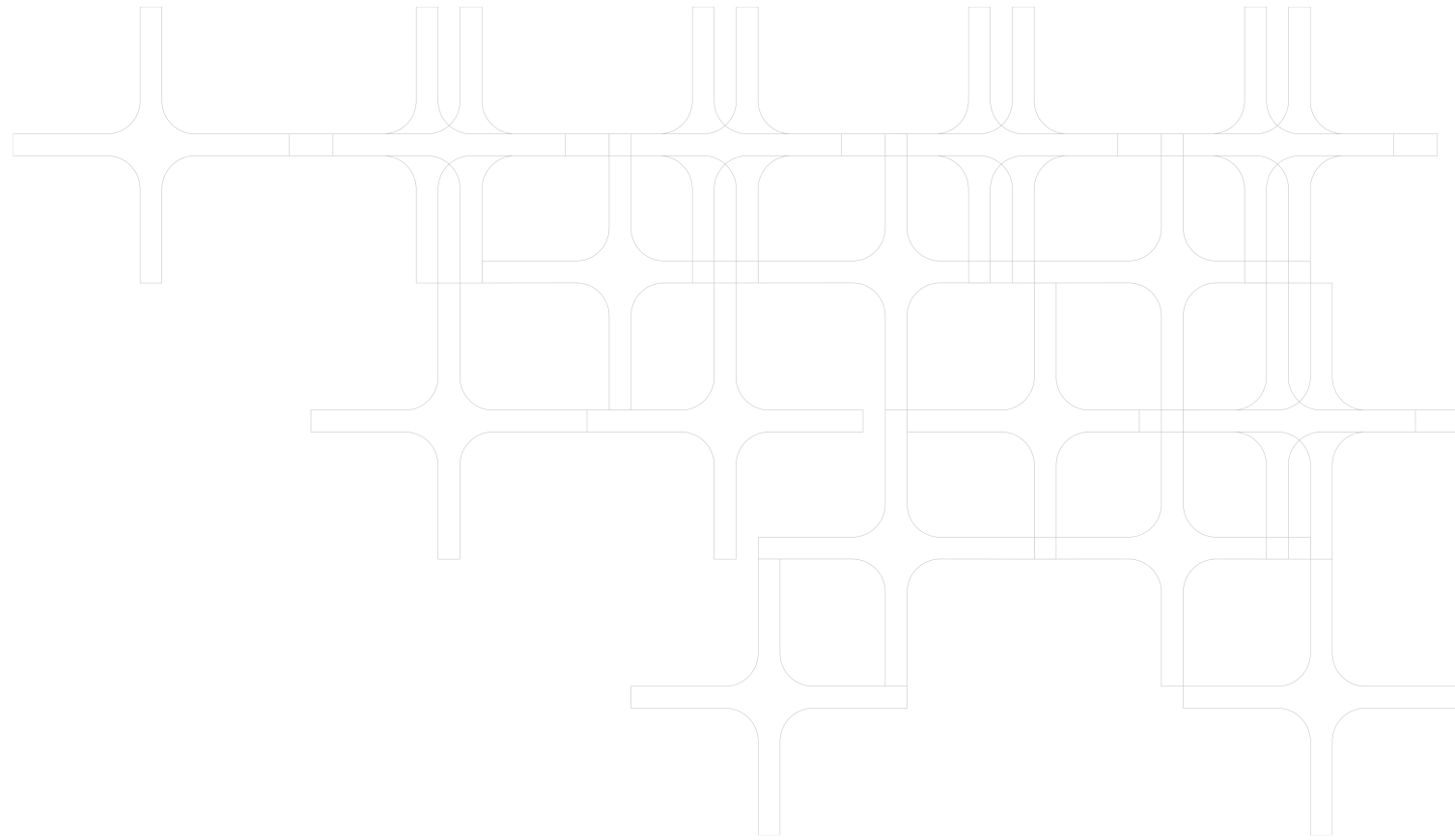


# SQL 활용

---

## 연산자 및 함수



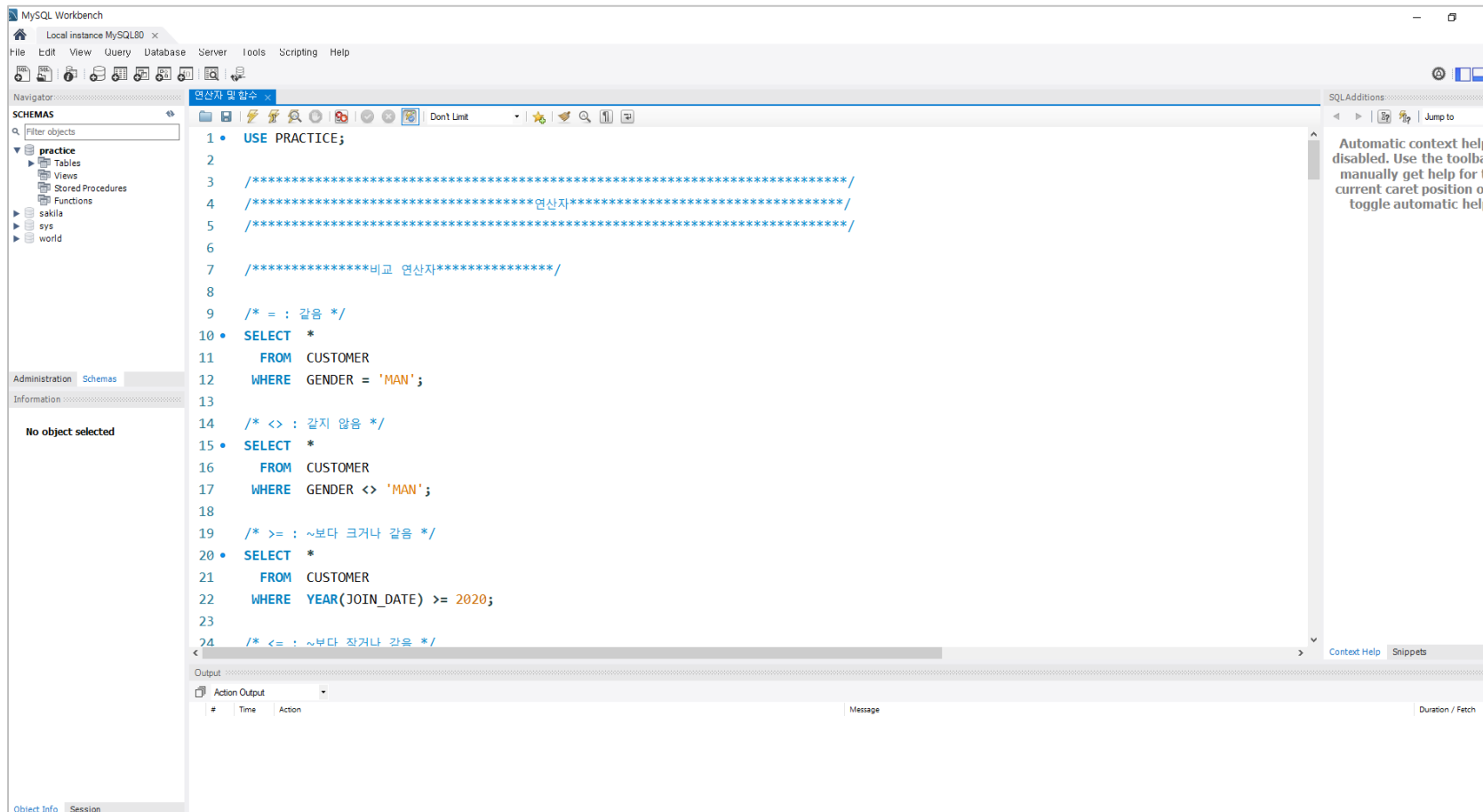
# 연산자 및 함수

이번 영상에서는  
연산자 및 함수 개념을 학습합니다.

연산자는 비교, 논리, 특수, 산술, 집합 연산자로 나뉩니다.

구분	연산자	설명
비교 연산자	= / > / < / >= / <= / <>	같음 / 보다 큼 / 보다 작음 / 크거나 같음 / 작거나 같음 / <b>같지 않음</b>
논리 연산자	AND / OR	앞, 뒤 조건 모두 만족 / 하나라도 만족
	NOT	뒤에 오는 조건과 <b>반대</b>
특수 연산자	BETWEEN a AND b / NOT BETWEEN a AND b	a와 b의 값 사이 / a와 b의 값 사이가 <b>아님</b>
	IN (List) / NOT IN (List)	리스트(List) 값 / 리스트(List) 값이 <b>아님</b>
	LIKE '비교문자열'	비교문자열과 같음
	NOT LIKE '비교문자열'	비교문자열이 <b>아님</b>
	IS NULL	NULL과 같음
	IS NOT NULL	NULL이 <b>아님</b>
산술 연산자	+, -, *, /	덧셈, 뺄셈, 곱셈, 나눗셈
집합 연산자	UNION	2개 이상 테이블 중복된 행 제거 하여 집합(* 열 개수와 데이터 타입 일치)
	UNION ALL	2개 이상 테이블 중복된 행 제거 <b>없이</b> 집합(* 열 개수와 데이터 타입 일치)

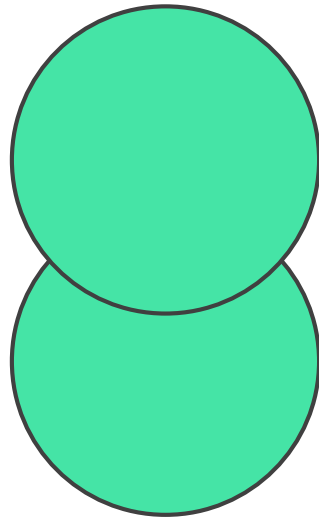
연산자는 **비교**, **논리**, **특수**, **산술**, **집합** 연산자로 나뉩니다.



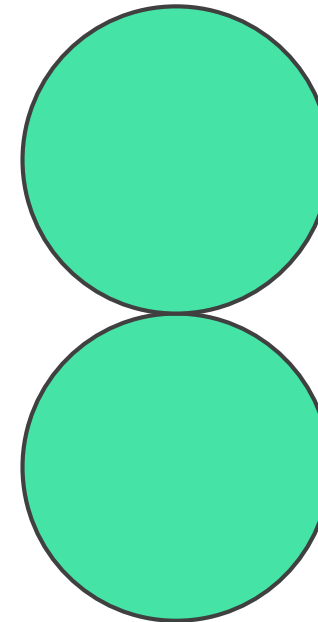
- UNION: 2개 이상 테이블의 중복된 행들을 제거 하여 집합
- UNION ALL: 2개 이상 테이블의 중복된 행들을 제거 **없이** 집합

\* 열 개수와 데이터 타입 일치

< UNION >



< UNION ALL >



- UNION: 2개 이상 테이블의 중복된 행들을 제거 하여 집합
- UNION ALL: 2개 이상 테이블의 중복된 행들을 제거 **없이** 집합

\* 열 개수와 데이터 타입 일치

< UNION >

order_no	mem_no	order_date	product_code	sales_qty
1	1000970	2019-05-02	505	2
2	1000970	2019-05-03	75	3



order_no	mem_no	order_date	product_code	sales_qty
2	1000970	2019-05-03	75	3
3	1000021	2019-05-07	494	2



order_no	mem_no	order_date	product_code	sales_qty
1	1000970	2019-05-02	505	2
2	1000970	2019-05-03	75	3
3	1000021	2019-05-07	494	2

< UNION ALL >

order_no	mem_no	order_date	product_code	sales_qty
1	1000970	2019-05-02	505	2
2	1000970	2019-05-03	75	3

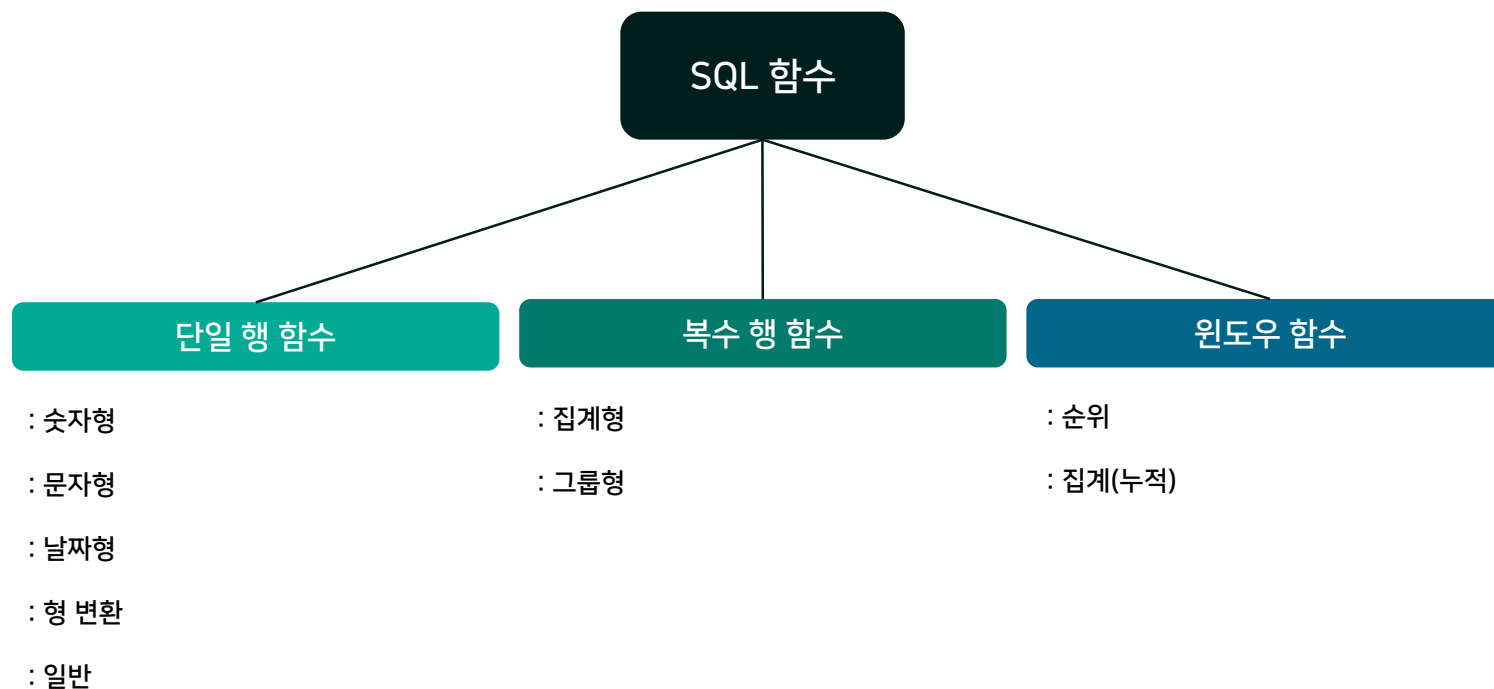


order_no	mem_no	order_date	product_code	sales_qty
2	1000970	2019-05-03	75	3
3	1000021	2019-05-07	494	2



order_no	mem_no	order_date	product_code	sales_qty
1	1000970	2019-05-02	505	2
2	1000970	2019-05-03	75	3
2	1000970	2019-05-03	75	3
3	1000021	2019-05-07	494	2

함수는 단일 및 복수 행 그리고 윈도우 함수로 나뉘며, **특정 규칙**에 의해 새로운 결과값으로 반환하는 명령어입니다.



단일 행 함수는 모든 행에 대하여 **각각** 함수가 적용되어 반환합니다.

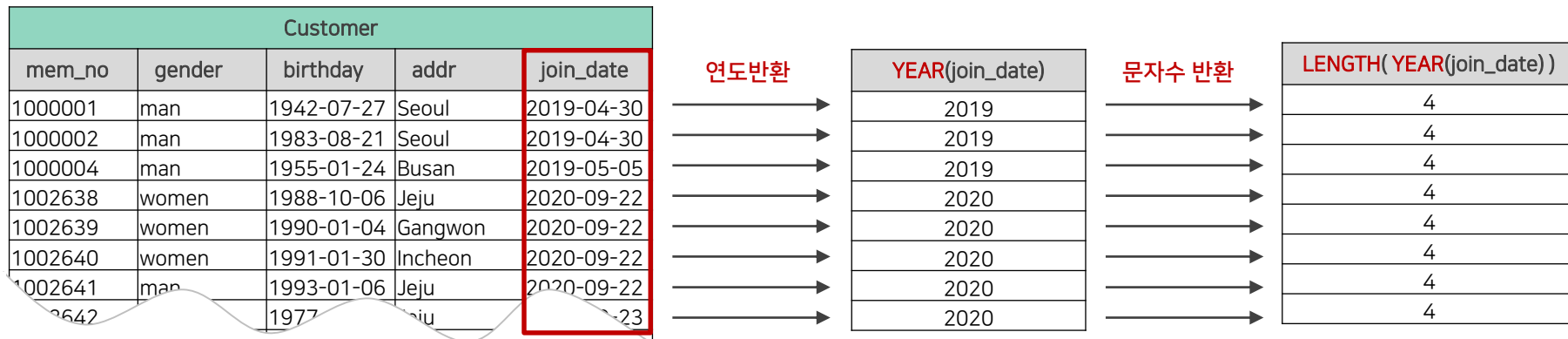
Customer					연도반환	YEAR(join_date)
mem_no	gender	birthday	addr	join_date		
1000001	man	1942-07-27	Seoul	2019-04-30	→	2019
1000002	man	1983-08-21	Seoul	2019-04-30	→	2019
1000004	man	1955-01-24	Busan	2019-05-05	→	2019
1002638	women	1988-10-06	Jeju	2020-09-22	→	2020
1002639	women	1990-01-04	Gangwon	2020-09-22	→	2020
1002640	women	1991-01-30	Incheon	2020-09-22	→	2020
1002641	man	1993-01-06	Jeju	2020-09-22	→	2020
1002642	man	1977-01-23	Seoul	2020-09-23	→	2020



단일 행 함수는 숫자형, 문자형, 날짜형, 형변환, 일반함수로 나뉩니다.

구분	함수	설명	함수 사용	반환값
숫자형 함수	ABS(숫자)	절대값 반환	ABS(-200)	200
	ROUND(숫자, N)	N 기준으로 반올림 값 반환	ROUND(2.18, 1)	2.2
	SQRT(숫자)	제곱근 값 반환	SQRT(9)	3
문자형 함수	LOWER(문자) / UPPER(문자)	소문자 / 대문자 반환	LOWER('AB') / UPPER('ab')	ab / AB
	LEFT(문자, N) / RIGHT(문자, N)	왼쪽 / 오른쪽부터 N만큼 반환	LEFT('AB', 1) / RIGHT('AB', 1)	A, B
	LENGTH(문자)	문자수 반환	LENGTH('AB')	2
날짜형 함수	YEAR / MONTH / DAY(날짜)	연 / 월 / 일 반환	YEAR / MONTH / DAY ('2022-12-31')	2022 / 12 / 31
	DATE_ADD(날짜, INTERVAL)	INTERVAL만큼 더한 값 반환	DATEADD('2022-12-31', INTERVAL -1 MONTH)	2020-11-30
	DATEDIFF(날짜a, 날짜b)	날짜a - 날짜b 일수 반환	DATEDIFF('2022-12-31', '2022-12-1');	30
형변환 함수	DATE_FORMAT(날짜, 형식)	날짜형식으로 변환	DATE_FORMAT('2022-12-31', '%m-%d-%y');	12-31-22
	CAST(형식a, 형식b)	형식a를 형식b로 변환	CAST('2022-12-31 12:00:00' AS DATE);	2022-12-31
일반 함수	IFNULL(A, B)	A가 NULL이면 B를 반환, 아니면 A 반환	IFNULL(NULL, 0);	0
	CASE WHEN [조건1] THEN [반환1] WHEN [조건2] THEN [반환2] ELSE [나머지] END	여러 조건별로 반환값 지정	SELECT * ,CASE WHEN GENDER = 'MAN' THEN '남성' ELSE '여성' END FROM CUSTOMER;	MAN은 남성 나머지는 여성

- 단일 행 함수는 모든 행에 대하여 **각각** 함수가 적용되어 반환합니다.
- 함수들을 **중첩**해서 사용가능합니다.



복수 행 함수는 여러 행들이 **하나의 결과값**으로 반환합니다.

Sales				
order_no	mem_no	order_date	product_code	sales_qty
1	1000970	2019-05-02	505	2
2	1000970	2019-05-03	75	3
3	1000021	2019-05-07	494	2
4	1000023	2019-05-07	644	2
5	1000025	2019-05-07	518	2
6	1000031	2019-05-07	518	1
7	1000037	2019-05-07	455	1
8	1000029	2019-05-08	614	2
		2019-05-08	200	

구매수량 반환

<b>SUM(sales_qty)</b>
20,823

복수 행 함수는 집계, 그룹 함수로 나뉩니다.

구분	함수	설명	함수 사용
집계 함수	COUNT * COUNT(DISTINCT)	행수 * 행수(중복제거)	COUNT(열) * COUNT(DISTINCT 열)
	SUM	합계	SUM(열)
	AVG	평균	AVG((열)
	MAX / MIN	최대 / 최소	MAX(열) / MIN(열)
그룹함수	WITH ROLLUP	GROUP BY 열들을 오른쪽에서 왼쪽순으로 그룹	GROUP BY (열1, 열2) WITH ROLLUP

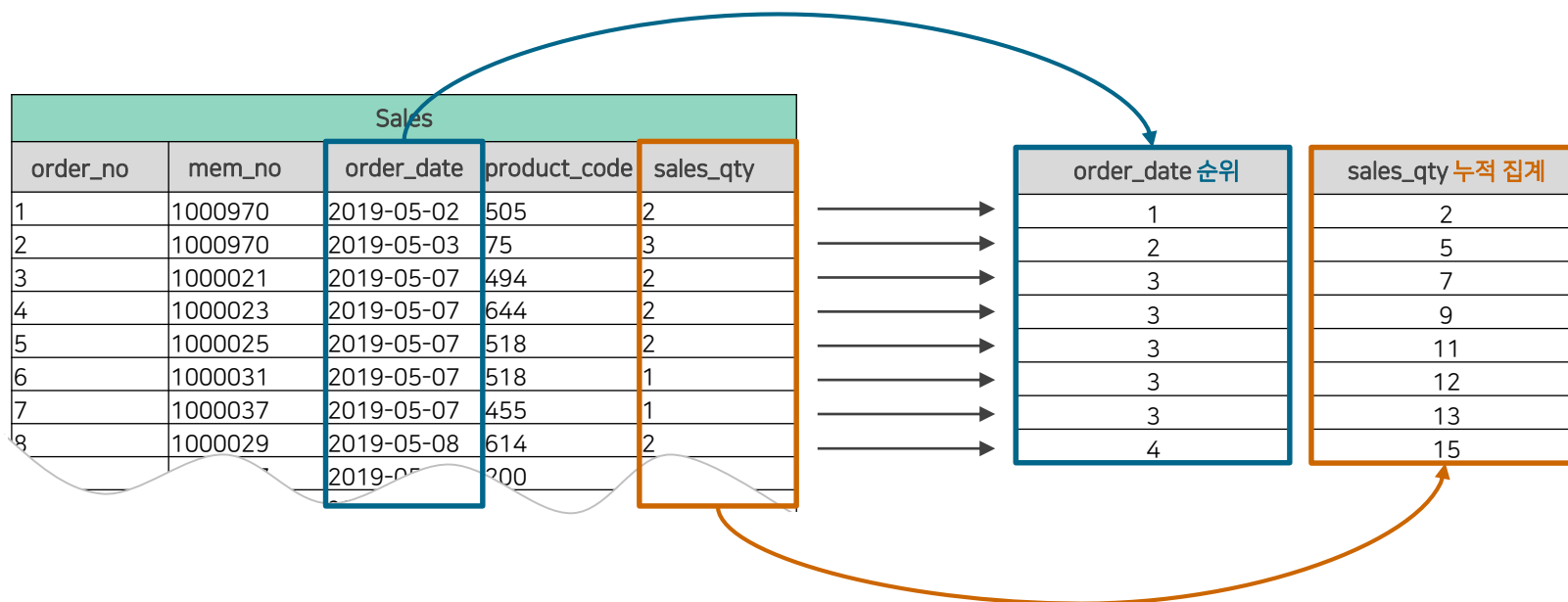
- 복수 행 함수는 여러 행들이 **하나의 결과값**으로 반환합니다.
- 주로 **GROUP BY**절과 함께 사용됩니다.

Sales				
order_no	mem_no	order_date	product_code	sales_qty
1	1000970	2019-05-02	505	2
2	1000970	2019-05-03	75	3
3	1000021	2019-05-07	494	2
4	1000023	2019-05-07	644	2
5	1000025	2019-05-07	518	2
6	1000031	2019-05-07	518	1
7	1000037	2019-05-07	455	1
8	1000029	2019-05-08	614	2
		2019-05-08	200	

회원별  
구매수량 반환

mem_no	SUM(sales_qty)
1000970	5
1000021	29
1000023	15
1000025	25
1000031	27
1000037	8
1000029	15
1000039	10

윈도우 함수는 **행과 행간의 관계**를 정의하여 결과 값을 반환합니다.



윈도우 함수는 순위, 집계(누적) 함수로 나뉩니다.

구분	함수	설명	예
순위 함수	ROW_NUMBER	고유한 순위 반환	1, 2, 3, 4, 5...
	RANK	동일한 값이면 동일한 순위 반환	1, 2, 3, 3, 5...
	DENSE_RANK	동일한 값이면 동일한 순위 반환(+ 하나의 등수로)	1, 2, 3, 3, 4...
집계 함수 (누적)	COUNT	누적 행수	-
	SUM	누적 합계	-
	AVG	누적 평균	-
	MAX / MIN	누적 최대 / 최소	-

윈도우 함수는 **ORDER BY**로 행과 행간의 순서를 정하며, **PARTITION BY**로 그룹화가 가능합니다.

< 윈도우 함수 기본 문법(**ORDER BY**) >

함수

OVER (**ORDER BY** 열 ASC or DESC)

\* ASC: 오름차순 / DESC: 내림차순

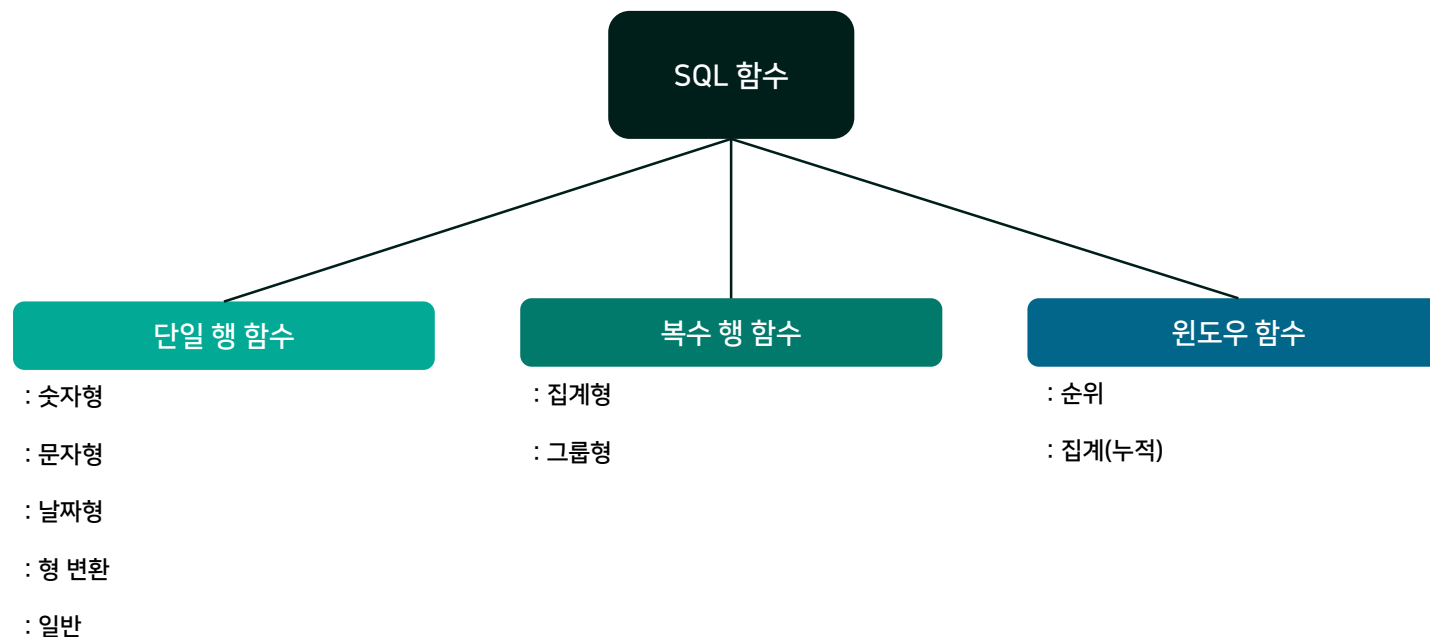
< 윈도우 함수 기본 문법(**ORDER BY**) + **PARTITION BY(그룹화)** >

함수

OVER (**PARTITION BY** 열 OVER BY 열 ASC or DESC)



- 연산자는 **비교, 논리, 특수, 산술, 집합** 연산자로 나뉩니다.
- 함수는 단일 및 복수 행 그리고 윈도우 함수로 나뉘며, **특정 규칙**에 의해 새로운 결과값으로 반환하는 명령어입니다.



# End of Document

-

# Thank you.

