

# 7강 자료

## SQL 명령어의 종류

SQL (Structured Query Language)

관계형 데이터베이스 관리 시스템(RDBMS)의 데이터를 관리하기 위해 설계된 특수 목적의 프로그래밍 언어

SQL 쿼리문은 데이터베이스에 저장된 값을 불러내기 위한 명령문

데이터베이스 관련 명령어와 테이블 관련 명령어로 구분되어 수행

일부 명령어를 제외하고 나머지 명령어에 대한 알파벳 대/소문자를 특별히 구분하지 않음

SQL 명령어	의미
CREATE	새로운 데이터베이스 생성
DROP	생성되어 있는 데이터베이스 삭제
USE	데이터베이스 목록에서 특정 데이터베이스를 선택

## 데이터베이스 생성

CREATE 명령어

데이터베이스를 생성할 때 사용

```
CREATE DATABASE 데이터베이스명;
```

## 데이터베이스 생성

1) 데이터베이스 생성 : 콘솔창에 다음과 같이 입력 후 엔터

```
CREATE DATABASE chap14;
```

2) 데이터 목록 확인

```
SHOW DATABASES;
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| chap14   |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0.00 sec)
```

## Database 삭제

DROP 명령어

생성되어 있는 데이터베이스를 삭제할 때 사용

DROP DATABASE 데이터베이스명;

## 데이터베이스 선택

- 1) 데이터베이스 생성 : CREATE DATABASE chap14;
- 2) 데이터 베이스 목록 확인하기 : SHOW DATABASES;
- 3) 데이터베이스 선택 : USE chap14;

```
mysql> CREATE DATABASE chap14;
Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| chap14   |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0.00 sec)

mysql> USE chap14;
Database changed
```

## 데이터베이스 선택

USE 명령어

생성되어 있는 데이터베이스를 선택할 때 사용

USE 데이터베이스명;

## Database 삭제

테이블 관련 SQL 명령어

SQL 쿼리문에서 사용하는 테이블 관련 명령어의 종류

SQL 명령어	의미
CREATE	새로운 테이블 생성
DESC (또는 DESCRIBE)	테이블의 구조 확인
ALTER	테이블의 구조 변경
RENAME	테이블의 이름 변경
DROP	테이블 삭제



## 데이터 타입

### 데이터 타입

실제 데이터를 저장하기 위해 행과 열로 구분된 테이블을 생성할 때는 열에 해당하는 필드의 데이터 타입을 설정  
필드의 데이터 타입을 문자열 또는 정수형 등으로 지정하여 데이터의 타입을 명확하게 지정

구분	데이터 타입	최대 길이	저장 크기
문자열	char	255	입력한 숫자 byte
	varchar	255	(입력한숫자+1) byte
숫자형	int	-2147483648~2147483647	정수형 4byte
	float	-3.402823466E+38 ~ -1.175494351E-38	실수형 8byte
	double	-1.7976931348623157E+308~-2.2250738585072014E-308	실수형 8byte

#### ※ char와 varchar의 차이

- char : 고정폭 문자열로 최대 255자까지 문자 저장
- varchar : 가변폭 문자열로 최대 255자까지 문자 저장
- 메모리 효율을 높이기 위해 가급적이면 문자열 데이터 타입은 varchar 타입으로 선언하는 경향이 더 많음

## 테이블 생성 예제

### CREATE 명령어

SQL 쿼리문에서 테이블을 생성할 때 사용

```
mysql> CREATE TABLE 테이블명(  
    -> 필드명1 타입,  
    -> 필드명2 타입,  
    -> ...  
    -> PRIMARY KEY(필드명), NOT NULL  
    -> );
```

#### ✓ Check Point\_ 기본 키 설정 : PRIMARY KEY(code), NOT NULL

PRIMARY KEY로 지정된 code 필드는 테이블에서 중복된 값과 빈 공백으로 생략할 수 없는 필수 입력 항목을 지정할 때 선언합니다. 주민등록번호나 학번 등 중복성 없이 독립된 데이터의 값을 테이블에 저장할 때는 특정 필드를 지정하여 기본 키로 설정합니다.

## 테이블 생성 예제

### 테이블 생성 예제

chap14에 새로운 테이블 member 을 생성하기 위한 예제 수행

member 테이블 구조

필드명	데이터 타입	의미	비고
memberid	varchar(10)N	10byte 크기의 문자열형	PRIMARY KEY
password	varchar(10)	10byte 크기의 문자열형	
name	varchar(20)	20byte 크기의 문자열형	
email	varchar(80)	80byte 크기의 문자열형	

## 테이블 생성

### 1) 테이블 생성

```
CREATE TABLE member (  
  memberId varchar(10) NOT NULL PRIMARY KEY,  
  password varchar(10) NOT NULL,  
  name varchar(20),  
  email varchar(80)  
);
```

```
mysql> use chap14  
Database changed  
mysql> CREATE TABLE member (  
  -> memberId varchar(10) NOT NULL PRIMARY KEY,  
  -> password varchar(10) NOT NULL,  
  -> name varchar(20),  
  -> email varchar(80)  
  -> );  
Query OK, 0 rows affected (0.05 sec)
```

### 2) 테이블 목록 확인 : SHOW TABLES;

```
mysql> show tables;  
+-----+  
| Tables_in_chap14 |  
+-----+  
| member           |  
+-----+  
1 row in set (0.02 sec)
```

## 테이블 구조 확인

DESC 또는 DESCRIBE 명령어

SQL 쿼리문에서 테이블의 구조를 확인할 때 사용

```
mysql> DESC 테이블명;  
mysql> DESCRIBE 테이블명;
```

## 테이블 구조 확인

- 1) DESC 명령문을 이용한 테이블 구조 확인 : DESC member ;
- 2) DESCRIBE 명령문을 이용한 테이블 구조 확인 : DESCRIBE member;

```
mysql> DESC member ;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| memberId | varchar(10) | NO   | PRI | NULL    |       |
| password | varchar(10) | NO   |     | NULL    |       |
| name     | varchar(20) | YES  |     | NULL    |       |
| email    | varchar(80) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> DESCRIBE member;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| memberId | varchar(10) | NO   | PRI | NULL    |       |
| password | varchar(10) | NO   |     | NULL    |       |
| name     | varchar(20) | YES  |     | NULL    |       |
| email    | varchar(80) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## 테이블 구조 변경

ALTER 명령어

SQL 쿼리문에서 테이블의 구조를 변경할 때 사용

새로운 필드 추가하기

ALTER 명령어와 옵션 ADD를 사용하여 테이블에 새로운 필드 추가

```
mysql> ALTER TABLE 테이블명 ADD 필드명 데이터 _ 타입 [AFTER 필드명];
```

member\_의 name 필드명 다음에 age 정수형 필드명을 추가하기 위해 다음 예제를 수행

## 테이블 구조 변경

- 1) 테이블에 새로운 필드 추가 : name 필드 다음에 age 필드명을 추가하기

```
ALTER TABLE member ADD age int AFTER name;
```

- 2) 변경된 테이블 구조 확인 ; DESC member;

```
mysql> ALTER TABLE member ADD age int AFTER name;
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> DESC member;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| memberId  | varchar(10)   | NO   | PRI | NULL    |       |
| password   | varchar(10)   | NO   |     | NULL    |       |
| name       | varchar(20)   | YES  |     | NULL    |       |
| age        | int           | YES  |     | NULL    |       |
| email      | varchar(80)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```



## 테이블 구조 변경

### 1) 기존 필드 삭제하기

DROP 명령어를 사용하여 기존 테이블에 존재하는 특정 필드 삭제

```
mysql> ALTER TABLE 테이블명 DROP 필드명;
```

member 에 존재하는 age 필드명을 삭제하기 위해 다음 예제를 수행

## 테이블 구조 변경

### 1) 테이블의 기존 필드 삭제

```
ALTER TABLE member DROP age;
```

### 2) 변경된 테이블 구조 확인 ; DESC member;

```
mysql> ALTER TABLE member DROP age;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> DESC member;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| memberId  | varchar(10)   | NO   | PRI | NULL    |       |
| password   | varchar(10)   | NO   |     | NULL    |       |
| name       | varchar(20)   | YES  |     | NULL    |       |
| email      | varchar(80)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## 테이블 이름 변경

RENAME 명령어

SQL 쿼리문에서 테이블의 이름을 변경할 때 사용

```
mysql> ALTER TABLE 기존 _ 테이블명 RENAME 변경 _ 테이블명;
```

테이블의 이름 'member'을 'student'로 변경하기 위해 다음 예제를 수행

## 테이블 이름 변경

1) 기존 테이블의 이름 변경하기

```
ALTER TABLE member RENAME student;
```

2) 테이블명 확인하기 : SHOW TABLES;

```
mysql> ALTER TABLE member RENAME student;
Query OK, 0 rows affected (0.03 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_chap14 |
+-----+
| student          |
+-----+
1 row in set (0.00 sec)
```

## 테이블 삭제

DROP 명령어

SQL 쿼리문에서 기존 테이블을 삭제할 때 사용

```
mysql> DROP TABLE 테이블명;
```

- DROP 명령어는 데이터베이스 또는 테이블을 삭제할 때 사용하는 명령어로,
- 데이터베이스를 삭제하면 그 안에 들어있는 모든 테이블도 같이 삭제되므로 주의 필요
- 데이터베이스 또는 테이블을 삭제하게 되면 복구 할 수 없음

## 테이블 삭제

1) 기존 테이블의 이름 변경하기

```
DROP TABLE student;
```

2) 테이블명 확인하기 : SHOW TABLES;

```
mysql> DROP TABLE student;  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> SHOW TABLES;  
Empty set (0.00 sec)
```

# 7강

## SQL 명령어의 종류

테이블의 구성요소

테이블은 행(레코드)과 열(필드)로 구성되어 있는 요소

The diagram illustrates the components of a table. A bracket labeled '필드명' (Field Name) points to the header row. A bracket labeled '레코드' (Record) points to the data rows. A bracket labeled '필드' (Field) points to the entire table structure.

memberId	passwd	name	address
202412345	202412345	홍길동	컴퓨터정보공학과
202435587	202435587	김길동	컴퓨터소프트웨어공학과
202443218	202443218	이효리	인공지능소프트웨어학과

## SQL 명령어의 종류

테이블 조작 명령어

SQL 쿼리문에서 사용하는 테이블 조작 명령어의 종류

SQL 명령어	의미
INSERT	데이터를 레코드 단위로 삽입
SELECT	테이블에 존재하는 레코드 검색
UPDATE	테이블에 존재하는 레코드 수정
DELETE	테이블에 존재하는 레코드 삭제



## 데이터 등록

### INSERT 명령어

SQL 쿼리문에서 테이블에 데이터를 등록할 때 사용

```
mysql> INSERT INTO 테이블명(필드명1, 필드명2, ...)  
-> VALUES(필드값1, 필드값2, ...);
```

member 에 다음 표에서 제시하는 데이터를 등록하기 위해 다음 예제를 수행

memberId	password	name	emali
202412345	202412345	홍길동	<a href="mailto:hgd@test.com">hgd@test.com</a>
202435587	202435587	김길동	<a href="mailto:kgd@test.com">kgd@test.com</a>
202443218	202443218	이효리	<a href="mailto:lhr@test.com">lhr@test.com</a>

## 데이터 등록

### 1) 레코드 등록하기

```
INSERT INTO member(memberId, password, name, email) VALUES ('202412345', '202412345', '홍길동', 'hgd@test.com');
```

```
INSERT INTO member(memberId, password, name, email) VALUES('203535587', '203535587', '김길동', 'kdg@test.com');
```

```
INSERT INTO member(memberId, password, name, email) VALUES('203943218', '203943218', '이효리', 'lhr@test.com');
```

```
mysql> insert into member(memberId, password, name, email) values ('202412345', '202412345', '홍길동', 'hgd@test.com');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into member(memberId, password, name, email) values('203535587', '203535587', '김길동', 'kdg@test.com');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO member(memberId, password, name, email) VALUES('203943218', '203943218', '이효리', 'lhr@test.com');  
Query OK, 1 row affected (0.00 sec)
```

# 7강

---

## 데이터 조회

SELECT 명령어

SQL 쿼리문에서 테이블에 등록된 데이터를 조회할 때 사용

```
mysql> SELECT *FROM 테이블명;
```

member 에 등록된 데이터를 조회하기 위해 다음 예제를 수행

## 데이터 조회

### 1) 테이블에 등록된 전체 데이터 조회

```
SELECT * FROM member;
```

### 2) name 필드의 데이터만 조회

```
SELECT name FROM member;
```

### 3) 테이블에서 '김'씨 성을 가진 데이터만 조회하기

```
SELECT memberId, name, email FROM member  
WHERE name LIKE '김%';
```

```
mysql> SELECT name FROM member;  
+-----+  
| name |  
+-----+  
| 홍길동 |  
| 김길동 |  
| 이효리 |  
+-----+  
3 rows in set (0.00 sec)  
  
mysql> SELECT name FROM member;  
+-----+  
| name |  
+-----+  
| 홍길동 |  
| 김길동 |  
| 이효리 |  
+-----+  
3 rows in set (0.00 sec)  
  
mysql> SELECT memberId, name, email FROM member WHERE name LIKE '김%';  
+-----+-----+-----+  
| memberId | name | email |  
+-----+-----+-----+  
| 203535587 | 김길동 | kdg@test.com |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

## 데이터 수정

### UPDATE 명령어

SQL 쿼리문에서 테이블에 등록된 데이터를 수정할 때 사용

```
mysql> UPDATE 테이블명  
-> SET 필드명 = 변경할 조건  
-> WHERE 필드명 = 기존 _ 속성값;
```

member 테이블의 address 필드명을 dept로 변경하고 등록된 데이터 중 특정 데이터를 수정하기 위해 다음 예제를 수행

## 데이터 수정

### 1) 테이블의 필드명 수정

```
ALTER TABLE member CHANGE email code varchar(20);
```

### 2) 변경된 필드명 확인

```
SELECT * FROM member;
```

```
mysql> ALTER TABLE member CHANGE email code varchar(20);
Query OK, 3 rows affected (0.06 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM member;
+-----+-----+-----+-----+
| memberId | password | name | code |
+-----+-----+-----+-----+
| 202412345 | 202412345 | 홍길동 | hgd@test.com |
| 203535587 | 203535587 | 김길동 | kdg@test.com |
| 203943218 | 203943218 | 이효리 | lhr@test.com |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

## 데이터 수정

### 3) 테이블에 등록된 특정 데이터 수정하기

UPDATE member

SET code = "code@test.com"

WHERE memberId = '202412345';

### 4) 변경된 필드명 확인

SELECT \* FROM member;

```
mysql> UPDATE member
->     SET code = "code@test.com"
->     WHERE memberId = '202412345';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from member;
+-----+-----+-----+-----+
| memberId | password | name | code |
+-----+-----+-----+-----+
| 202412345 | 202412345 | 홍길동 | code@test.com |
| 203535587 | 203535587 | 김길동 | kdg@test.com |
| 203943218 | 203943218 | 이효리 | lhr@test.com |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

## 데이터 삭제

DELETE 명령어

SQL 쿼리문에서 테이블에 등록된 특정 데이터를 삭제할 때 사용

```
mysql> DELETE FROM 테이블명  
      -> WHERE 필드명 = 조건;
```

### 5.1 특정 레코드 삭제

member 테이블에 존재하는 특정 레코드를 삭제하기 위해 다음 예제를 수행



## 데이터 삭제

- 1) 테이블의 특정 레코드 삭제

DELETE FROM member

WHERE memberId='202412345';

- 2) 변경된 필드명 확인

SELECT \* FROM member;

```
mysql> DELETE FROM member
-> WHERE memberId='202412345';
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from member;
```

memberId	password	name	code
203535587	203535587	김길동	kdg@test.com
203943218	203943218	이효리	lhr@test.com

2 rows in set (0.00 sec)

## 데이터 삭제

### 전체 레코드 삭제

DELETE 명령어로 기존 테이블에 존재하는 모든 레코드를 한꺼번에 삭제

```
mysql> DELETE FROM 테이블명;
```

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

#### 1) 테이블의 전체 레코드 삭제

```
DELETE FROM member;
```

#### 2) 변경된 필드명 확인

```
SELECT * FROM member;
```

```
mysql> DELETE FROM member;
Query OK, 2 rows affected (0.00 sec)

mysql> SELECT * FROM member;
Empty set (0.00 sec)
```

End of Document

Thank you