



# SQL 활용

## 데이터 검색



한국기술교육대학교  
온라인평생교육원

## 학습내용

- 간단한 데이터 검색
- 복잡한 데이터 검색

## 학습목표

- 테이블에 저장된 데이터를 검색할 수 있다.
- 다양한 검색 조건을 지정할 수 있다.

## ● 간단한 데이터 검색

### 1. AS 키워드와 \*

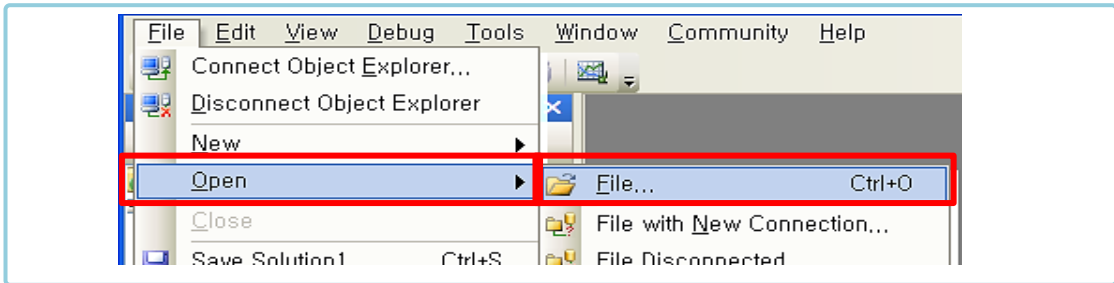
#### ◆ 실습용 데이터 생성

- 실습을 위한 테이블 및 튜플들을 생성함

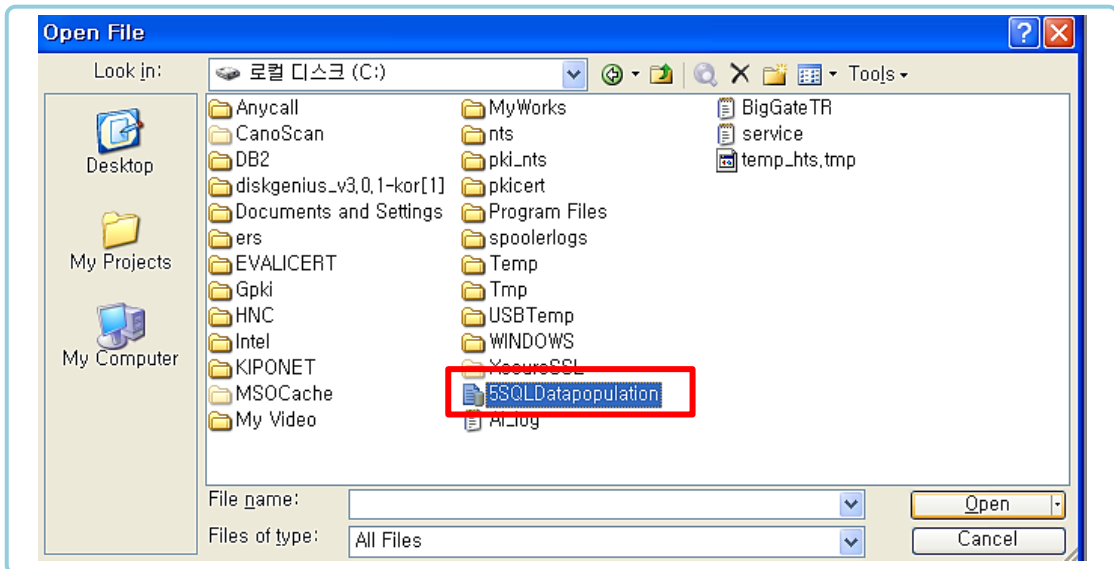
```
5SQLDatapopulate.sql
```

- DB 이름 : MagicCorp
- 테이블 : DEPARTMENT, EMPLOYEE, SALGRADE

#### ① File ⇨ Open ⇨ File...



#### ② 실습용 파일 선택

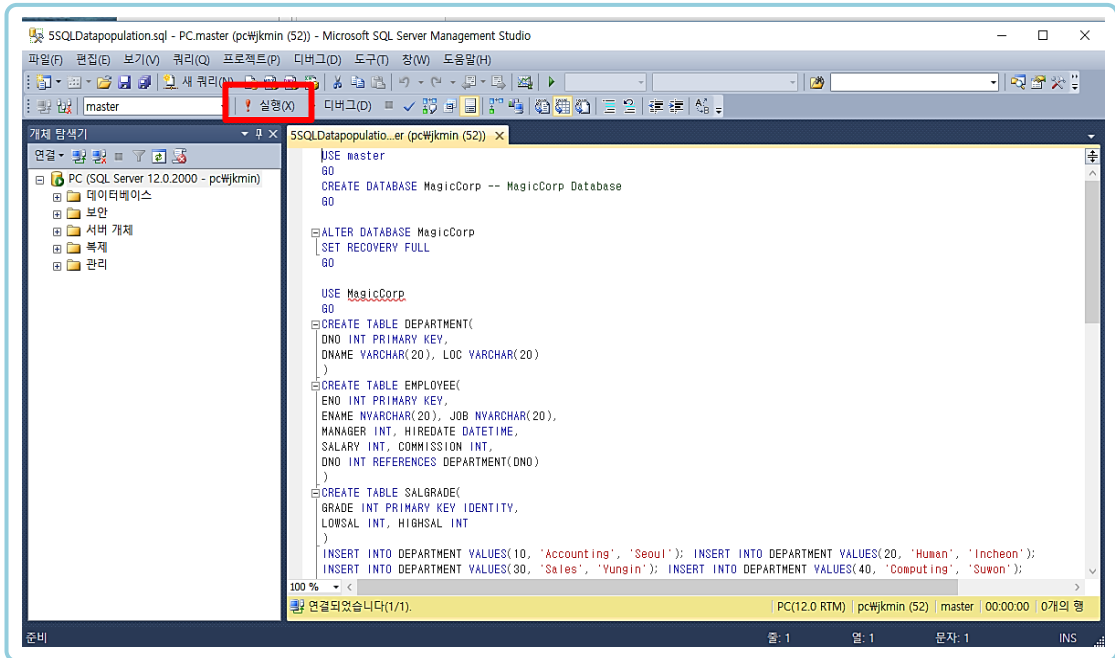


## ● 간단한 데이터 검색

### 1. AS 키워드와 \*

#### ◆ 실습용 데이터 생성

##### ③ 실행



## ● 간단한 데이터 검색

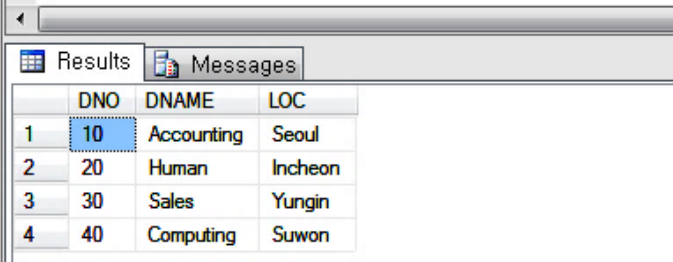
### 1. AS 키워드와 \*

#### ◆ 무조건 검색

```
SELECT 속성명1, 속성명2, ...  
FROM 테이블명
```

#### Q 모든 부서 정보 검색

```
USE MagicCorp  
GO  
  
SELECT DNO, DNAME, LOC FROM DEPARTMENT
```



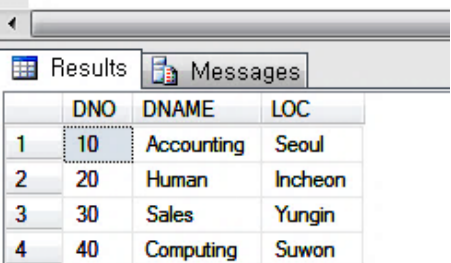
	DNO	DNAME	LOC
1	10	Accounting	Seoul
2	20	Human	Incheon
3	30	Sales	Yungin
4	40	Computing	Suwon

#### ◆ \*

- 모든 속성명을 쓰기 힘들
  - “\*” 를 사용함
  - SELECT절에서 \*는 모든 속성이란 의미임

#### Q 모든 부서 정보 검색

```
07STARSQL.sql - J...ministrator (52))  
USE MagicCorp  
GO  
  
SELECT * FROM DEPARTMENT
```



	DNO	DNAME	LOC
1	10	Accounting	Seoul
2	20	Human	Incheon
3	30	Sales	Yungin
4	40	Computing	Suwon

## ● 간단한 데이터 검색

### 1. AS 키워드와 \*

#### ◆ AS 키워드

- ① 질의 결과의 속성명을 바꾸어서 나타나게 함
- ② 질의결과에 수식을 넣을 수 있음
- ③ 속성명이 없음
- ④ AS 키워드로 속성명을 부여함

Q 사원 테이블에서 사원명과 봉급\*12 검색

#### ■ 속성명이 없음

```
USE MagicCorp
GO

SELECT ENAME, SALARY*12 FROM EMPLOYEE
```

	ENAME	(No column name)
1	e1	3600
2	e2	3000
3	e3	6000
4	e4	7200
5	e5	5400
6	e6	5760
7	e7	6240
8	e8	6000
9	e9	12000
10	e10	6000
11	e11	3360
12	e12	3600
13	e13	6720
14	e14	3000

#### ■ 속성명 생성

```
USE MagicCorp
GO

SELECT ENAME, SALARY*12 AS YEARSAL FROM EMPLOYEE
```

	ENAME	YEARSAL
1	e1	3600
2	e2	3000
3	e3	6000
4	e4	7200
5	e5	5400
6	e6	5760
7	e7	6240
8	e8	6000
9	e9	12000
10	e10	6000
11	e11	3360
12	e12	3600
13	e13	6720
14	e14	3000

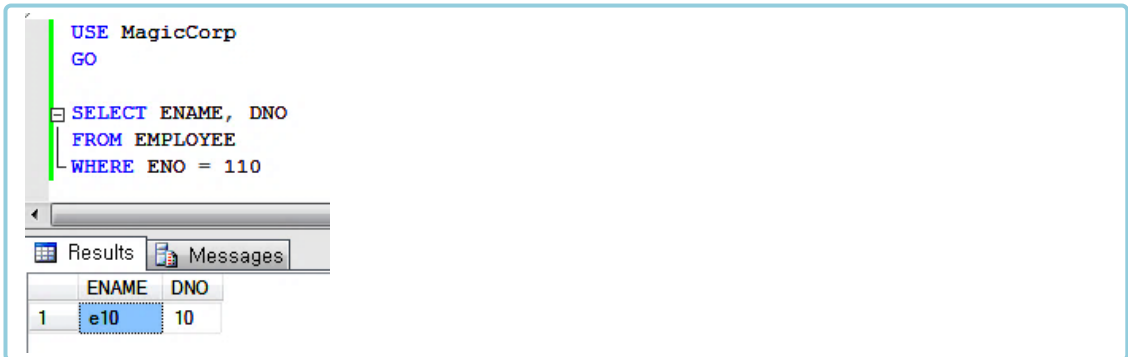
## ● 간단한 데이터 검색

### 2. 간단한 조건 검색

#### ◆ WHERE 절

**Q** 사원 테이블(EMPLOYEE)에서 사원번호(ENO) 110번의 이름(ENAME)과 부서번호(DNO) 검색

- 조건 : 사원번호 110번
- WHERE 절을 사용함



```
USE MagicCorp
GO

SELECT ENAME, DNO
FROM EMPLOYEE
WHERE ENO = 110
```

	ENAME	DNO
1	e10	10

#### ◆ WHERE 절

WHERE 절에서 문자, 숫자, 날짜의 크기나 순서를 비교함

- 같다 : =
- 같지 않다 : !=, <>
- 크다 : >
- 크거나 같다 : >=
- 작다 : <
- 작거나 같다 : <=

#### ◆ 논리 연산자

WHERE 절에서 여러 개의 조건을 결합할 경우

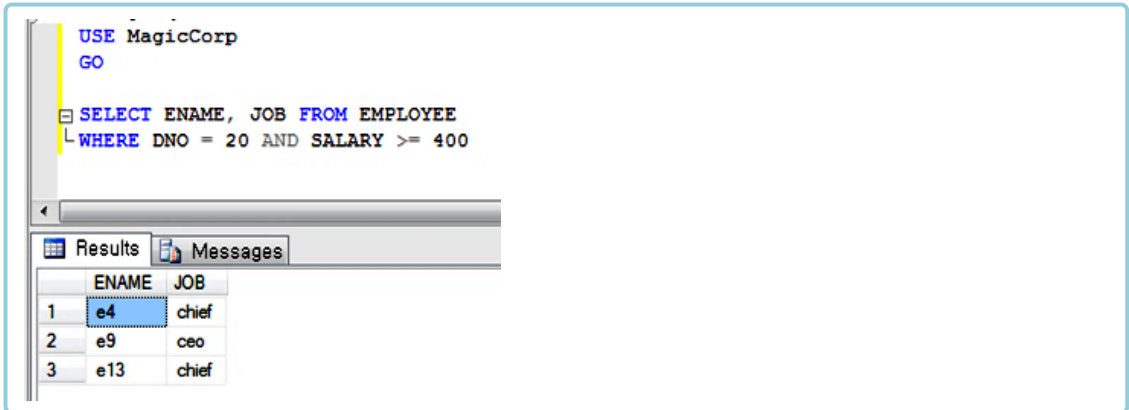
- X AND Y : X, Y가 참일 때 참을 반환
- X OR Y : X나 Y가 참일 때 참을 반환
- NOT X : X가 거짓일 때 참을 반환

## ● 간단한 데이터 검색

### 2. 간단한 조건 검색

- Q 사원 테이블에서 부서번호(DNO)가 20번이고 봉급(SALARY)이 400이상인 사원의 이름(ENAME)과 직책(JOB) 검색

▪ 조건 : DNO = 20 AND SALARY >= 400



The screenshot shows a SQL Developer window with the following SQL query:

```
USE MagicCorp
GO

SELECT ENAME, JOB FROM EMPLOYEE
WHERE DNO = 20 AND SALARY >= 400
```

Below the query, the 'Results' tab is active, displaying a table with 3 rows and 2 columns (ENAME, JOB):

	ENAME	JOB
1	e4	chief
2	e9	ceo
3	e13	chief

### ◆ DISTINCT

- SQL은 Bag 을 기반으로 함
  - 중복된 것들도 다 나옴
- 중복된 것을 제거하고 한번만 나오게 하는 방법
  - DISTINCT를 사용함

예 SELECT DISTINCT 속성명 ...



## ● 간단한 데이터 검색

### 2. 간단한 조건 검색

#### ◆ DISTINCT

- SQL은 Bag 을 기반으로 함
  - 중복된 것들도 다 나옴
- 중복된 것을 제거하고 한번만 나오게 하는 방법
  - DISTINCT를 사용함



SELECT DISTINCT 속성명 ...

#### Q 사원 테이블에서 모든 직급(JOB) 검색

```
USE MagicCorp
GO

SELECT JOB from EMPLOYEE
```

	JOB
1	staff
2	deputy
3	section
4	chief
5	section
6	chief
7	chief
8	senior
9	ceo

#### Q 사원 테이블에서 모든 직급을 중복 없이 검색

```
USE MagicCorp
GO

SELECT DISTINCT JOB from EMPLOYEE
```

	JOB
1	ceo
2	chief
3	deputy
4	section
5	senior
6	staff

## ● 복잡한 데이터 검색

### 1. BETWEEN, IN, IS NULL

#### ◆ BETWEEN a AND b

- 검색 조건의 상한과 하한을 지정함
  - 속성 X가 10보다 크거나 같고 50보다 작거나 같음
  - $X \geq 10$  AND  $X \leq 50$

X BETWEEN 10 AND 50

- Q 사원 테이블에서 봉급이 400보다 크거나 같고 600보다는 작거나 같은  
사원들의 정보 검색

```
USE MagicCorp
GO

SELECT * FROM EMPLOYEE
WHERE SALARY BETWEEN 400 AND 600
```

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	103	e3	section	105	2005-02-10 00:00:00.000	500	100	30
2	104	e4	chief	108	2003-09-02 00:00:00.000	600	NULL	20
3	105	e5	section	105	2005-04-07 00:00:00.000	450	200	30
4	106	e6	chief	108	2003-10-09 00:00:00.000	480	NULL	30
5	107	e7	chief	108	2004-01-08 00:00:00.000	520	NULL	10
6	108	e8	senior	103	2004-03-08 00:00:00.000	500	0	30
7	110	e10	section	103	2005-04-07 00:00:00.000	500	NULL	10
8	113	e13	chief	103	2002-10-09 00:00:00.000	560	NULL	20

## ● 복잡한 데이터 검색

### 1. BETWEEN, IN, IS NULL

#### ◆ IN(a, b, c, ...)

- 속성값이 a, b, c, ... 중 하나라도 일치하면 참
  - 속성 X가 10이거나 20이거나 30임
  - $X = 10 \text{ OR } X = 20 \text{ OR } X = 30$

X IN(10, 20, 30)

Q 직급이 'section'이거나 'senior'인 직원들의 이름과 직급 검색

```
USE MagicCorp
GO

SELECT ENAME, JOB FROM EMPLOYEE
WHERE JOB IN ('section', 'senior')
```

	ENAME	JOB
1	e3	section
2	e5	section
3	e8	senior
4	e10	section

## ● 복잡한 데이터 검색

### 1. BETWEEN, IN, IS NULL

#### ◆ IS NULL

- NULL 값은 어떤 비교를 하든 거짓임
  - 사원 테이블에서 COMMISSION 값이 NULL인 튜플들이 있을 경우

```
USE MagicCorp
GO

SELECT * FROM EMPLOYEE
```

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	101	e1	staff	113	2007-03-01 00:00:00.000	300	NULL	20
2	102	e2	deputy	105	2007-04-02 00:00:00.000	250	80	30
3	103	e3	section	105	2005-02-10 00:00:00.000	500	100	30
4	104	e4	chief	108	2003-09-02 00:00:00.000	600	NULL	20
5	105	e5	section	105	2005-04-07 00:00:00.000	450	200	30
6	106	e6	chief	108	2003-10-09 00:00:00.000	480	NULL	30

## ● 복잡한 데이터 검색

### 1. BETWEEN, IN, IS NULL

#### ◆ IS NULL

- NULL 값은 어떤 비교를 하든 거짓임
  - 사원 테이블에서 COMMISSION 값이 NULL인 튜플들이 있을 경우
    - ①  $X = \text{NULL} \Rightarrow X$ 가 NULL 값이어도 이 결과는 거짓
    - ② 결과가 없음

```
USE MagicCorp
GO

SELECT * FROM EMPLOYEE
WHERE COMMISSION = NULL
```

Results Messages

ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
-----	-------	-----	---------	----------	--------	------------	-----

⇒ IS NULL을 이용함

```
USE MagicCorp
GO

SELECT * FROM EMPLOYEE
WHERE COMMISSION IS NULL
```

Results Messages

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	101	e1	staff	113	2007-03-01 00:00:00.000	300	NULL	20
2	104	e4	chief	108	2003-09-02 00:00:00.000	600	NULL	20
3	106	e6	chief	108	2003-10-09 00:00:00.000	480	NULL	30
4	107	e7	chief	108	2004-01-08 00:00:00.000	520	NULL	10
5	109	e9	ceo	NULL	1996-10-04 00:00:00.000	1000	NULL	20
6	110	e10	section	103	2005-04-07 00:00:00.000	500	NULL	10

## ● 복잡한 데이터 검색

### 1. BETWEEN, IN, IS NULL

#### ◆ IS NULL

- NULL 값은 어떤 비교를 하든 거짓임
  - NULL 값이 아닌 것들을 찾는 방법
    - ⇒ IS NOT NULL을 이용함

```
USE MagicCorp
GO

SELECT * FROM EMPLOYEE
WHERE COMMISSION IS NOT NULL
```

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	102	e2	deputy	105	2007-04-02 00:00:00.000	250	80	30
2	103	e3	section	105	2005-02-10 00:00:00.000	500	100	30
3	105	e5	section	105	2005-04-07 00:00:00.000	450	200	30
4	108	e8	senior	103	2004-03-08 00:00:00.000	500	0	30

## ● 복잡한 데이터 검색

### 2. 문자열 검색

#### ◆ LIKE 연산자

- 컬럼에 저장된 문자열 중에서 LIKE 연산자에서 지정한 문자 패턴과 부분적으로 일치하면 참이 되는 연산자

#### ◆ 부분 문자열 검색에 사용되는 패턴

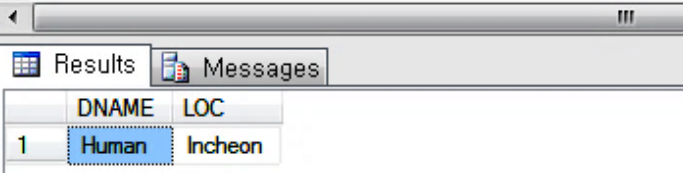
- % : 임의의 길이의 문자열
- \_ : 글자 한자

#### ◆ 부분 문자열 검색에 사용되는 패턴

- Q 부서 테이블에서 부서명(DNAME)이 H로 시작하고 n으로 끝나는 부서의 위치(Loc) 검색

```
USE MagicCorp
GO

SELECT DNAME, LOC from DEPARTMENT
WHERE DNAME LIKE 'H%n'
```



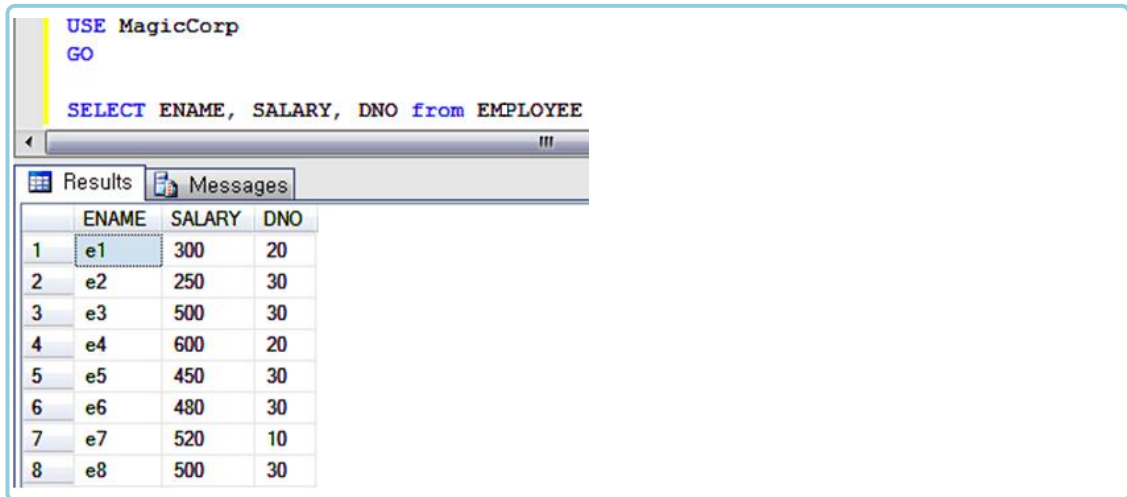
	DNAME	LOC
1	Human	Incheon

## ● 복잡한 데이터 검색

### 2. 문자열 검색

#### ◆ 질의문의 결과는 테이블에 입력된 순서대로 출력됨

- 데이터의 출력 순서를 특정 속성값을 기준으로 오름차순 또는 내림차순으로 정렬해야 하는 경우가 자주 발생함



```
USE MagicCorp
GO

SELECT ENAME, SALARY, DNO from EMPLOYEE
```

	ENAME	SALARY	DNO
1	e1	300	20
2	e2	250	30
3	e3	500	30
4	e4	600	20
5	e5	450	30
6	e6	480	30
7	e7	520	10
8	e8	500	30



## ● 복잡한 데이터 검색

### 3. ORDER BY 절

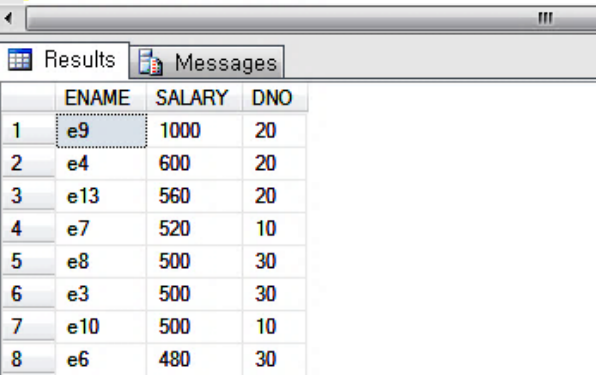
#### ◆ ORDER BY 절

ORDER BY {column\_name} [ASC|DESC]

- ASC : 오름차순으로, 기본값(생략가능)
- DESC : 내림차순, 생략불가능

Q 봉급(Salary) 기준 내림차순으로 직원들의 이름, 봉급, 부서 번호 출력

```
SELECT ENAME, SALARY, DNO from EMPLOYEE
ORDER BY SALARY DESC
```



The screenshot shows a SQL query window with the query: `SELECT ENAME, SALARY, DNO from EMPLOYEE ORDER BY SALARY DESC`. The results are displayed in a table with columns ENAME, SALARY, and DNO. The results are ordered by salary in descending order.

	ENAME	SALARY	DNO
1	e9	1000	20
2	e4	600	20
3	e13	560	20
4	e7	520	10
5	e8	500	30
6	e3	500	30
7	e10	500	10
8	e6	480	30

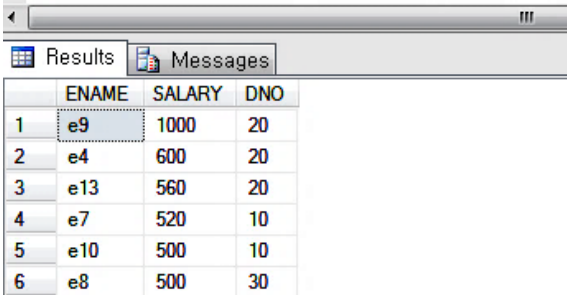
#### ◆ 다중 속성 정렬

Q 봉급(Salary) 기준 내림차순으로 직원들의 이름, 봉급, 부서 번호 출력

- 봉급이 같은 경우에는 부서번호가 낮은 순으로(오름차순) 정렬함

```
USE MagicCorp
GO

SELECT ENAME, SALARY, DNO from EMPLOYEE
ORDER BY SALARY DESC, DNO ASC
```



The screenshot shows a SQL query window with the query: `USE MagicCorp GO SELECT ENAME, SALARY, DNO from EMPLOYEE ORDER BY SALARY DESC, DNO ASC`. The results are displayed in a table with columns ENAME, SALARY, and DNO. The results are ordered by salary in descending order, and then by department number in ascending order for ties.

	ENAME	SALARY	DNO
1	e9	1000	20
2	e4	600	20
3	e13	560	20
4	e7	520	10
5	e10	500	10
6	e8	500	30

# 핵심요약

## 1. 간단한 데이터 검색

### ■ 제약조건의 설정

#### ■ 무조건 검색

```
SELECT 속성명1, 속성명2, ...  
FROM 테이블명
```

#### ■ \*

- 모든 속성명을 쓰기 힘들 경우 사용
- SELECT절에서 \*는 모든 속성이란 의미임

#### ■ AS 키워드

- AS 키워드로 속성명을 부여함

### ■ 간단한 조건 검색

#### ■ WHERE 절

#### ■ 비교 연산자

- 같다 : =
- 같지 않다 : !=, <>
- 크다 : >
- 크거나 같다 : >=
- 작다 : <
- 작거나 같다 : <=

#### ■ 논리 연산자

- WHERE 절에서 여러 개의 조건을 결합할 경우
- X AND Y : X, Y가 참일 때 참을 반환
- X OR Y : X나 Y가 참일 때 참을 반환
- NOT X : X가 거짓일 때 참을 반환

## 핵심요약

### 1. 간단한 데이터 검색

#### ■ DISTINCT

- SQL은 Bag을 기반으로 함
  - 중복된 것들도 다 제시 됨
- 중복된 것을 제거하고 한번만 나오게 하려면?
  - DISTINCT를 사용함

# 핵심요약

## 2. 복잡한 데이터 검색

### ■ BETWEEN, IN, IS NULL

- BETWEEN a AND b
  - 검색 조건의 상한과 하한을 지정함
- IN(a, b, c, ...)
  - 속성값이 a, b, c, ... 중 하나라도 일치하면 참
- IS NULL
  - NULL 값은 어떤 비교를 하든 거짓임

### ■ 문자열 검색

- LIKE 연산자
  - 컬럼에 저장된 문자열 중에서 LIKE 연산자에서 지정한 문자 패턴과 부분적으로 일치하면 참이 되는 연산자
- ORDER BY 절
  - 질의문의 결과는 테이블에 입력된 순서대로 출력
  - 데이터의 출력 순서를 특정 속성값을 기준으로 오름차순 또는 내림차순으로 정렬해야 하는 경우가 자주 발생함

ORDER BY {column\_name} [ASC|DESC]

- ▶ ASC : 오름차순으로, 기본값 (생략가능)
- ▶ DESC : 내림차순, 생략불가능