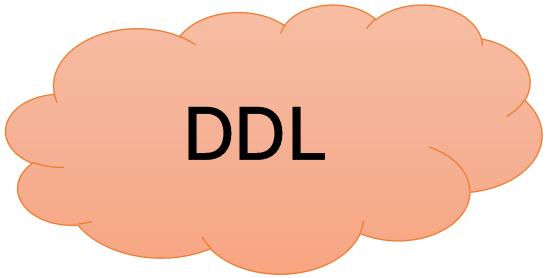
K-Digital Training 웹 풀스택 과정

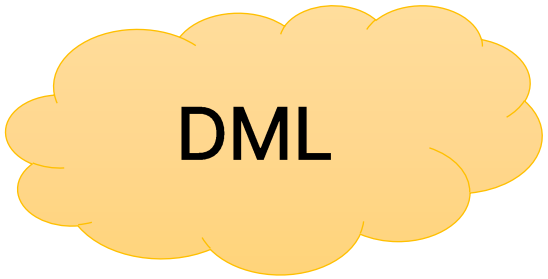
데이터베이스2

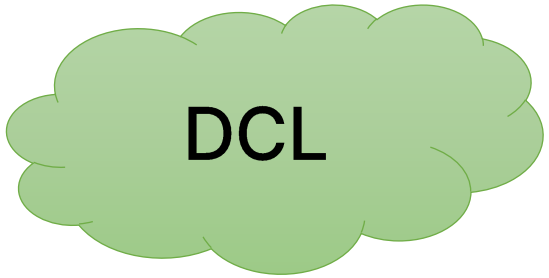
SQL문

SQL문

- Structured Query Language
- 구조적 쿼리 언어
- 관계형 데이터베이스를 제어하고 관리할 수 있는 목적의 프로그래밍 언어

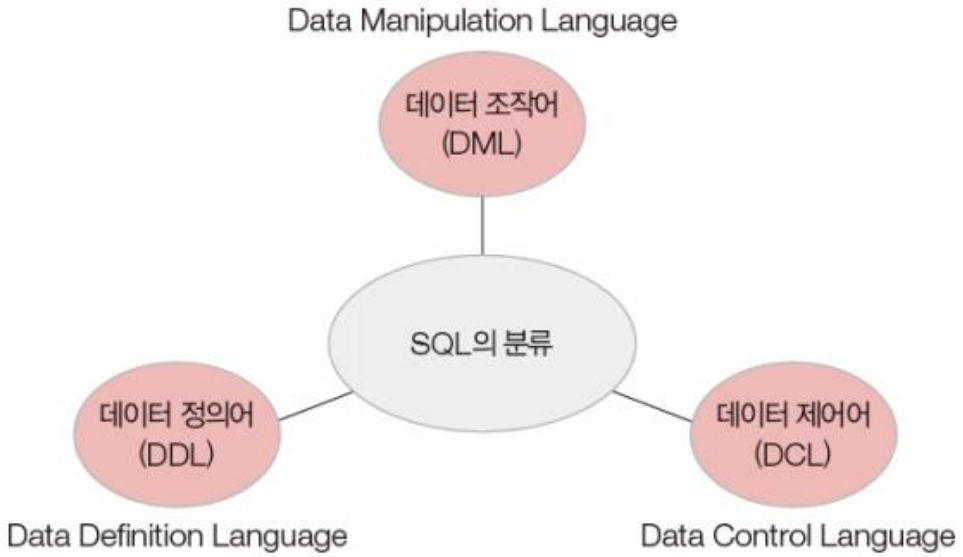






SQL 분류

- 기능에 따른 분류



- 데이터 정의어 (DDL, Data Definition Language)
- 데이터 조작어 (DML, Data Manipulation Language)
- 데이터 제어어 (DCL, Data Control Language)

이미지 출처: https://hoit1302.tistory.com/129

SQL 분류

SQL 분류	명령어	설명
데이터 조작어 (DML)	SELECT	데이터를 조회하거나 검색하기 위한 명령어
	INSERT	데이블의 데이터를 변경하는 명령어. 데이터 삽입, 수정, 삭제
	UPDATE	
	DELETE	
데이터 정의어 (DDL)	CREATE	테이블이나 관계의 구조를 생성하는데 사용하는 명령어
	ALTER	
	DROP	
데이터 제어어 (DCL)	GRANT	데이터베이스에 접근하고 데이터 사용 권한을 주거나 회수하는데 사용하는 명령어
	REVOKE	

데이터 정의어
DDL

SQL문 – DDL

- DDL (Data Definition Language)
- 데이터베이스 or 테이블을 정의하는 언어

종류	역할
CREATE	데이터베이스, 테이블 등을 생성하는 역할을 한다.
ALTER	테이블을 수정하는 역할을 한다.
DROP	데이터베이스, 테이블을 삭제하는 역할을 한다.
TRUNCATE	테이블을 초기화 시키는 역할을 한다.

DDL - CREATE

데이터베이스 만들기 + 한글 인코딩

한글인코딩?

프로그래밍 언어는 영어 기반이므로 한글을 사용할 수 있도록!!

```
CREATE DATABASE 이름 DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
```

한글사용 : create database `demo` default character set utf8 default collate utf8_general_ci;

테이블 만들기

```
CREATE TABLE 테이블명 (
    필드1 값형식,
    필드2 값형식
);
```

주의) 명령어 끝에는 세미콜론(;)을 붙여야한다!!

문장이 끝났음을 알려주기 위함

문자형 데이터 형식

CHAR(n)	고정 길이 데이터 타입(최대 255byte)- 지정된 길이보다 짧은 데이터 입력될 시 나머지 공간 공백으로 채워진다.
VARCHAR(n)	가변 길이 데이터 타입(최대 65535byte)- 지정된 길이보다 짧은 데이터 입력될 시 나머지 공간은 채우지 않는다.
TINYTEXT(n)	문자열 데이터 타입(최대 255byte)
TEXT(n)	문자열 데이터 타입(최대 65535byte)
MEDIUMTEXT(n)	문자열 데이터 타입(최대 16777215byte)
LONGTEXT(n)	문자열 데이터 타입(최대 4294967295byte)

숫자형 데이터 형식

TINYINT(n)	정수형 데이터 타입(1byte) -128 ~ +127 또는 0 ~ 255수 표현 가능하다.
SMALLINT(n)	정수형 데이터 타입(2byte) -32768 ~ 32767 또는 0 ~ 65536수 표현 가능하다.
MEDIUMINT(n)	정수형 데이터 타입(3byte) -8388608 ~ +8388607 또는 0 ~ 16777215수 표현 가능하다.
INT(n)	정수형 데이터 타입(4byte) -2147483648 ~ +2147483647 또는 0 ~ 4294967295수 표현 가능하다.
BIGINT(n)	정수형 데이터 타입(8byte) - 무제한 수 표현 가능하다.
FLOAT(길이,소수)	부동 소수형 데이터 타입(4byte) -고정 소수점을 사용 형태이다.
DECIMAL(길이,소수)	고정 소수형 데이터 타입고정(길이+1byte) -소수점을 사용 형태이다.
DOUBLE(길이,소수)	부동 소수형 데이터 타입(8byte) -DOUBLE을 문자열로 저장한다

날짜형 데이터 형식

DATE	날짜(년도, 월, 일) 형태의 기간 표현 데이터 타입(3byte)
TIME	시간(시, 분, 초) 형태의 기간 표현 데이터 타입(3byte)
DATETIME	날짜와 시간 형태의 기간 표현 데이터 타입(8byte)
TIMESTAMP	날짜와 시간 형태의 기간 표현 데이터 타입(4byte) -시스템 변경 시 자동으로 그 날짜와 시간이 저장된다.
YEAR	년도 표현 데이터 타입(1byte)

DDL - CREATE

```sql
CREATE TABLE member (
  id VARCHAR(10) NOT NULL PRIMARY KEY,
  name VARCHAR(10) NOT NULL,
  birthday DATE NOT NULL
);
```

DDL - ALTER

- 테이블의 특정 컬럼(열)을 삭제하거나 추가, 변경할 때 사용하는 명령어

```
# 1. 컬럼 삭제
ALTER TABLE 테이블명 DROP COLUMN 컬럼명;

# 2. 컬럼 추가
ALTER TABLE 테이블명 ADD 컬럼명 타입

# 3. 컬럼 속성 변경
ALTER TABLE 테이블명 MODIFY 컬럼명 타입;
```

DDL – DROP vs. TRUNCATE

```
DROP TABLE 테이블명;
```

- 테이블 삭제하기
- 테이블을 잘못 만들었거나 더 이상 필요 없는 경우

```
TRUNCATE TABLE 테이블명;
```

- 테이블 초기화하기 (테이블의 모든 행(row) 일괄 삭제)

이미지 출처: https://lee-mandu.tistory.com/476

SQL 공통

```
# 데이터베이스 목록 보기
SHOW DATABASES;

# 데이터베이스 이용하기
USE 데이터베이스명;

# 테이블 목록 보기
SHOW TABLES;

# 테이블 구조 보기
DESC 테이블명;
```

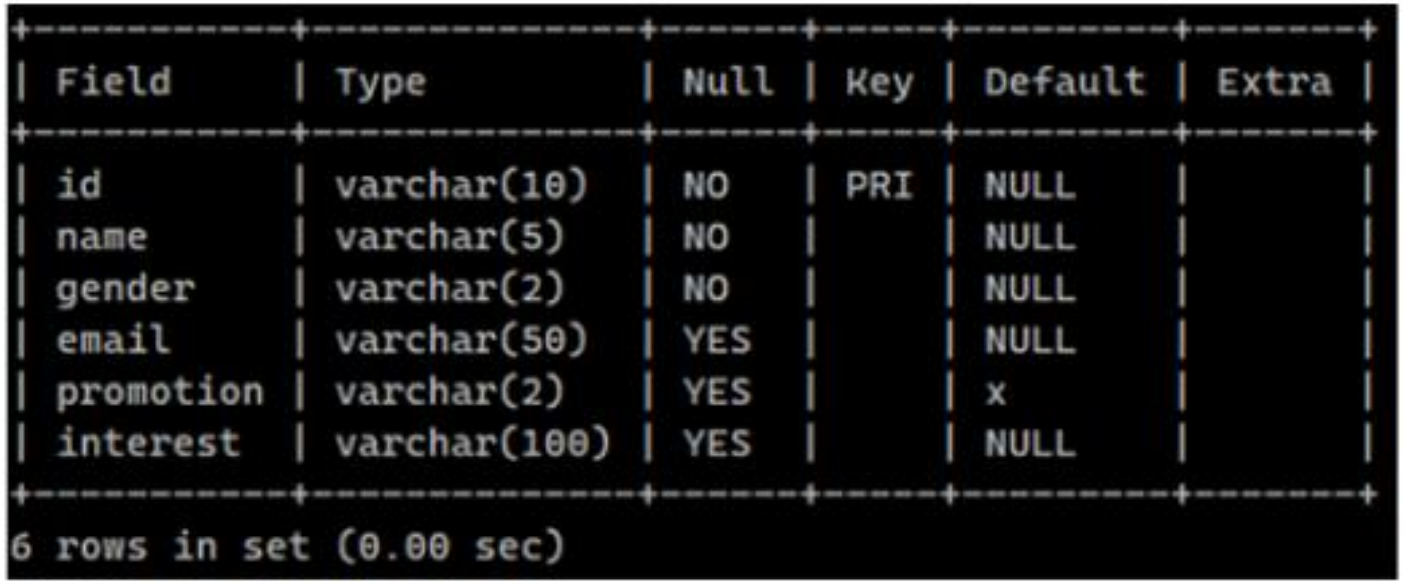
실습1. DDL

- DDL을 이용해 아래 테이블 완성하기
- 테이블명: member

```
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| id        | varchar(20) | NO   | PRI | NULL    |       |
| name      | varchar(5)  | NO   |     | NULL    |       |
| age       | int         | YES  |     | NULL    |       |
| gender    | varchar(2)  | NO   |     | NULL    |       |
| email     | varchar(50) | YES  |     | NULL    |       |
| promotion | varchar(2)  | YES  |     | x       |       |
+-----------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

실습2. DDL

- 이전 실습에서 생성한 테이블을 ALTER 명령어를 이용해 구조 변경



Field	Type	Null	Key	Default	Extra
id	varchar(10)	NO	PRI	NULL	
name	varchar(5)	NO		NULL	
gender	varchar(2)	NO		NULL	
email	varchar(50)	YES		NULL	
promotion	varchar(2)	YES		x	
interest	varchar(100)	YES		NULL	

6 rows in set (0.00 sec)

*힌트
- id 컬럼: 값 형식 변경
- age 컬럼: 삭제
- interest 컬럼: 추가

데이터 정의어
DML

SQL문 – DML

- DML (Data Manipulation Language)
- 데이터베이스의 내부 데이터를 관리하기 위한 언어

종류	역할
SELECT	데이터베이스에서 데이터를 검색(조회)하는 역할을 한다.
INSERT	테이블에 데이터를 추가하는 역할을 한다.
UPDATE	테이블에서 데이터를 수정하는 역할을 한다.
DELETE	테이블에서 데이터를 삭제하는 역할을 한다.

CRUD

- 대부분의 컴퓨터 소프트웨어가 가지는 기본적인 처리 기능
- Create(생성)
- Read(읽기)
- Update(갱신)
- Delete(삭제)

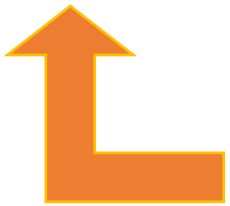
이름	조작	SQL
Create	생성	INSERT
Read	읽기	SELECT
Update	갱신	UPDATE
Delete	삭제	DELETE

DML - INSERT

- 테이블에 데이터를 추가하기 위해 사용

방법1
```
INSERT INTO 테이블명 (필드1, 필드2, 필드3) VALUES (값1, 값2, 값3);
```

방법2
```
INSERT INTO 테이블명 VALUES(값1, 값2, 값3);
```



필드를 명시하지 않을 때는 테이블의 모든 컬럼에 값을 추가할 때만 사용할 수 있다.

SELECT 문

- 데이터를 검색하는 기본 문장
- 질의어 (query) 라고도 함
- SQL 문 중 가장 많이 사용되는 문법

```
SELECT 속성이름, ... FROM 테이블이름 [WHERE 검색조건]
```

*대괄호([]) 안의 SQL 예약어는 선택적으로 사용 가능

SQL 문 내부적 실행 순서

- 홍지수 고객의 주소를 찾으시오.

```
SELECT addr FROM customer WHERE custname='홍지수';
```

(1) FROM customer

	custid	custname	addr	phone	birth
▶	bunny	강해린	대한민국 서울	01012341234	2000-02-23
	hello	이지민	대한민국 포항	01022221234	1999-08-08
	imminji01	강민지	영국 런던	01060001000	1995-01-11
	jjjeee	홍은정	대한민국 서울	01099991111	2004-08-17
	jy9987	강지연	일본 삿포로	01012312323	1996-09-01
	kiwi	최지수	미국 뉴욕	01050005000	1990-12-25
	lalala	홍지수	미국 로스앤젤레스	01010109090	2007-05-16
	minjipark	박민지	프랑스 파리	01088776655	1998-04-08
	wow123	이민혁	일본 삿포로	01011223344	1994-05-31

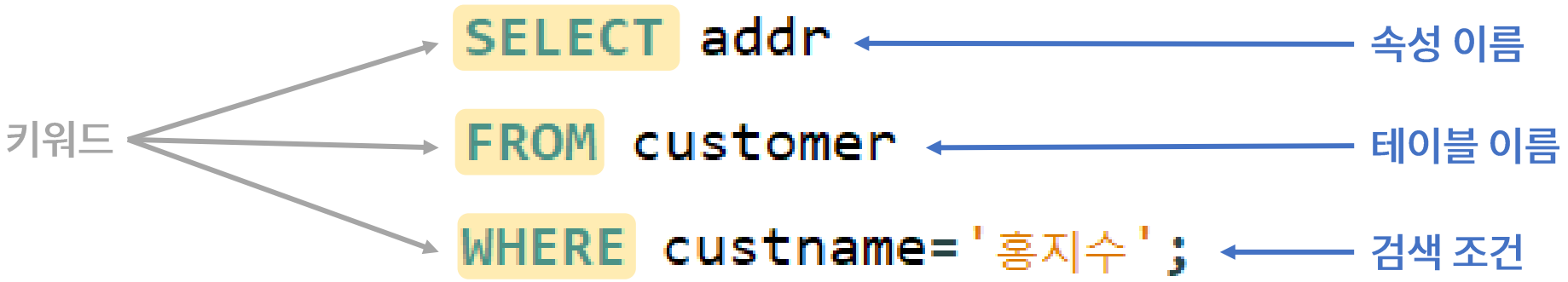
(2) WHERE custname='홍지수'

	custid	custname	addr	phone	birth
▶	lalala	홍지수	미국 로스앤젤레스	01010109090	2007-05-16

(3) SELECT addr

	addr
▶	미국 로스앤젤레스

SELECT 문 구성 요소



DML - SELECT

- 데이터를 검색(조회)하기 위해 사용

```
SELECT * FROM 테이블명;
SELECT * FROM 테이블명 WHERE 필드1 = 조건값1;
SELECT * FROM 테이블명 WHERE 필드1 = 조건값1 ORDER BY 필드1 ASC;
SELECT 필드1, 필드2, 필드3 FROM 테이블명 WHERE 필드1 = 조건값1 ORDER BY 필드1 ASC;
SELECT 필드1, 필드2, 필드3 FROM 테이블명 WHERE 필드1 = 조건값1 ORDER BY 필드1 ASC LIMIT 개수;
```

WHERE 절 - 비교연산자

=	같다
>	보다 크다
>=	보다 크거나 같다
<	보다 작다
<=	보다 작거나 같다

WHERE 절 – 부정연산자

!=	같지 않다.
^=	같지 않다.
<>	같지 않다.
NOT 컬럼명 =	~와 같지 않다.

WHERE 절 – 범위, 집합, 패턴, NULL

BETWEEN a AND b	a와 b의 값 사이에 있으면 참 (a와 b 값도 포함)
IN (list)	리스트에 있는 값 중에서 어느 하나라도 일치하면 참
LIKE '비교문자열'	비교문자열과 형태가 일치하면 사용 (%, _ 사용) • `%` : 0개 이상의 어떤 문자 • `_` : 1개의 단일문자
IS NULL	NULL 값인 경우 true, 아니면 false

WHERE 절 – 논리연산자

AND	앞에 있는 조건과 뒤에 오는 조건이 참(TRUE)가 되면 결과도 참(TRUE)
OR	앞에 있는 조건과 뒤에 오는 조건 중 하나라도 참(TRUE)면 결과는 참(TRUE)
NOT	뒤에 오는 조건과 반대되는 결과를 돌려준다.

ORDER BY

- 결과가 출력되는 순서 조절
- where 절과 함께 사용 가능
  - 단, where 절 뒤에 나와야 함

```
SELECT 속성이름, ...
  FROM 테이블이름
  [WHERE 검색조건]
  [ORDER BY 속성이름]
```

- ASC: Ascending, 오름차순 (기본값)
- DESC: Descending, 내림차순

DISTINCT

- 중복된 데이터 제거

```
SELECT [DISTINCT] 속성이름, ...
  FROM 테이블이름
  [WHERE 검색조건]
  [ORDER BY 속성이름]
```

LIMIT

- 출력 개수 제한

```
SELECT [DISTINCT] 속성이름, ...
  FROM 테이블이름
  [WHERE 검색조건]
  [ORDER BY 속성이름]
  [LIMIT 개수]
```

집계 함수

SUM()	합계
AVG()	평균
MAX()	최대값
MIN()	최소값
COUNT()	행 개수
COUNT(DISTINCT)	중복 제외한 행 개수

GROUP BY

- group by
  - 속성이름끼리 그룹으로 묶는 역할
- having
  - group by절의 결과를 나타내는 그룹의 조건 걸기

```
SELECT [DISTINCT] 속성이름, ...
  FROM 테이블이름
  [WHERE 검색조건]
  [GROUP BY] 속성이름
  [HAVING] 조건식
  [ORDER BY 속성이름]
  [LIMIT 개수]
```

DML – UPDATE, DELETE

UPDATE

- 데이터를 수정하기 위해 사용

```
UPDATE 테이블명 SET 필드1=값1 WHERE 필드2=조건2;
```

DELETE

- 데이터를 삭제하기 위해 사용

```
DELETE FROM 테이블명 WHERE 필드1=값1;
```

실습3. CREATE 문

- 아래 조건을 충족하는 user 테이블 생성

```
테이블명 : user

컬럼수 : 6개

컬럼 설명 > 컬럼명, 종류(길이), 비고 순
☞ 1번째 컬럼 : id VARCHAR(10) not null primary key
☞ 2번째 컬럼 : pw VARCHAR(20) not null
☞ 3번째 컬럼 : name VARCHAR(5) not null
☞ 4번째 컬럼 : gender ENUM( 'F', 'M', '' ) default ''
☞ 5번째 컬럼 : birthday DATE not null
☞ 6번째 컬럼 : age int(3) not null default 0
```

실습4. INSERT 문

- 이전 실습에서 생성한 user 테이블에서 INSERT 문을 이용해 데이터 추가
- SELECT * FROM user 명령어를 이용해 모든 회원 목록 출력

```
회원명부 > id, pw, name, gender, birthday, age 순

hong1234, 8o4bkg, 홍길동, M, 1990-01-31, 33
sexysung, 87awjkdf, 성춘향, F, 1992-03-31, 31
power70, qxur8sda, 변사또, M, 1970-05-02, 53
hanjo, jk48fn4, 한조, M, 1984-10-18, 39
widowmaker, 38ewifh3, 위도우, F, 1976-06-27, 47
dvadva, k3f3ah, 송하나, F, 2001-06-03, 22
jungkrat, 4ifha7f, 정크랫, M, 1999-11-11, 24
```

실습5. SELECT 문

- 이전에 생성한 user 테이블에서 다음 조건을 만족하는 SQL 문 작성하기

1. 모든 회원목록을 가져오는데, 이때 birtday 컬럼의 값을 기준으로 오름차순 정렬하여 가져오시오.
2. 회원 목록 중 gender 컬럼의 값이 "M" 인 회원목록을 가져오는데, 이때 name 컬럼의 값을 기준으로 내림차순 정렬하여 가져오시오.
3. 1990 년대에 태어난 회원의 id, name 컬럼을 가져와 목록으로 보여주시오.
4. 6월생 회원의 목록을 birthday 기준으로 오름차순 정렬하여 가져오시오.
5. gender 컬럼의 값이 "M" 이고, 1970 년대에 태어난 회원의 목록을 가져오시오.
6. 모든 회원목록 중 age를 기준으로 내림차순 정렬하여 가져오는데, 그때 처음 3개의 레코드만 가져오시오.
7. 모든 회원 목록 중 나이가 25 이상 50 이하인 회원의 목록을 출력하시오.
8. id 컬럼의 값이 hong1234 인 레코드의 pw 컬럼의 값을 12345678로 변경하시오.
9. id 컬럼의 값이 jungkrat인 레코드를 삭제하시오.

JOIN

JOIN

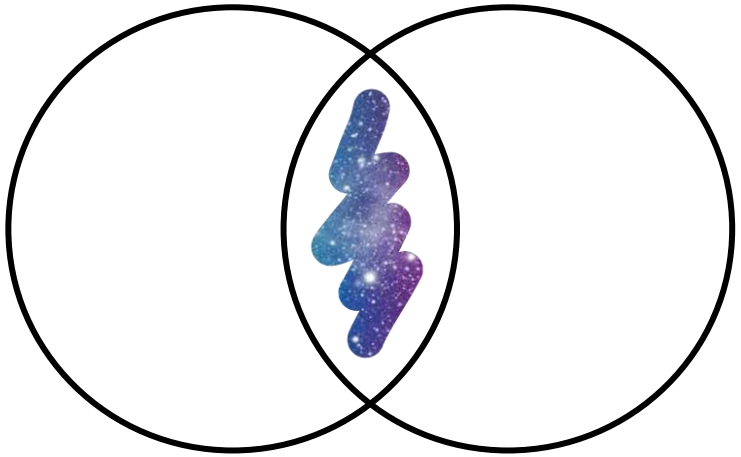
- 두 테이블을 묶어서 하나의 테이블을 만듦
- 왜? 두 테이블을 엮어야 원하는 형태가 나오기도 함

JOIN 종류

1. Inner Join
2. Left Outer Join
3. Right Outer Join

Outer Join

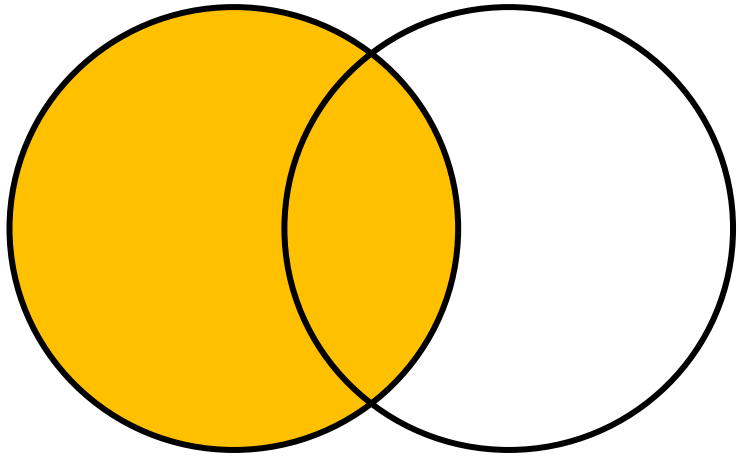
Inner Join



```
SELECT 속성이름, ...
  FROM 테이블A, 테이블B
  WHERE 조인조건 AND 검색조건;
```

```
SELECT 속성이름, ...
  FROM 테이블A INNER JOIN 테이블B ON 조인조건
  WHERE 검색조건;
```

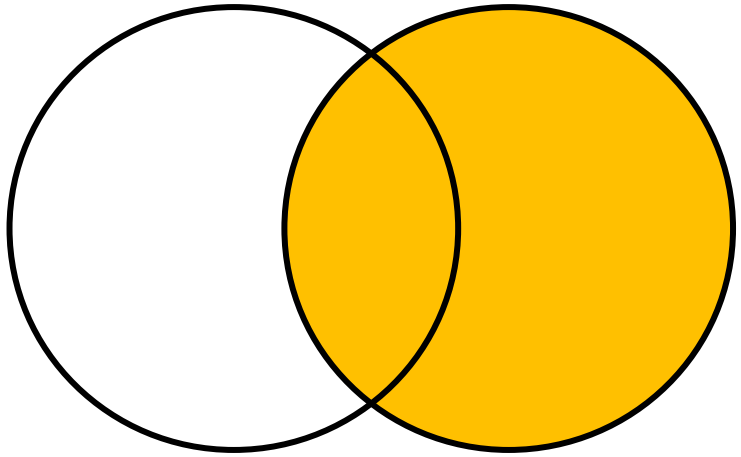
Left Outer Join



*Outer Join은 Inner Join과 다르게 공통되지 않은 row도 유지한다

```
SELECT 속성이름, ... FROM 테이블A LEFT [OUTER] JOIN 테이블B ON 조인조건
```

Right Outer Join



```
SELECT 속성이름, ... FROM 테이블A RIGHT [OUTER] JOIN 테이블B ON 조인조건
```

데이터 제어어
DCL

데이터 제어어 (DCL)

- Data Control Language
- 데이터베이스에 접근해 읽거나 쓰는 것을 제한할 수 있는 권한을 부여 or 박탈

GRANT

- 특정 데이터베이스 사용자에게 특정 작업에 대한 수행 권한 부여

```
GRANT permission_type ON db_name.table_name
  TO username@host IDENTIFIED BY 'pw' [WITH GRANT OPTION];
```

REVOKE

- 특정 데이터베이스 사용자에게 특정 작업에 대한 권한 박탈

```
REVOKE permission_type ON db_name.table_name FROM 'username'@'host';
```