제03장

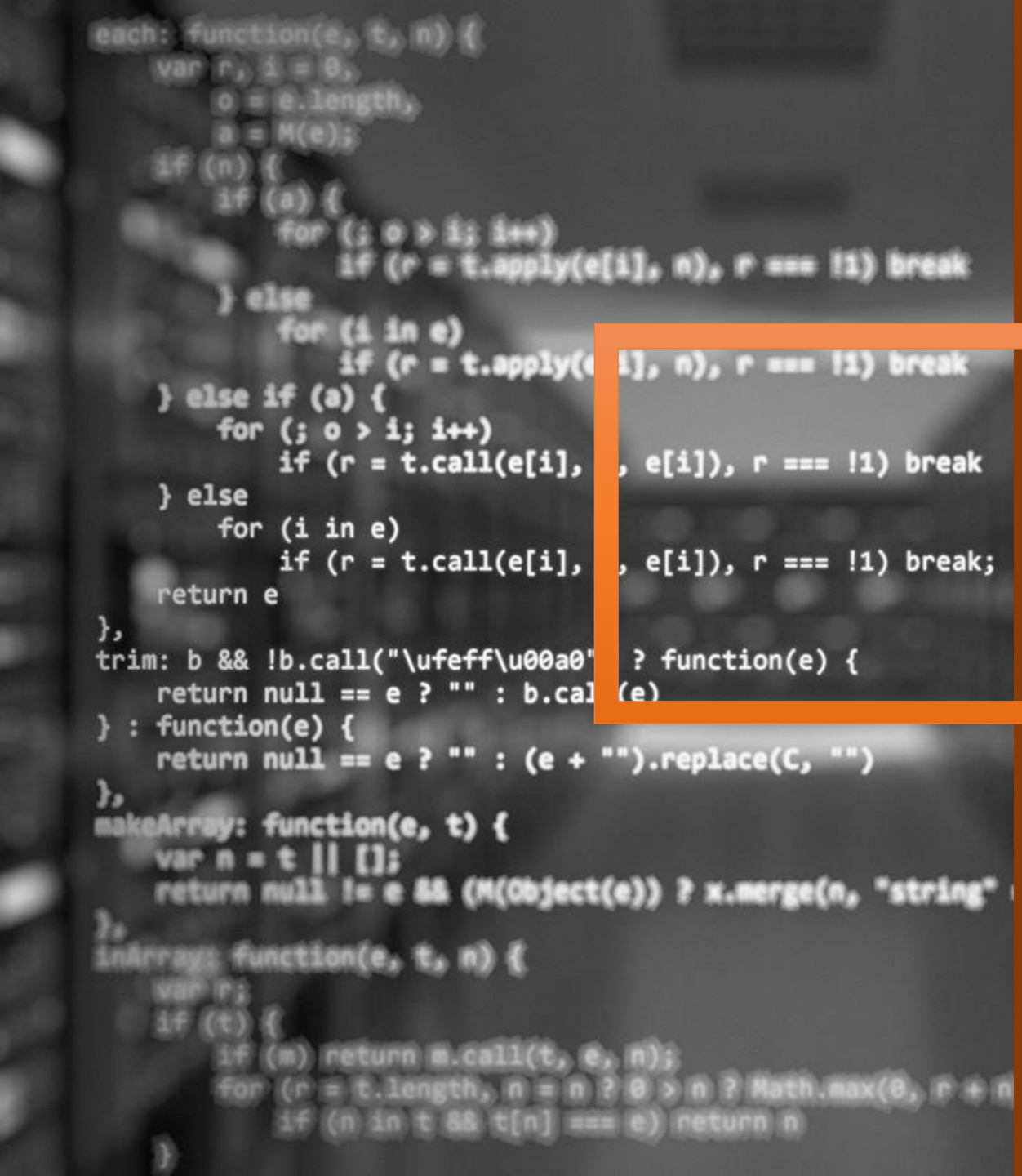
DDL 활용

MySQL

학습목표

1. DDL에 대해서 알 수 있다.
2. 데이터 타입에 대해서 알 수 있다.

목차



1. DDL

DDL

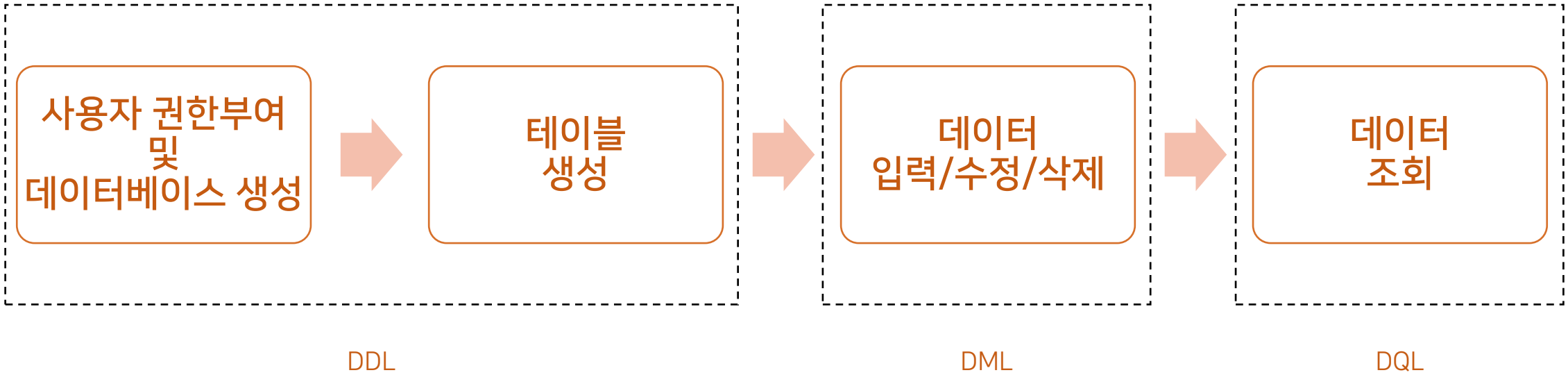
DDL

- Data Definition Language (데이터 정의어)
- 데이터와 데이터 간의 관계를 정의하여 데이터베이스 구조를 설정하는 SQL문
- 테이블(Table), 뷰(View) 등 데이터베이스 객체를 생성/수정/삭제하는 기능을 담당하는 SQL문
- 실행 후 작업을 취소하는 것이 불가능

DDL 종류

- CREATE : 데이터베이스 객체 생성
- ALTER : 데이터베이스 객체 수정
- DROP : 데이터베이스 객체 삭제
- TRUNCATE : 데이터베이스 객체 데이터 및 저장 공간 삭제

데이터베이스 구축 절차



데이터베이스 생성

- 데이터베이스 목록 확인
  - `SHOW DATABASES;`
- 데이터베이스 생성
  - `CREATE DATABASE 데이터베이스_이름;`
- 데이터베이스 삭제
  - `DROP DATABASE 데이터베이스_이름;`
- 데이터베이스 사용
  - `USE 데이터베이스_이름;`

테이블

테이블

- 관계형 데이터베이스에서는 정보를 테이블(릴레이션) 형태로 보관함
- 행(Row)과 열(Column)의 집합체

열(Column), 필드(Field), 속성(Attribute)

행(Row), 레코드(Record), 튜플(Tuple)

고객아이디	이름	전화번호	주소록
0001	김은미	111-1111	서울 마포구
0002	나윤희	222-2222	서울 영등포구
0003	이연수	333-3333	서울 성동구
0004	조서윤	444-4444	서울 강남구

고객 테이블

테이블 생성

테이블 생성

- 테이블에 대한 구조를 정의하고, 데이터를 저장하기 위한 공간을 할당하는 과정
- 테이블을 구성하는 칼럼을 정의하는 과정

테이블 이름 규칙

- 소문자를 이용하길 권장한다.
- 가급적 숫자는 사용하지 않는다.
- 각 단어를 밑줄(_)을 이용해 연결하는 snake case 방식을 사용한다.
- 테이블임을 알리는 prefix나 suffix 값을 사용하길 권장한다.

칼럼 생성

- 칼럼의 데이터 타입
  - INT, VARCHAR, CHAR, DATE, DATETIME 등

- 칼럼 이름 규칙
  - 소문자를 이용하길 권장한다.
  - 가급적 숫자는 사용하지 않는다.
  - 각 단어를 밑줄(_)을 이용해 연결하는 snake case 방식을 사용한다.
  - PK는 id 또는 테이블명_id 형식으로 만들기를 권장한다.
  - 지나친 줄임말은 지양한다.

무결성 제약조건

무결성

- 데이터베이스의 데이터에 문제가 없는 상태를 의미함

무결성 제약조건 (Constraint)

- 데이터 무결성을 지키기 위해 설정하는 제한된 조건을 의미함

무결성 제약조건 종류

- NOT NULL — 필수, NULL값을 허용하지 않음
- UNIQUE — 중복 값을 허용하지 않음
- CHECK — 값의 유효성을 검사
- PRIMARY KEY — 각 레코드를 구별하는 칼럼(NOT NULL + UNIQUE)
- FOREIGN KEY — 다른 테이블의 값을 참조할 때 사용하는 키

AUTO_INCREMENT

- 칼럼을 생성할 때 AUTO_INCREMENT 옵션을 추가하면 해당 칼럼은 1부터 자동으로 1씩 증가하는 정수 값을 가질 수 있음
- AUTO_INCREMENT로 지정한 칼럼은 반드시 PRIMARY KEY 로 지정해 줘야 함
- ALTER 문을 이용하여 시작 값을 변경할 수 있음
  - ALTER TABLE 테이블명 AUTO_INCREMENT=1000;
- @@auto_increment_increment 시스템 변수를 수정하여 증가 값을 변경할 수 있음
  - SET @@auto_increment_increment=2;

테이블 생성 방법

```
DROP TABLE IF EXISTS 테이블_이름;
CREATE  TABLE  테이블_이름 (
        칼럼_이름  데이터_타입  NOT NULL  [ [ AUTO_INCREMENT ] PRIMARY KEY ],
        칼럼_이름  데이터_타입  [ 제약조건 ]  [ Default ],
        FOREIGN KEY(칼럼_이름)  REFERENCES  참조테이블(칼럼_이름)
          [ ON UPDATE Rule ] [ ON DELETE Rule ]
);
```

Rule 의 종류

- RESTRICT : 변경/삭제 시 다른 개체가 참조 중이라면 변경/삭제 취소
- CASCADE : 변경/삭제 시 다른 개체가 참조 중이라면 함께 변경/삭제
- NO ACTION : RESTRICT와 동일
- SET NULL : 변경/삭제 시 다른 개체가 참조 중이라면 해당 참조를 NULL 로 변경

테이블 구조 확인

- DESC
  - 테이블 생성 여부와 테이블의 구조를 확인하기 위한 명령

- tbl_member 테이블 구조 확인

```
DESC tbl_member;
```

Field	Type	Null	Key	Default	Extra
mem_id	char(8)	NO	PRI	NULL	
mem_name	varchar(10)	NO		NULL	
mem_number	tinyint	NO		NULL	
addr	char(2)	NO		NULL	
phone1	char(3)	YES		NULL	
phone2	char(8)	YES		NULL	
height	tinyint unsigned	YES		NULL	
debut_date	date	YES		NULL	

테이블 정의 변경

- 단일 칼럼 추가하기
  - `ALTER TABLE 테이블_이름 ADD COLUMN 칼럼_이름 데이터_타입 [제약조건] [Default];`

- 복수 칼럼 추가하기
  -
```
ALTER TABLE 테이블_이름
  ADD COLUMN 칼럼_이름 데이터_타입 [제약조건] [Default],
  ADD COLUMN 칼럼_이름 데이터_타입 [제약조건] [Default],
  ADD COLUMN 칼럼_이름 데이터_타입 [제약조건] [Default];
```

- 단일 칼럼 수정하기
  - `ALTER TABLE 테이블_이름 MODIFY COLUMN 칼럼_이름 데이터_타입 [제약조건];`

- 기본값 변경하기
  - `ALTER TABLE 테이블_이름 ALTER COLUMN 칼럼_이름 SET DEFAULT [Default];`

테이블 정의 변경

- 칼럼 삭제하기
  - `ALTER TABLE 테이블_이름 DROP COLUMN 칼럼_이름;`
- 칼럼 이름 바꾸기
  - `ALTER TABLE 테이블_이름 RENAME COLUMN 기존_칼럼_이름 TO 새_칼럼_이름;`
- 테이블 이름 바꾸기
  - `RENAME TABLE 기존_테이블_이름 TO 새_테이블_이름;`
  - `ALTER TABLE 기존_테이블_이름 RENAME 새_테이블_이름;`

테이블 삭제

- 테이블 삭제하기
  - `DROP TABLE 테이블_이름;`

- 테이블 존재 확인 후 삭제하기
  - `DROP TABLE IF EXISTS 테이블_이름;`

주의 !!

DROP TABLE 명령어는 테이블을 완전히 삭제한다.
테이블에 데이터가 있을 경우, 모든 데이터가 지워지고
이를 복구할 수 없으므로 주의해서 사용해야 한다.

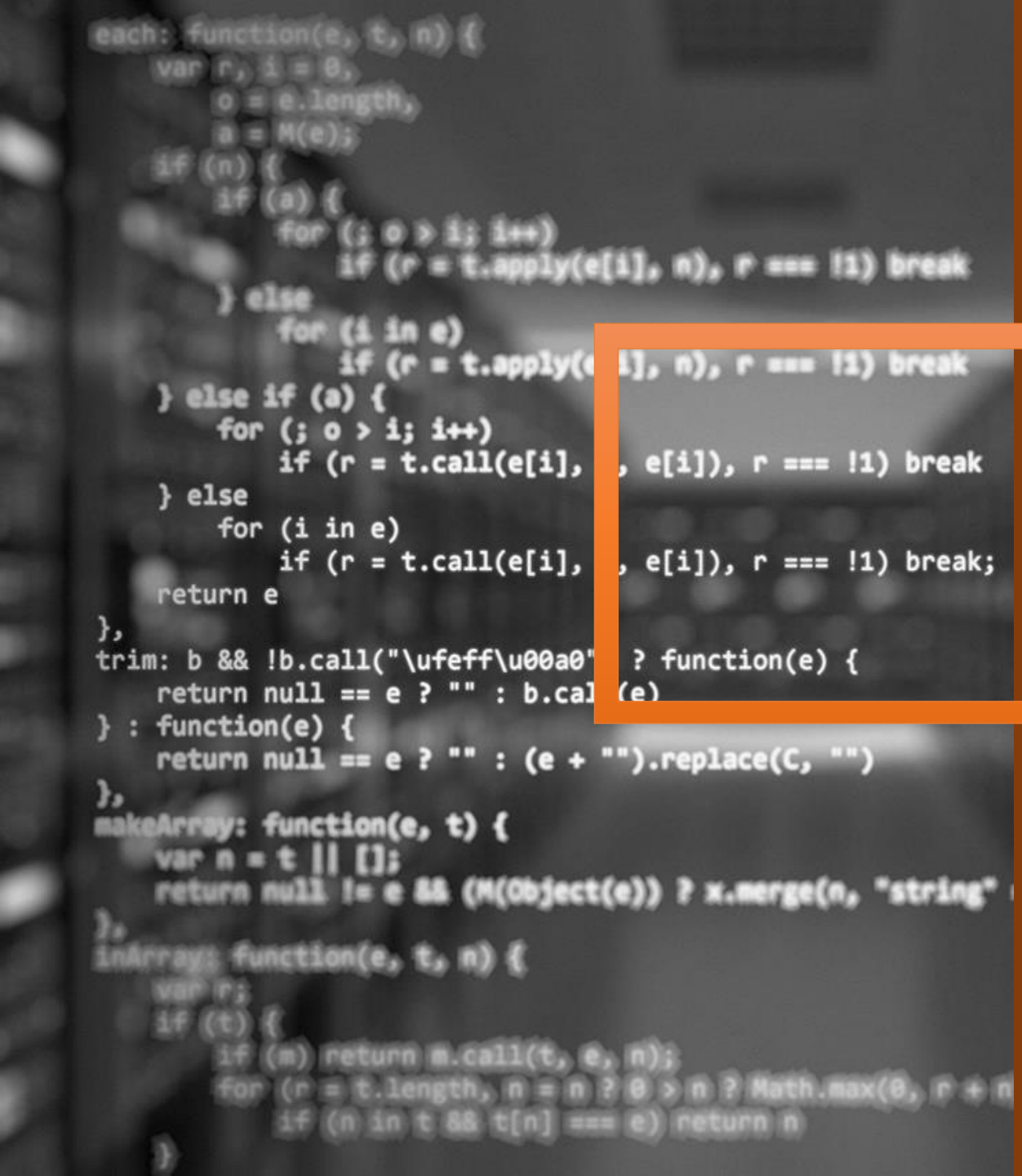
테이블 데이터 삭제

▪ 테이블 데이터 삭제하기

- `TRUNCATE TABLE 테이블_이름;`

주의 !!

TRUNCATE TABLE 명령어는 테이블의 구조는 남기고,
모든 행(ROW) 만 삭제하는 명령이다.
DELETE 문과 비교하면 각 행(ROW)의 삭제 로그가 남지 않기 때문에
삭제 자체는 빠르지만 삭제된 데이터를 복구할 수 없다.
AUTO_INCREMENT 옵션도 초기화된다.



2. 데이터 타입

정수형 데이터 타입

데이터 타입	의미
TINYINT	크기 : 1바이트 범위 : -128 ~ 127
SMALLINT	크기 : 2바이트 범위 : -32,758 ~ 32,767
INT	크기 : 4바이트 범위 : -2,147,483,648 ~ 2,147,483,647
BIGINT	크기 : 8바이트 범위 : 약 -900경 ~ 900경

실수형 데이터 타입

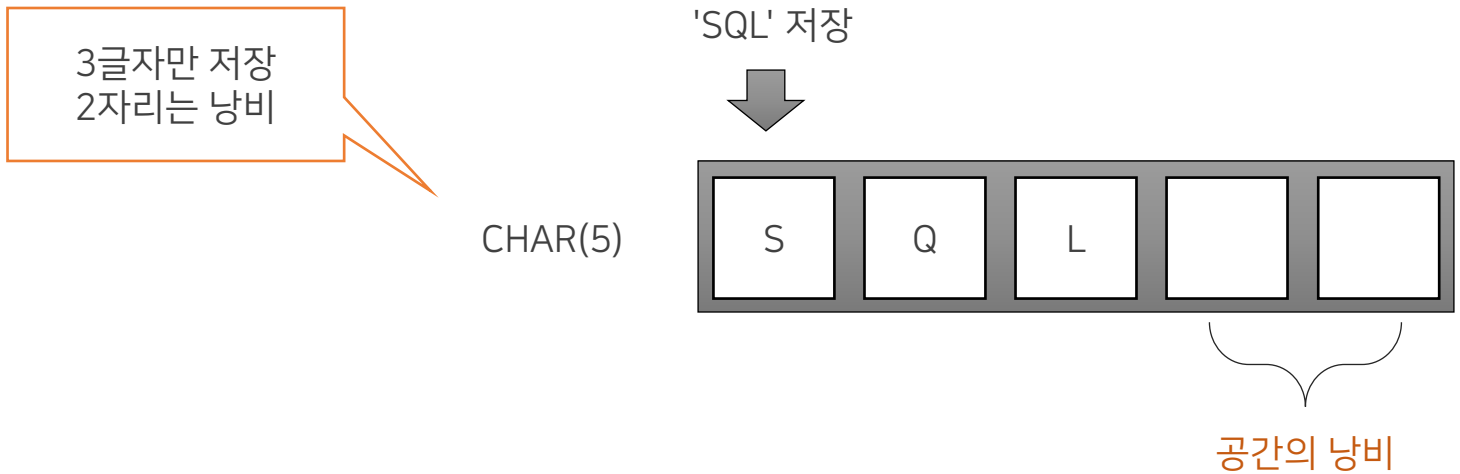
데이터 타입	의미
FLOAT	크기 : 4바이트 소수점 아래 7자리까지 표현 가능
DOUBLE	크기 : 8바이트 소수점 아래 15자리까지 표현 가능

문자형 데이터 타입

데이터 타입	의미
CHAR(size)	size 크기의 고정 길이 문자 타입 최대크기 : 255자
VARCHAR(size)	size 크기의 가변 길이 문자 타입 최대크기 : 16,383자

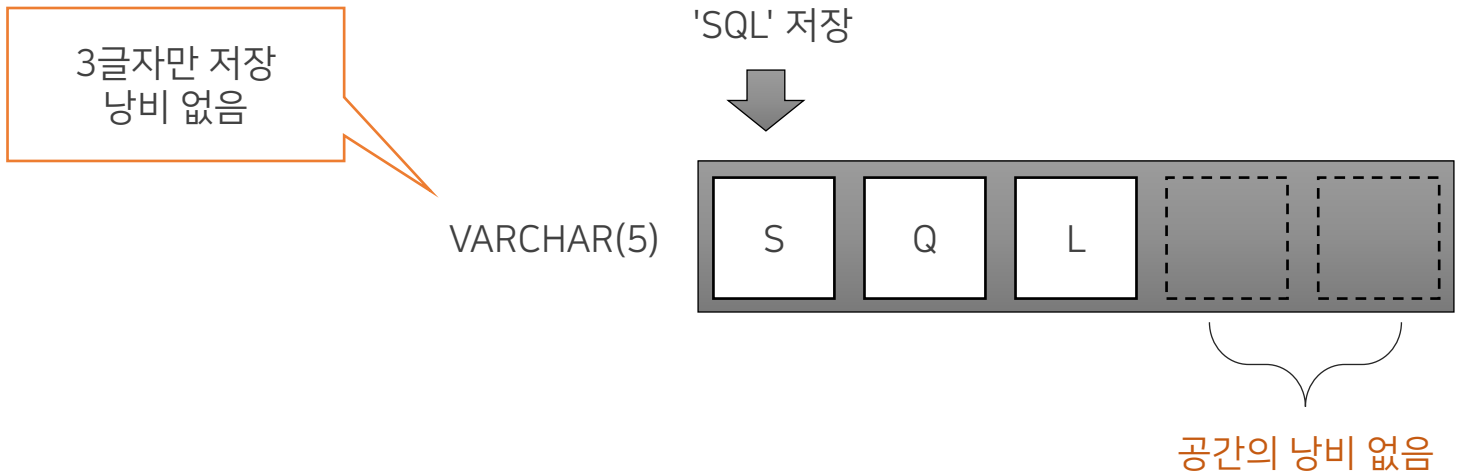
CHAR

- CHAR Type
  - 고정 길이의 문자열 저장 최대 255자
  - 지정된 길이보다 짧은 데이터를 입력해도 지정된 길이를 유지함
  - 길이의 편차가 심한 데이터는 저장 공간의 낭비로 이어짐
  - 주민등록번호, 학번 등 길이가 일정하거나 비슷한 경우에 사용하는 것이 유리



VARCHAR

- VARCHAR Type
  - 가변 길이의 문자열 저장 최대 16,383자
  - 지정된 길이보다 짧은 데이터를 입력하면 데이터만큼의 공간만 사용함 (낭비가 없음)
  - 데이터 길이의 편차가 심한 데이터에서 사용하는 것이 유리
  - 실무에서는 CHAR 타입보다 VARCHAR 타입이 많이 사용



날짜형 데이터 타입

데이터 타입	의미
DATE	크기 : 3바이트 날짜만 저장 YYYY-MM-DD 형식
TIME	크기 : 3바이트 시간만 저장 HH:MM:SS 형식
DATETIME	크기 : 8바이트 날짜 및 시간을 저장 YYYY-MM-DD HH:MM:SS 형식

대용량 데이터 타입

데이터 타입	의미
TEXT	텍스트 데이터 최대 : 65,536자
LONGTEXT	텍스트 데이터 최대 : 4,294,967,295자
BLOB (Binary Long Object)	바이너리 데이터 최대 : 65,536바이트
LONGBLOB	바이너리 데이터 최대 : 4,294,967,295바이트

실습. tbl_customer

- testuser 사용자 생성 및 db_ddl 데이터베이스 권한 부여
  - CREATE USER 'testuser'@'%' IDENTIFIED BY '1234';
  - GRANT ALL PRIVILEGES ON db_ddl.* TO 'testuser'@'%';

- db_ddl 데이터베이스 생성
  - CREATE DATABASE IF NOT EXISTS db_ddl;

- tbl_customer 테이블 생성

칼럼명	데이터 타입	제약조건
cust_id	INT	기본키, 자동증가
cust_name	VARCHAR(30)	필수
phone	VARCHAR(30)	중복불가
age	SMALLINT	0 ~ 100 사이만 가능
join_dt	DATE	기본값 : 현재 날짜