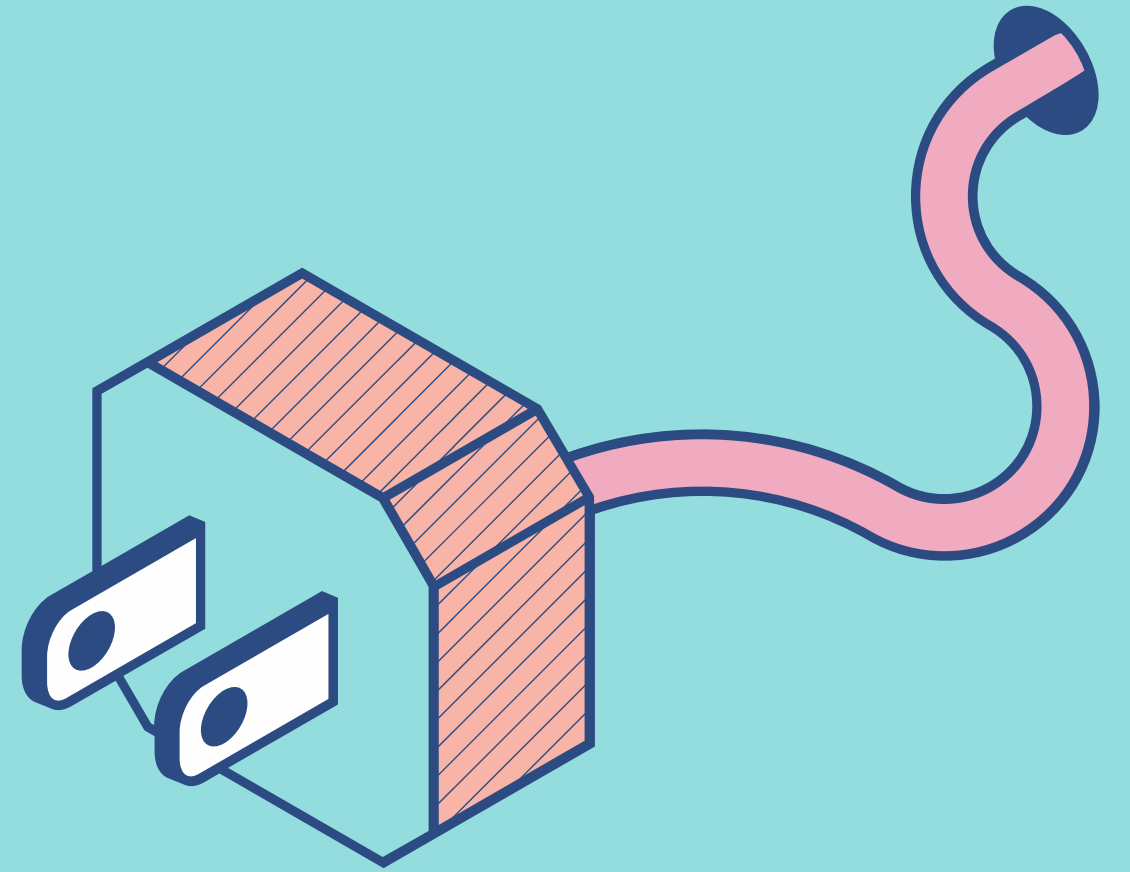


2.데이터베이스 구성요소

- 데이터베이스란
- 데이터베이스 종류
- 데이터베이스 시스템
- 관계형 데이터베이스
- 데이터베이스의 구성요소
- SQL 문법

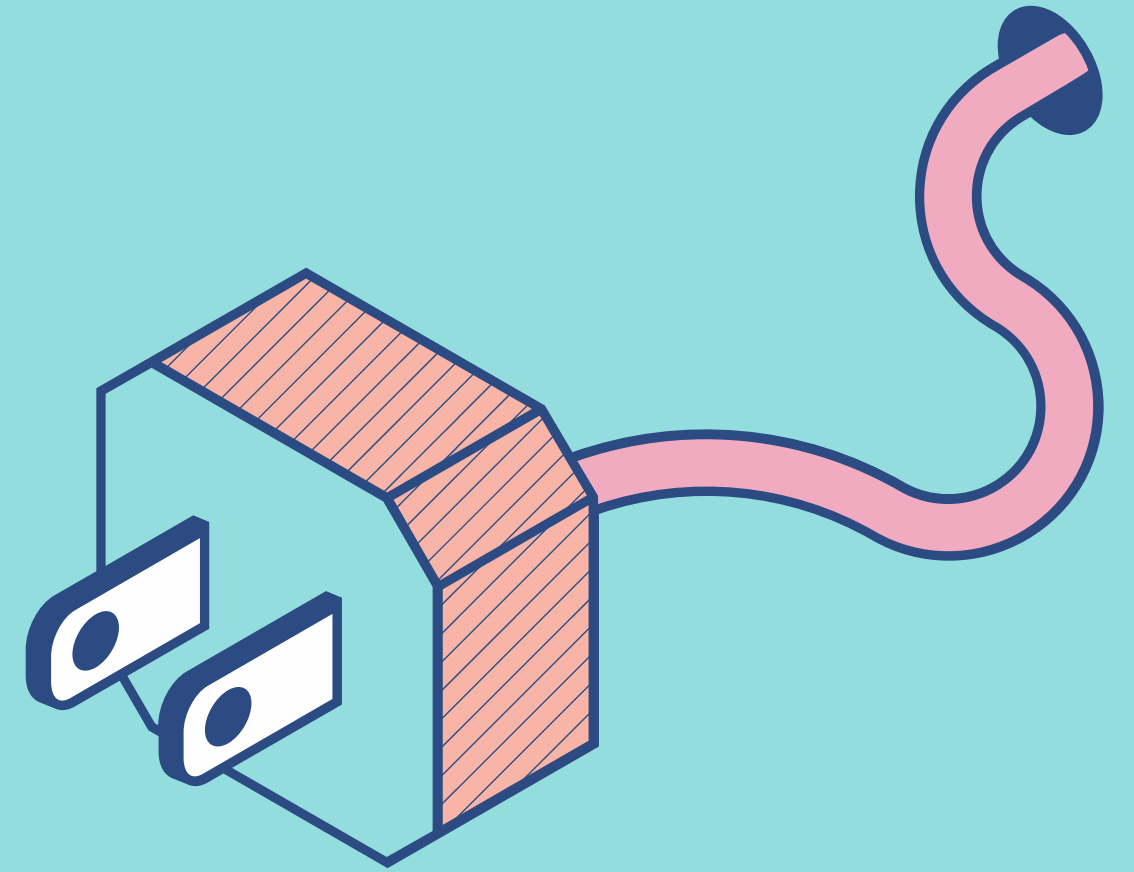
데이터베이스란

- 여러 사람이 공유할 목적으로 방대한 데이터를 체계적으로 정리하여 저장
- 데이터를 효율적으로 관리하고 검색 (예, 전화번호)

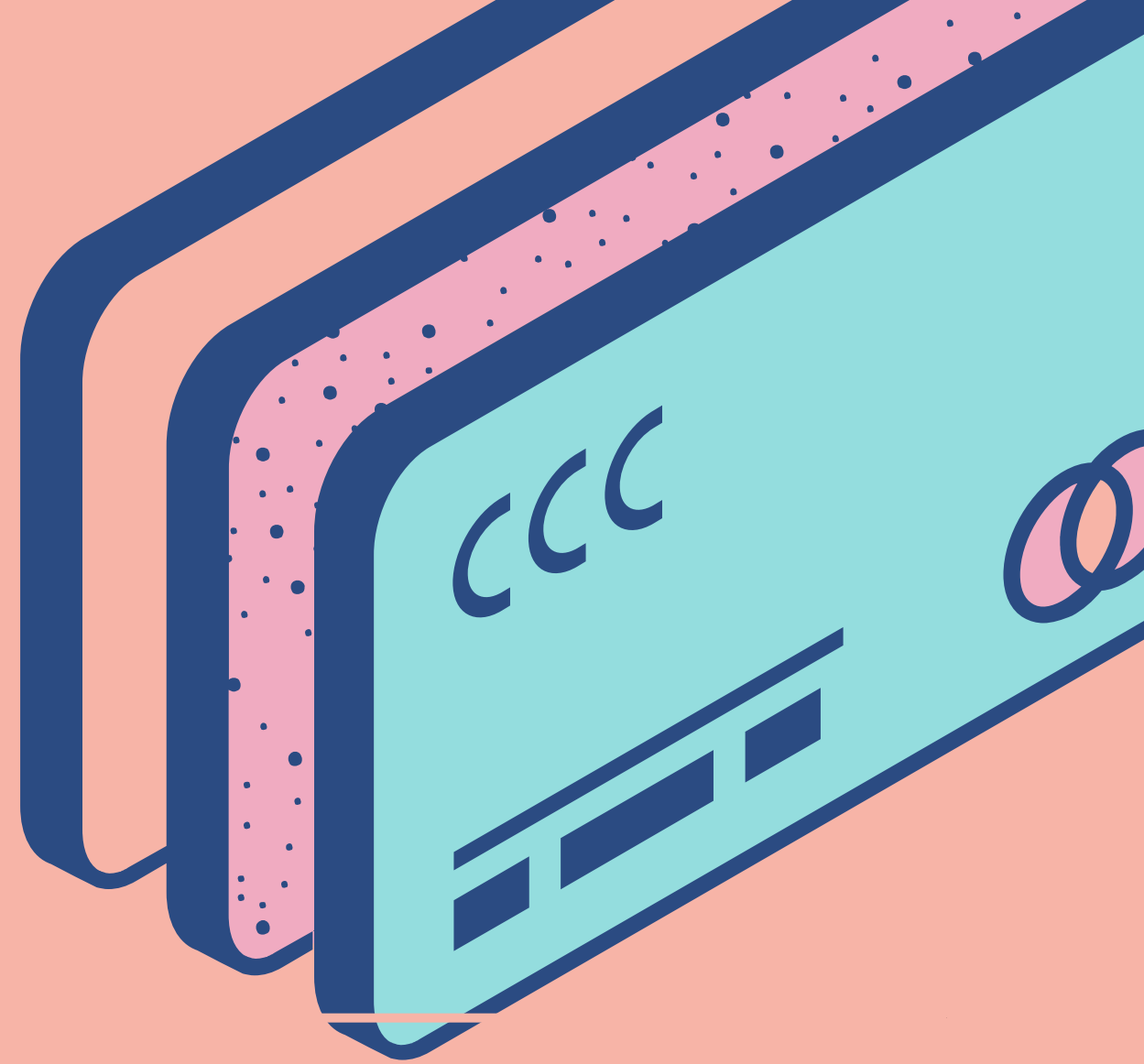


데이터베이스 관리 시스템이란

- 데이터베이스를 구성하고 운영하는 소프트웨어 시스템
- 오라클, MS SQL 서버, MySQL 등 일반적으로 알고 있는 데이터베이스 제품을 의미



데이터베이스 종류



오라클

- 사용층이 가장 넓다.
- 대형 서버까지 설치할 수 있다.
- 많은 하드웨어 자원이 필요하다.
- DBMS 관리가 복잡하다.

MySQL

- 중소 규모 서비스에 쉽게 도입할 수 있다.
- 대형 RDBMS에 사용해도 손색이 없다.
- 발전이 둔화되었다.
- MariaSQL로의 이전이 가속화되고 있다.

MS SQL 서버

- 비교적 초기 도입 비용이 저렴하다.
- 윈도우 서버 환경에서 최적화되었다.
- 윈도우 서버 운영체제에 서만 동작한다.

IBM DB2

- IBM 제품과의 호환성이 뛰어나다.
- 호환성이 제한된다.



데이터베이스 시스템

- 데이터 중복을 최소화
- 데이터 불일치 문제 해결
- 데이터 쉽게 공유
- 정보 표준화
- 데이터 보안성 제공
- 데이터 무결성 유지
- 대량의 데이터를 더 빠르게 검색
- 텍스트 이외의 데이터 관리
- 애플리케이션 개발 용이



데이터베이스 시스템 _추가적인 요구사항

- 하드웨어(서버 장비, 하드디스크)가 추가적으로 요구됨.
- 데이터 증가 시 하드웨어 추가 비용 발생
- 데이터베이스를 관리하는 DBA 필요
- 데이터 백업 및 복구 관련 전문기술 필요
- 데이터 백업 비용 발생

관계형 데이터베이스

- 데이터를 효율적으로 관리하기 위해 데이터에 관계 개념을 부여
- 대부분의 데이터베이스가 관계형 데이터베이스에 기반을 둠
- 데이터의 기본 관리 단위는 테이블(table)
- 테이블은 field와 record로 구성

데이터베이스의 구성 요소

키(Key)

데이터베이스에서 데이터를 다른 데이터와 구분할 수 있게 해주는 고유 정보

주 키(Primary Key)

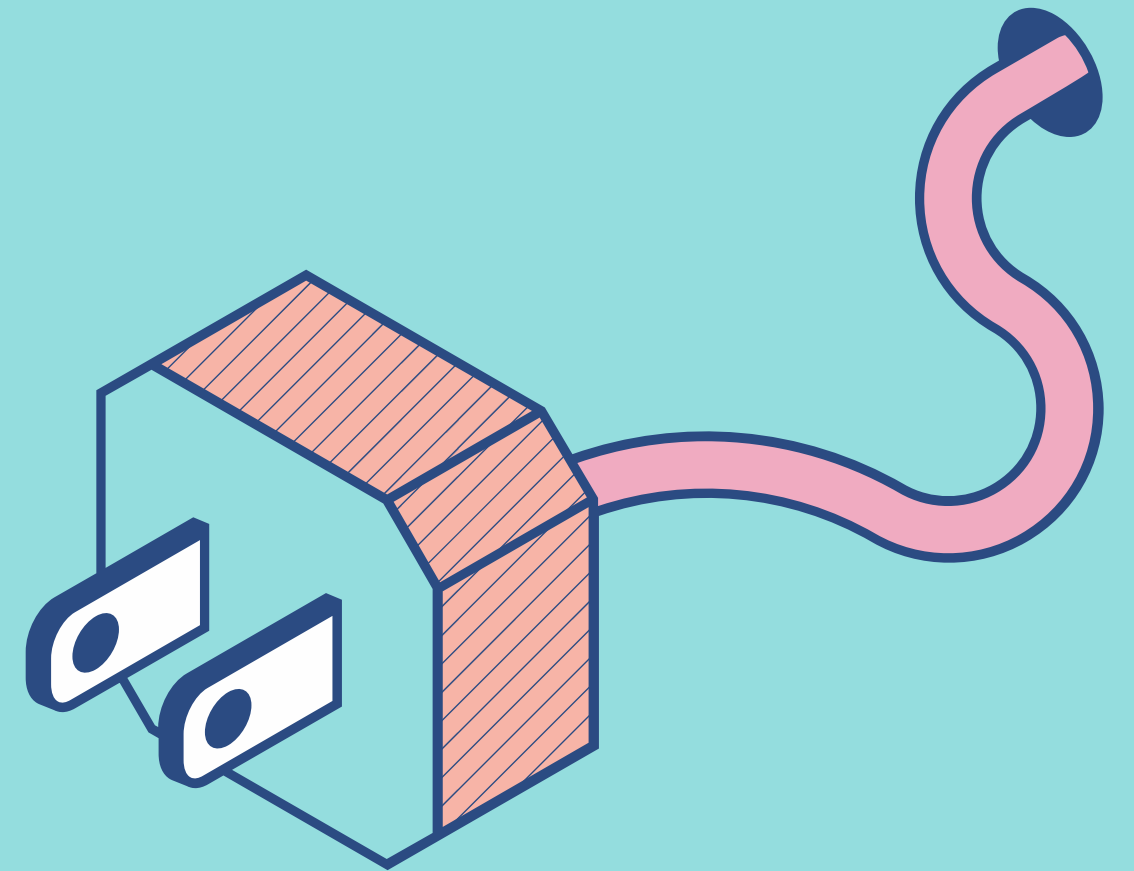
테이블 하나 당 하나만 존재하며, 각각의 레코드를 구분해 줌.
특별한 의미 없이 중복되지 않고 순차적으로 증가하는 숫자 값을 사용.

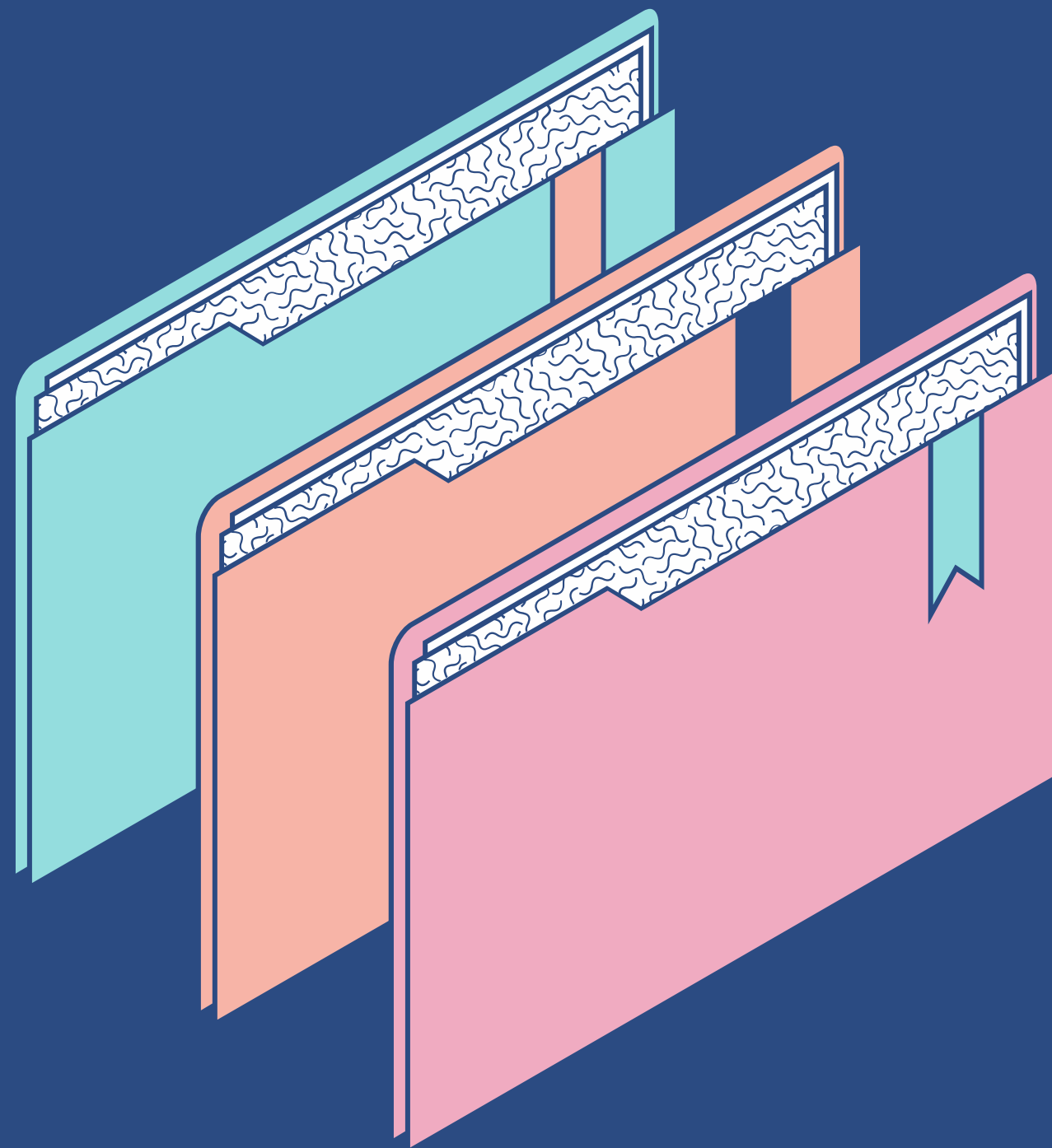
외래 키(Foreign Key)

테이블 간의 관계를 나타내고 데이터의 일관성 유지를 위해 사용

SQL 문법

- SQL은 데이터베이스의 데이터를 관리하기 위한 쿼리 언어
- 모든 데이터베이스는 SQL을 통해서만 데이터 관리 가능
- 기본적으로 ANSI 표준이며, 데이터베이스 회사별로 조금씩 다름





3.기본 SQL 명령문

- MySQL 버전 확인 명령문
- MySQL 데이터베이스 목록 확인
- 특정 데이터베이스 선택
- 테이블 목록 확인
- 특정 테이블의 구조 확인
- 데이터 조회
- MySQL 접속 종료

MySQL 버전 확인 명령문

```
MYSQL> SELECT VERSION();
```


MySQL 데이터베이스 목록 확인

```
MYSQL> SHOW DATABASES;
```

특정 데이터베이스 선택

MYSQL> USE 데이터베이스명;

테이블 목록 확인

```
MYSQL> SHOW TABLES;
```

특정 테이블의 구조 확인

MYSQL> DESC 테이블명;

데이터 조회

MYSQL> SELECT 필드1, 필드2 FROM 테이블명;

MySQL 접속 종료

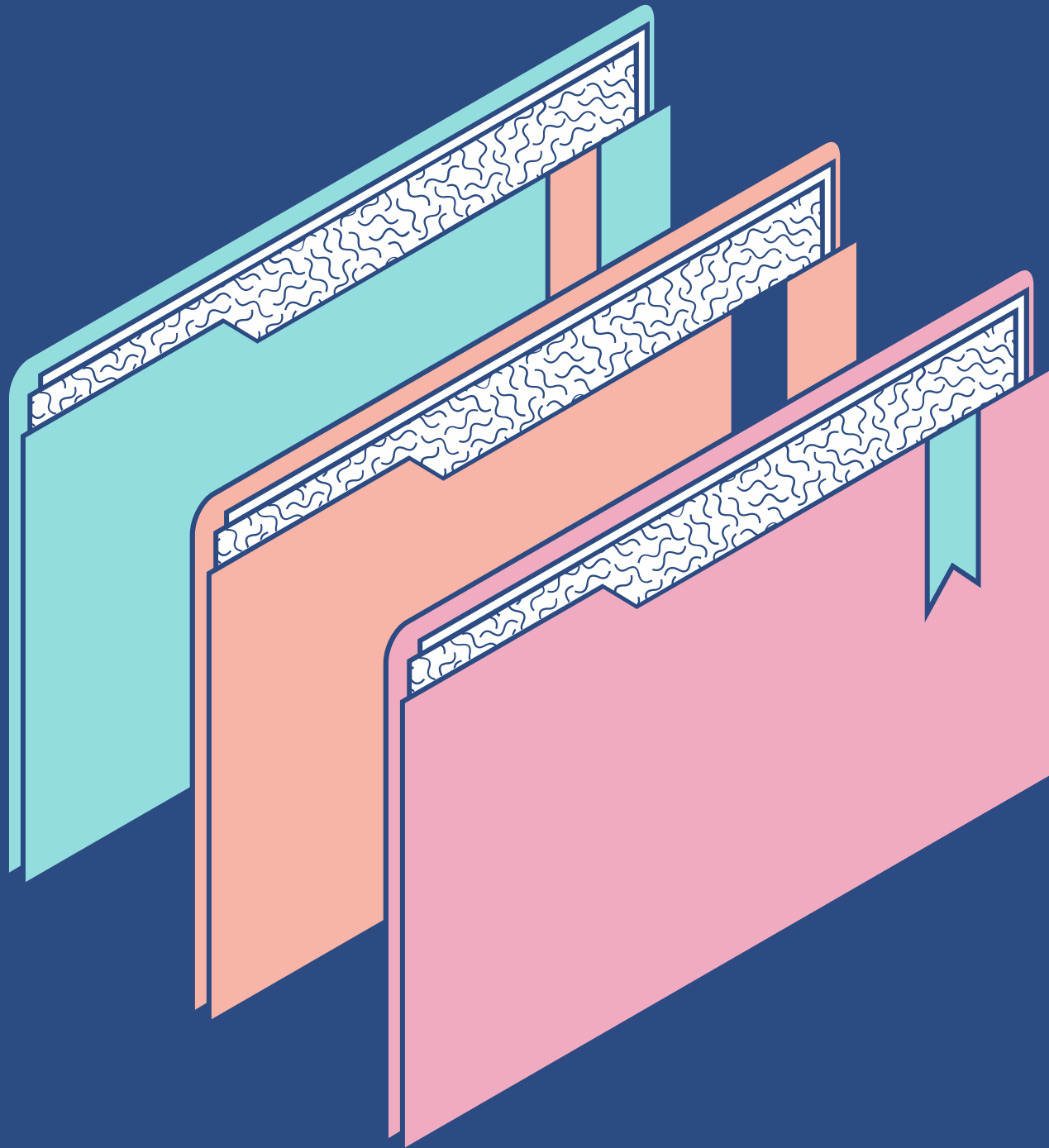
MYSQL> EXIT

또는

MYSQL> \Q

4.테이블 생성

- MySQL 데이터 형
- 테이블 생성
- 테이블 관리 명령



MySQL 데이터 형



	데이터 형	저장공간 크기	설명 및 특징
숫자형	INT(size)	4 bytes	숫자형 칼럼(실수)
	FLOAT	4 bytes	숫자형 칼럼(실수)
	DOUBLE	8 bytes	숫자형 칼럼(실수)
	REAL	8 bytes	숫자형 칼럼(실수)

MySQL 데이터 형



	데이터 형	저장공간 크기	설명 및 특징
날짜형	DATETIME	8 bytes	날짜형 칼럼
	DATE	3 bytes	날짜형 칼럼
	TIMESTAMP	4 bytes	날짜형 칼럼

MySQL 데이터 형



	데이터 형	저장공간 크기	설명 및 특징
문자형	CHAR	1~255까지 저장	문자형 칼럼
	VARCHAR	1~255까지 저장	문자형 칼럼
	BOLB	최대길이 65536	문자형 칼럼

테이블 생성

MYSQL> DROP TABLE IF EXISTS 테이블명;

MYSQL> CREATE TABLE 테이블명 (
 필드1 데이터형 NOT NULL 또는 NULL,
 필드2 데이터형 NOT NULL 또는 NULL,
 ...
);

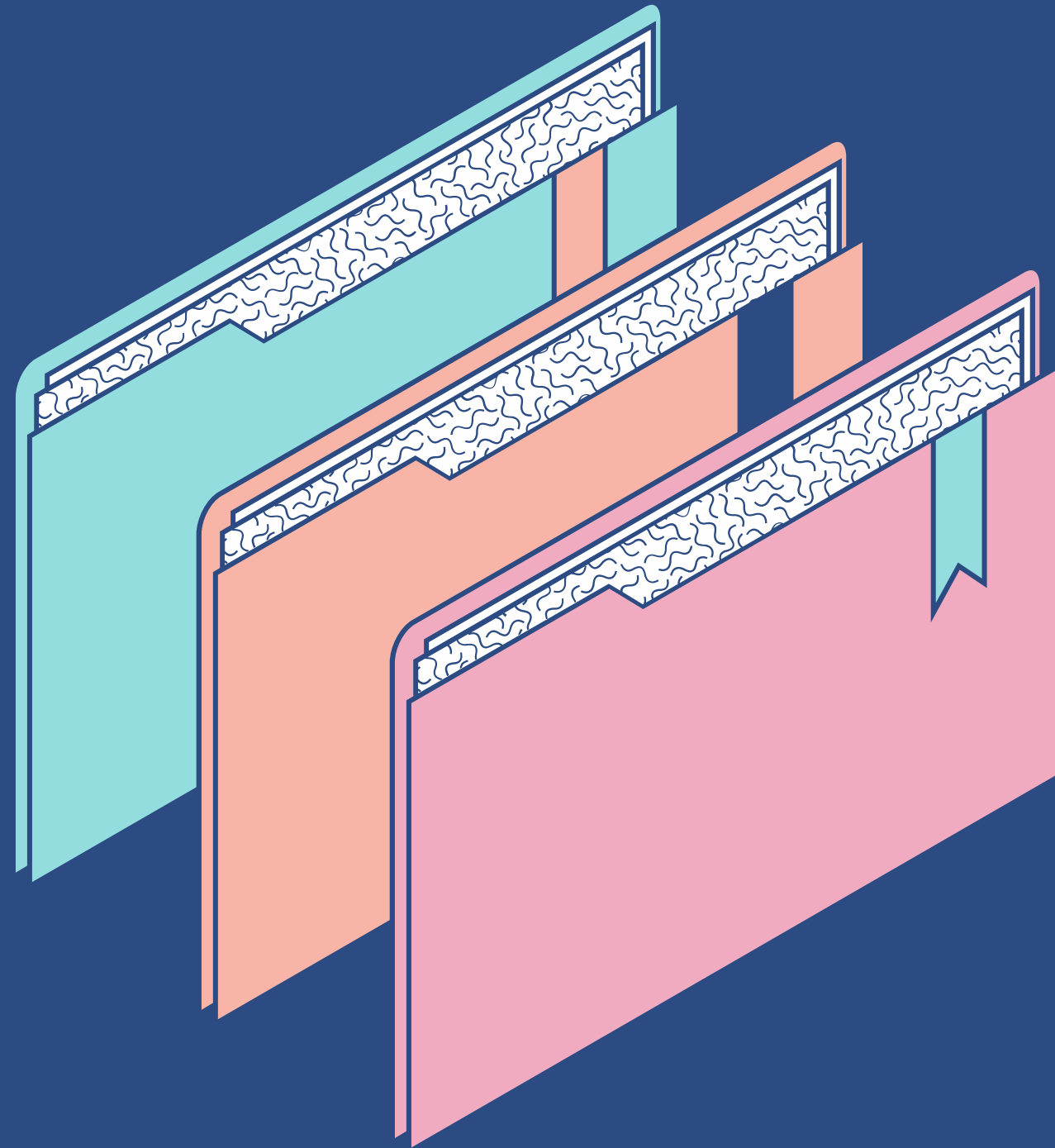
테이블 관리 명령

테이블 구조 변경 : 특정 키를 주 키로 설정

```
MYSQL> ALTER TABLE 테이블명  
        ADD PRIMARY KEY(필드명);
```

테이블 삭제

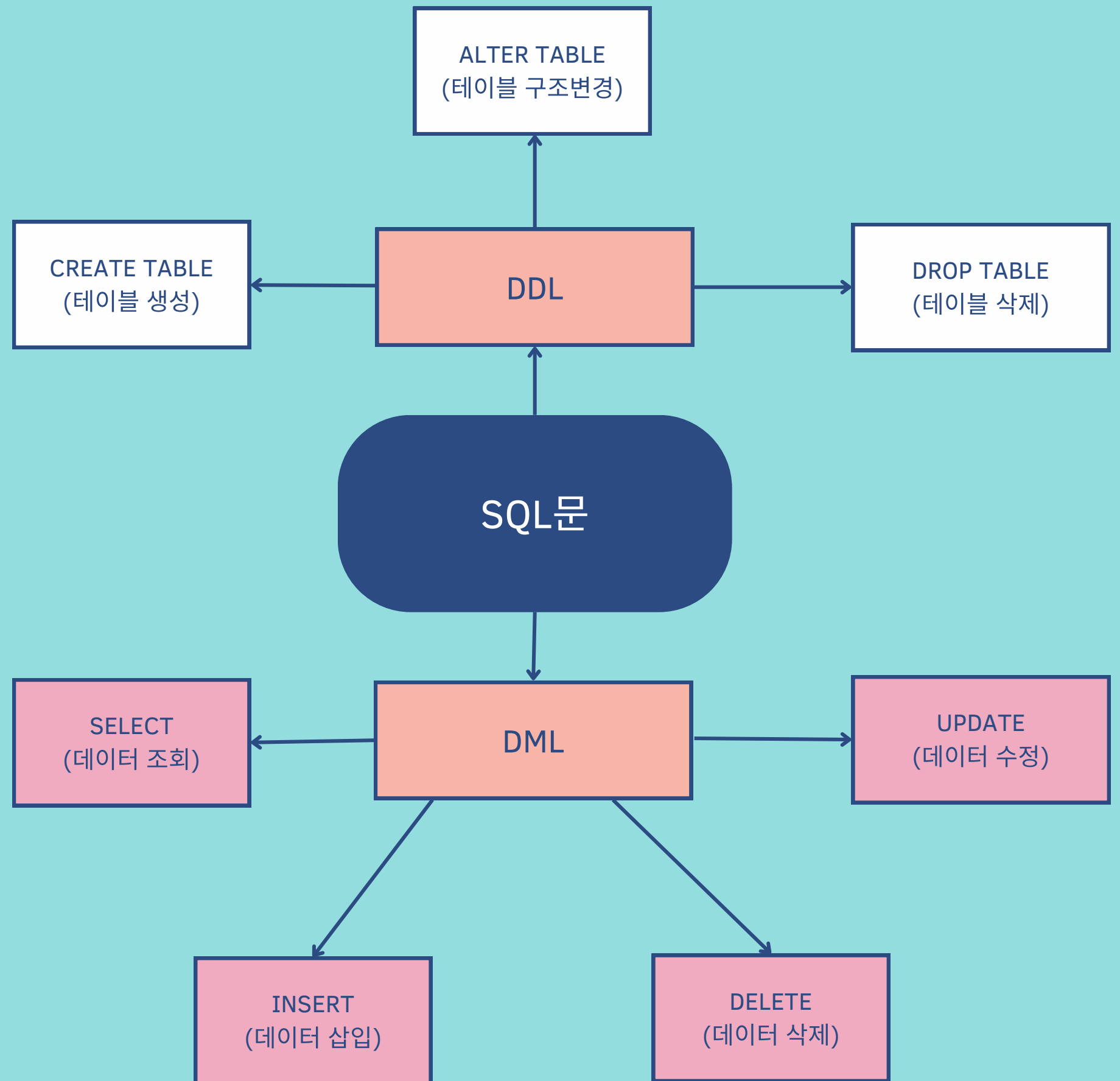
```
MYSQL> DROP TABLE 테이블명;
```



INSERT, UPDATE, DELETE

- DML
- 데이터 입력(INSERT)
- 데이터 변경(UPDATE)
- 데이터 삭제(DELETE)

SQL문 - DDL, DML



데이터 입력

```
MYSQL> INSERT INTO 테이블명(필드1, 필드2, ...)  
VALUES(필드값1, 필드값2, ...);
```

데이터 변경

```
MYSQL> UPDATE 테이블명  
      SET 필드명 = 변경할 조건  
      WHERE 필드명 = 기존_속성값;
```


데이터 삭제

테이블에 등록된 특정 데이터 삭제

```
MYSQL> DELETE FROM 테이블명  
        WHERE 필드명 = 조건;
```

데이터 삭제

전체 레코드 삭제

`MYSQL> DELETE FROM 테이블명;`

- 삭제된 데이터는 다시 복구할 수 없으므로 테이블에 존재하는 데이터를 삭제할 때는 신중하게 수행해야 함.

감사합니다

