

**CODINGO** x **posco**

**K-Digital Training** 스마트 팩토리 3기

# 데이터 조작어 II

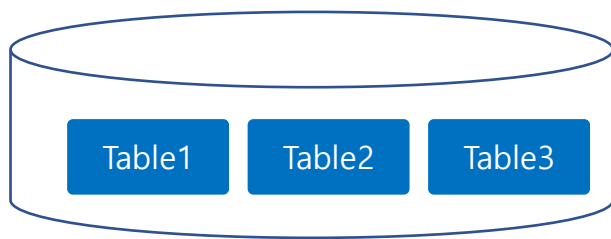
(DML, Data Manipulation Language)



# SELECT 문

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

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



**SELECT**

조회 결과

\*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
	jjjee	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'

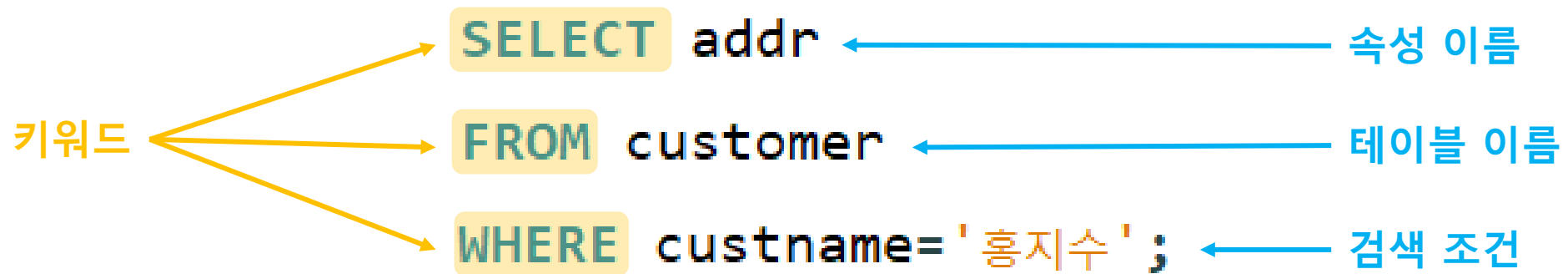
	custid	custname	addr	phone	birth
▶	lalala	홍지수	미국 로스앤젤레스	01010109090	2007-05-16



(3) SELECT addr

	addr
▶	미국 로스앤젤레스

# SELECT 문 구성 요소



# WHERE 조건 – 비교 연산자

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

# WHERE 조건 – 부정 연산자

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

# WHERE 조건 – 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 ) <ul style="list-style-type: none"><li>• % : 0개 이상의 어떤 문자</li><li>• _ : 1개의 단일문자</li></ul>
IS NULL	NULL 값인 경우 true, 아니면 false



# WHERE 조건 – 복합 조건

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.

# 와일드 문자 종류

와일드 문자	의미	예시
%	0개 이상의 문자열과 일치	'%서울%': 서울을 포함하는 문자열
_	특정 위치의 1개의 문자	'_민%': 두번째 글자가 '민'인 문자열

# ORDER BY

- 결과가 출력되는 순서 조절
- where 절과 함께 사용 가능
  - 단, where 절 뒤에 나와야 함
- **ASC**: Ascending, 오름차순 (기본값)
- **DESC**: Descending, 내림차순

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

```
SELECT * FROM user WHERE company='codingon' ORDER BY id ASC;
```

# DISTINCT

- 중복된 데이터 제거

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

```
SELECT DISTINCT gender From user;
```

# LIMIT

- 출력 개수 제한

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

```
SELECT * FROM user LIMIT 2;
```

# 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 그룹을 제한

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

SELECT gender, COUNT(name) as cnt FROM user **GROUP BY** gender;