



데이터베이스

데이터 검색





학습목표

- ➔ SELECT절과 WHERE절을 활용하여 저장된 데이터를 검색할 수 있다.
- ➔ 다양한 연산자를 활용하여 복잡한 데이터를 검색할 수 있다.



학습내용

- ➔ SELECT절과 WHERE절
- ➔ 복잡한 데이터 검색



SELECT절과 WHERE절



SELECT절

01 무조건 검색

- | SELECT: 속성명1, 속성명2, ... 입력
- | FROM: 테이블명 입력
- » 예 | 모든 부서 정보를 검색하시오.

```
USE MagicCorp  
GO
```

```
SELECT DNO, DNAME, LOC FROM DEPARTMENT
```

Results

Messages

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



SELECT절과 WHERE절



SELECT절

02

*의 사용

- 모든 속성명을 입력하기 힘들 때, '*' 사용
- SELECT절에서 '*'는 모든 속성을 의미
- 예 | 모든 부서 정보를 검색하시오.

07STARSQL.sql - J...ministrator (52))

USE MagicCorp
GO

SELECT * FROM DEPARTMENT

Results Messages

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



SELECT절과 WHERE절



SELECT절

03 AS 키워드

질의 결과의 속성명을 바꾸어서 나타낼 때 사용

예 | 사원 테이블에서 사원명과 봉급*12를 검색하시오.

```
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

No column name

속성명이 없는 경우



```
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

AS YEARSAL 입력

입력된 속성명(YEARSAL) 반영

AS 키워드로 속성명을 부여



SELECT절과 WHERE절



SELECT절



04 DISTINCT

질문의 결과 중 **중복된 것을 한 번만 나오게 할 때** 사용

- 사용 예: SELECT DISTINCT 속성명 ...



질문의 결과 중 **중복된 것이 나오는 이유는?**

- SQL(Structured Query Language)이 **Bag**을 기반으로 하기 때문



SELECT절과 WHERE절



SELECT절

04

DISTINCT

질의 결과 중 중복된 것을 한 번만 나오게 할 때 사용

예 | 사원 테이블에서 모든 직급을 중복 없이 검색하시오.

```
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
10	section
11	staff
12	staff
13	chief
14	staff

모든 직급(Job)
검색



```
ute USE MagicCorp
GO

SELECT DISTINCT JOB from EMPLOYEE
```

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

SELECT DISTINCT JOB 입력



SELECT절과 WHERE절



WHERE절



01 WHERE절

예 | 사원 테이블(EMPLOYEE)에서 사원번호(ENO) 110번의 이름(ENAME)과 부서번호(DNO)를 검색하시오.

조건

WHERE절을 사용하여, 사원번호 110번 검색

```
USE MagicCorp
GO

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

	ENAME	DNO
1	e10	10

SELECT ENAME, DNO
FROM EMPLOYEE
WHERE ENO = 110
입력 후 검색



SELECT절과 WHERE절



WHERE절

02 비교 연산자

WHERE절에서 문자, 숫자, 날짜의 크기나 순서를 비교할 때 사용

같다

=

같지 않다

!=, <>

크다

>

크거나 같다

>=

작다

<

작거나 같다

<=

03 논리 연산자

WHERE절에서 여러 개의 조건을 결합할 때 사용

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



SELECT절과 WHERE절



WHERE절

04 비교 연산자와 논리 연산자의 예

예 | 사원 테이블에서 부서번호(DNO)가 20번이고 봉급(SALARY)이 400 이상인 사원의 이름(ENAME)과 직책(JOB)을 검색하시오.

조건 DNO = 20 AND SALARY >= 400

WHERE DNO = 20
AND SALARY >= 400
입력 후 검색

```
USE MagicCorp
GO

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

	ENAME	JOB
1	e4	chief
2	e9	ceo
3	e13	chief



SELECT절과 WHERE절



WHERE절

05 다양한 비교 연산자들

| BETWEEN a AND b

- 검색 조건의 상한과 하한을 지정

≫ 예 | 속성 X가 10보다 크거나 같고 50보다 작거나 같다.

- $X \geq 10 \text{ AND } X \leq 50$
- $X \text{ BETWEEN } 10 \text{ AND } 50$





SELECT절과 WHERE절



WHERE절

05

다양한 비교 연산자들

■ BETWEEN a AND b

» 예 | 사원 테이블에서 봉급이 400보다 크거나 같고 600보다는 작거나 같은 사원들의 정보를 검색하시오.

```
USE MagicCorp
GO

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

WHERE SALARY
BETWEEN 400 AND 600
입력 후 검색

	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



SELECT절과 WHERE절



WHERE절

05

다양한 비교 연산자들

IN(a, b, c, ...)

- 속성값이 a, b, c, ... 중 하나라도 일치하면 참

예 | 속성 X가 10이거나 20이거나 30이다.

- X = 10 OR X = 20 OR X = 30
- X IN(10, 20, 30)

예 | 직급이 'section'이거나 'senior'인 직원들의 이름과 직급을 검색하시오.

```
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

WHERE JON IN('section', 'senior')
입력 후 검색



SELECT절과 WHERE절



WHERE절

05

다양한 비교 연산자들

IS NULL

- 사원 테이블에서 **COMMISSION** 값이 NULL인 튜플들이 있는 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
7	107	e7	chief	108	2004-01-08 00:00:00.000	520	NULL	10
8	108	e8	senior	103	2004-03-08 00:00:00.000	500	0	30
9	109	e9	ceo	NULL	1996-10-04 00:00:00.000	1000	NULL	20
10	110	e10	section	103	2005-04-07 00:00:00.000	500	NULL	10
11	111	e11	staff	107	2007-03-01 00:00:00.000	280	NULL	30
12	112	e12	staff	106	2007-08-09 00:00:00.000	300	NULL	20
13	113	e13	chief	103	2002-10-09 00:00:00.000	560	NULL	20
14	114	e14	staff	106	2007-11-09 00:00:00.000	250	NULL	10



SELECT절과 WHERE절



WHERE절

05

다양한 비교 연산자들

IS NULL

예 | 사원 테이블에서 COMMISSION 값이 NULL인 튜플들을 검색하시오.

```

USE MagicCorp
GO

SELECT * FROM EMPLOYEE
WHERE COMMISSION = NULL

```

WHERE COMMISSION = NULL

Results

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

- X = NULL
→ NULL값은 어떤 비교를 해도 거짓

IS NULL을 이용



SELECT절과 WHERE절



WHERE절

05

다양한 비교 연산자들

IS NULL

예 | 사원 테이블에서 COMMISSION 값이 NULL인 튜플들을 검색하시오.

```
USE MagicCorp
GO

SELECT * FROM EMPLOYEE
WHERE COMMISSION IS NULL
```

WHERE COMMISSION IS NULL 입력 후 검색

	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
7	111	e11	staff	107	2007-03-01 00:00:00.000	280	NULL	30
8	112	e12	staff	106	2007-08-09 00:00:00.000	300	NULL	20
9	113	e13	chief	103	2002-10-09 00:00:00.000	560	NULL	20
10	114	e14	staff	106	2007-11-09 00:00:00.000	250	NULL	10



SELECT절과 WHERE절



WHERE절



05 다양한 비교 연산자들

IS NOT NULL

예 | 사원 테이블에서 COMMISSION 값이 NULL이 아닌 튜플들을 검색하시오.

```
USE MagicCorp
GO

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

WHERE COMMISSION IS NOT NULL 입력 후 검색

	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



SELECT절과 WHERE절



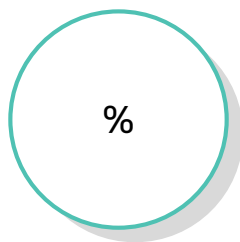
WHERE절

05 다양한 비교 연산자들

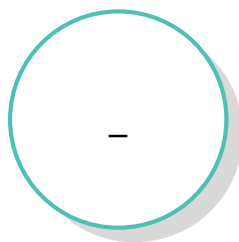
| LIKE

- 컬럼에 저장된 문자열 중 LIKE 연산자에서 **지정한 문자 패턴과 부분적으로 일치하면 참**이 되는 연산자

[부분 문자열 검색에 사용되는 패턴]



임의의 길이로
이루어진 문자열



글자, 한자



SELECT절과 WHERE절



WHERE절

05

다양한 비교 연산자들

LIKE

예 | 부서 테이블에서 부서명(DNAME)이 H로 시작하고 n으로 끝나는 부서의 위치(LOC)를 검색하시오.

```
USE MagicCorp
GO

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

	DNAME	LOC
1	Human	Incheon

WHERE DNAME LIKE 'H%n'
입력 후 검색



SELECT절과 WHERE절



집합 연산자



01 집합 연산자

[테이블을 구성하는 튜플 집합에 대해
테이블의 **부분 집합을 결과로 반환**하는 연산자]





SELECT절과 WHERE절



집합 연산자



02 집합 연산자의 종류

UNION

예 | 부서번호가 10인 직원들과 직급이 staff인 직원들을 검색하시오.

```

USE MagicCorp
GO

(select *
FROM EMPLOYEE
WHERE DNO = 10)
UNION
(select *
FROM EMPLOYEE
WHERE JOB = 'staff')

```

UNION
(select *
FROM EMPLOYEE
WHERE JOB = 'staff') 입력 후 검색

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	101	e1	staff	113	2007-03-01 00:00:00.000	300	NULL	20
2	107	e7	chief	108	2004-01-08 00:00:00.000	520	NULL	10
3	110	e10	section	103	2005-04-07 00:00:00.000	500	NULL	10
4	111	e11	staff	107	2007-03-01 00:00:00.000	280	NULL	30
5	112	e12	staff	106	2007-08-09 00:00:00.000	300	NULL	20
6	114	e14	staff	106	2007-11-09 00:00:00.000	250	NULL	10



SELECT절과 WHERE절



집합 연산자



02 집합 연산자의 종류

UNION과 UNION ALL

UNION

VS

UNION ALL



중복된 결과를 제거하고,
하나의 결과로 출력



중복된 결과까지
모두 출력



SELECT절과 WHERE절



집합 연산자



02 집합 연산자의 종류

UNION ALL

예 | 부서번호 10인 직원들과 직급이 staff인 직원들을 검색하시오.
(중복 허용)

```
USE MagicCorp
GO

(select *
FROM EMPLOYEE
WHERE DNO = 10)
UNION ALL
(select *
FROM EMPLOYEE
WHERE JOB = 'staff')
```

UNION ALL
(select *
FROM EMPLOYEE
WHERE JOB = 'staff') 입력 후 검색

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	107	e7	chief	108	2004-01-08 00:00:00.000	520	NULL	10
2	110	e10	section	103	2005-04-07 00:00:00.000	500	NULL	10
3	114	e14	staff	106	2007-11-09 00:00:00.000	250	NULL	10
4	101	e1	staff	113	2007-03-01 00:00:00.000	300	NULL	20
5	111	e11	staff	107	2007-03-01 00:00:00.000	280	NULL	30
6	112	e12	staff	106	2007-08-09 00:00:00.000	300	NULL	20
7	114	e14	staff	106	2007-11-09 00:00:00.000	250	NULL	10



SELECT절과 WHERE절



집합 연산자



02 집합 연산자의 종류

INTERSECT

예 | 부서번호가 10이고 직급이 staff인 직원들을 검색하시오.

```

USE MagicCorp
GO

(select *
FROM EMPLOYEE
WHERE DNO = 10)
INTERSECT
(select *
FROM EMPLOYEE
WHERE JOB = 'staff')

```

INTERSECT
(select *
FROM EMPLOYEE
WHERE JOB = 'staff') 입력 후 검색

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	114	e14	staff	106	2007-11-09 00:00:00.000	250	NULL	10



SELECT절과 WHERE절



집합 연산자



02 집합 연산자의 종류

| EXCEPT

» 예 | 부서번호가 10인 사원들 중 직급이 staff가 아닌 사원들을 검색하시오.

```
USE MagicCorp
GO

(select *
FROM EMPLOYEE
WHERE DNO = 10)
EXCEPT
(select *
FROM EMPLOYEE
WHERE JOB = 'staff')
```

EXCEPT
(select *
FROM EMPLOYEE
WHERE JOB = 'staff') 입력 후 검색

100 %

결과 메시지

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	107	e7	chief	108	2004-01-08 00:00:00,000	520	NULL	10
2	110	e10	section	103	2005-04-07 00:00:00,000	500	NULL	10



SELECT절과 WHERE절



집합 연산자

02 집합 연산자의 종류

외부 합집합

● 합병 호환성

[U, \cap - 연산의 **피연산자(릴레이션)** 들이 지켜야 할 제약 조건]



● **차수**(Degree: 속성의 수)가 같아야 함

● **대응되는 속성의 쌍 별로 타입**(Domain)이 같아야 함

● **대응되는 속성의 쌍 별로 의미**(Semantic)가 같아야 함

● 합병 호환성의 불일치

Q

두 질의 결과가 합병 호환성을 만족하지 않을 때, 합집합을 구하려면?

- 외부 합집합($U+$)을 활용



SELECT절과 WHERE절



집합 연산자

02 집합 연산자의 종류

외부 합집합

- SQL(Structured Query Language)에서의 외부 합집합

두 질의 결과의 속성 수와
타입을 일치



모든 속성에서 사용할 수 있는
속성값 'NULL' 활용



강제로 동일한 구조를 생성

- SQL(Structured Query Language)에서의 외부 합집합

예 |

```
(select eno, ename, dno, NULL from EMPLOYEE)
UNION
(select NULL, NULL, dno, dname from DEPARTMENT)
```

공통속성

	eno	ename	dno	(No column name)
1	101	e1	20	NULL
2	102	e2	30	NULL
3	103	e3	30	NULL
4	104	e4	20	NULL
5	105	e5	30	NULL
6	106	e6	30	NULL
7	107	e7	10	NULL
8	108	e8	30	NULL
9	109	e9	20	NULL
10	110	e10	10	NULL
11	111	e11	30	NULL
12	112	e12	20	NULL
13	113	e13	20	NULL
14	114	e14	10	NULL
15	NULL	NULL	10	Accounting
16	NULL	NULL	20	Human
17	NULL	NULL	30	Sales
18	NULL	NULL	40	Computing

- 두 질의 결과에서 공통인 속성은 dno
- 다른 것들은 NULL로 출력
 - 속성의 수가 각각 4개, 타입도 일치
 - 합집합 가능



데이터 검색



복잡한 데이터 검색



ORDER BY절

질의문의 결과는 테이블에 입력된 순서대로 출력

```
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
9	e9	1000	20
10	e10	500	10
11	e11	280	30
12	e12	300	20
13	e13	560	20
14	e14	250	10



복잡한 데이터 검색



ORDER BY절

01

ORDER BY절의 개념

데이터의 출력 순서를 특정 속성값을 기준으로
오름차순 또는 내림차순으로 정렬할 때 사용

ORDER BY {column_name} [ASC|DESC]

ASC

오름차순으로, 기본 값(생략 가능)

DESC

내림차순, 생략 불가능



ORDER BY절

02

ORDER BY절의 활용

» 예 | 봉급(SALARY) 기준 내림차순으로 직원들의 이름, 봉급, 부서번호를 출력하시오.

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

	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
9	e5	450	30
10	e1	300	20
11	e12	300	20
12	e11	280	30
13	e14	250	10
14	e2	250	30

ORDER BY SALARY DESC
입력 후 출력



ORDER BY절

02

ORDER BY절의 활용

다중 속성 정렬

» 예 | 봉급(SALARY) 기준 내림차순으로 직원들의 이름, 봉급, 부서번호를 출력하시오. 단, 봉급이 같은 경우 부서번호가 낮은 순으로 (오름차순)으로 정렬하시오.

```
USE MagicCorp
GO

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

	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
7	e3	500	30
8	e6	480	30
9	e5	450	30
10	e1	300	20
11	e12	300	20
12	e11	280	30
13	e14	250	10
14	e2	250	30

ORDER BY SALARY DESC,
DNO ASC 입력 후 출력



집단 함수와 GROUP BY절

01

집단 함수

[테이블 전체 행을 **하나 이상의 컬럼** 기준으로 **그룹화**하여, 그 **그룹별 통계값**을 출력하는 함수]

예 | 전체 직원의 평균 봉급을 구하시오.

SELECT AVG(SALARY)
FROM EMPLOYEE
입력 후 출력

```
USE MagicCorp
GO

SELECT AVG(SALARY)
FROM EMPLOYEE
```

```
USE MagicCorp
GO

SELECT AVG(SALARY)
FROM EMPLOYEE
```

Results

Messages

(No column name)	
1	463

종류

SUM	그룹의 합계	AVG	그룹의 평균
COUNT	그룹의 개수	MAX	그룹의 최대값
MIN	그룹의 최소값	STDEV	그룹의 표준편차
VAR	그룹의 분산		

일반적으로 집단 함수는 NULL값을 제외한 속성값들의 통계값을 반환



집단 함수와 GROUP BY절



01 집단 함수

예 | 사원의 최대 봉급, 최소 봉급을 구하시오.

```
USE MagicCorp
GO

SELECT MAX(SALARY) AS MAXSAL, MIN(SALARY) AS MINSAL
FROM EMPLOYEE
```

SELECT MAX(SALARY) AS MAXSAL,
MIN(SALARY) AS MINSAL
FROM EMPLOYEE 입력 후 출력

Results

Messages

	MAXSAL	MINSAL
1	1000	250



집단 함수와 GROUP BY절

01 집단 함수

1 COUNT(*)와 COUNT(속성명)

01 COUNT(*)

- ▶ 테이블에서 조건을 만족하는 행의 개수를 반환하는 함수
- ▶ * 는 모든 속성들이란 의미

02 COUNT(속성명)

- ▶ NULL이 아닌 속성값의 개수

03 COUNT(DISTINCT 속성명)

- ▶ 속성값이 NULL이 아니며, 중복되지 않는 속성값들의 개수
 - SUM, AVG도 DISTINCT를 사용
 - MIN, MAX에서 DISTINCT는 무의미



집단 함수와 GROUP BY절



01 집단 함수

■ COUNT(*)와 COUNT(속성명)

» 예 | 사원 테이블의 튜플 수와 COMMISSION의 개수를 구하시오.

```
USE MagicCorp
GO

SELECT COUNT(*) AS NUMROW, COUNT(COMMISSION)
FROM EMPLOYEE
```

SELECT COUNT(*) AS NUMROW,
COUNT(COMMISSION)
FROM EMPLOYEE 입력 후 출력

Results

Messages

	NUMROW	(No column name)
1	14	4



복잡한 데이터 검색



집단 함수와 GROUP BY절

01 집단 함수

COUNT(*)와 COUNT(속성명)

예 | 사원의 직급(JOB) 수와 중복되지 않는 직급(JOB)의 수를 구하시오.

```
USE MagicCorp
GO

SELECT COUNT(JOB) AS NUMJOB, COUNT(DISTINCT JOB) AS NUMDISTINCTJOB
FROM EMPLOYEE
```

SELECT COUNT(JOB) AS NUMJOB,
COUNT(DISTINCT JOB) AS NUMDISTINCTJOB
FROM EMPLOYEE 입력 후 출력

Results

Messages

	NUMJOB	NUMDISTINCTJOB
1	14	6



복잡한 데이터 검색



집단 함수와 GROUP BY절

02

GROUP BY절

특정 속성을 기준으로 테이블 전체를 그룹으로 나누기 위한 절

예 | 부서별 사원의 평균 봉급

구문

SELECT 컬럼_리스트

FROM 테이블명

WHERE 조건

GROUP BY 컬럼_리스트



집단 함수와 GROUP BY절

02 GROUP BY절

예 | 부서별 직원들의 평균 봉급과 부서번호를 검색하시오.

```
USE MagicCorp
GO

SELECT DNO, AVG(SALARY) AS DNOAVGSAL
FROM EMPLOYEE
GROUP BY DNO
```

SELECT DNO,
AVG(SALARY) DNOAVGSAL,
FROM EMPLOYEE
GROUP BY DNO 입력 후 출력

Results

Messages

	DNO	DNOAVGSAL
1	10	423
2	20	552
3	30	410



복잡한 데이터 검색



집단 함수와 GROUP BY절

02 GROUP BY절

GROUP BY 사용 시 주의점

- 1 SELECT절에는 집단 연산자나 **GROUP BY에 사용한 속성명만 사용 가능**
- 2 공통되는 속성값으로 그룹핑을 했으므로, **각 그룹에서 개별 튜플로 접근 불가**

» 예 | 부서별 직원들의 평균 봉급과 부서번호, 직원이름을 검색하시오.(오류)

```
SELECT DNO, AVG(SALARY) AS DNOAVGSAL, ENAME
FROM EMPLOYEE
GROUP BY DNO 입력 후 출력
```

```
USE MagicCorp
GO
```

```
SELECT DNO, AVG(SALARY) AS DNOAVGSAL, ENAME
FROM EMPLOYEE
GROUP BY DNO
```

Messages

```
Msg 8120, Level 16, State 1, Line 2
Column 'EMPLOYEE.ENAME' is invalid in the select list because it is not contained in either an a
```



집단 함수와 GROUP BY절

02

GROUP BY절

HAVING

- HAVING절은 GROUP BY절의 **종속절**
→ GROUP BY 없이 HAVING은 나타날 수 없음

HAVING절



각 그룹에 대한 제약 조건을
기술할 때 사용

VS

WHERE절



테이블 전체에 대한 제약 조건을
기술할 때 사용



집단 함수와 GROUP BY절

02 GROUP BY절

HAVING

예 | 부서의 최대 봉급이 500 초과인 부서에 대해서만 부서별 직원들의 평균 봉급과 부서번호를 출력하시오.

```
USE MagicCorp  
GO
```

```
SELECT DNO, AVG(SALARY) AS DNOAVGSAL  
FROM EMPLOYEE  
GROUP BY DNO  
HAVING MAX(SALARY) > 500
```

SELECT DNO, AVG(SALARY) AS DNOAVGSAL
FROM EMPLOYEE
GROUP BY DNO
HAVING MAX(SAKARY) > 500 입력 후 출력



집단 함수와 GROUP BY절

02

GROUP BY절

다중 속성 GROUP BY

- 하나 이상의 속성들을 이용하여 그룹을 나누고, 그룹별로 다시 서브 그룹을 나누고자 할 때 사용

GROUP BY 컬럼1, 컬럼2, ..., 컬럼n

예 | 부서별 · 직급별
 직원들의 봉급의
 합을 구하시오.

SELECT dno, job,
 sum(salary) as [AVGSAL]
 FROM employee
 GROUP By dno,
 job 입력 후 출력

```
USE MagicCorp
GO
```

```
select dno, job, sum(salary) as [AVGSAL]
from employee
group by dno, job
```

Results Messages

	dno	job	AVGSAL
1	20	ceo	1000
2	10	chief	520
3	20	chief	1160
4	30	chief	480
5	30	deputy	250
6	10	section	500
7	30	section	950
8	30	senior	500
9	10	staff	250
10	20	staff	600
11	30	staff	280



순위 함수

01 TOP() 함수

질의 결과 튜플 수의 제한

- 질의 결과는 **ORDER BY** 절을 이용하여 정렬
- ORDER BY** 정렬 기준에서 **특정 등수 · 비율**까지만 보고 싶을 때 **TOP(n)** 함수를 사용

SELECT TOP(n) 속성명

...

ORDER BY 속성명

예 | 사원들 중 급여 기준 5등까지만 결과로 출력하시오.

```
USE MagicCorp
GO
select top(5) *
from employee
order by salary desc
```

SELECT top(5) *
FROM employee
order by salary desc 입력 후 출력

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	109	e9	ceo	NULL	1996-10-04 00:00:00.000	1000	NULL	20
2	104	e4	chief	1008	2003-09-02 00:00:00.000	600	NULL	20
3	113	e13	chief	1003	2002-10-09 00:00:00.000	560	NULL	20
4	107	e7	chief	1008	2004-01-08 00:00:00.000	520	NULL	10
5	103	e3	section	1005	2005-02-10 00:00:00.000	500	100	30

5등에 동률이 있을 경우 임의의 한 개가 출력



데이터 검색



복잡한 데이터 검색



순위 함수

01 TOP() 함수

Q

질의 결과 중 동률이 있을 경우, 동률 결과를 모두 보고 싶을 때는?

- WITH TIES를 활용

SELECT TOP(n) WITH TIES 속성명
...
ORDER BY 속성명

TOP(n) WITH TIES의 활용

예 | 사원들 중 급여 기준 5등까지만 결과로 출력하시오.

```
USE MagicCorp
GO
select top(5) WITH TIES *
from employee
order by salary desc
```

SELECT top(5) *
WITH TIES * 입력 후 출력

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	109	e9	ceo	NULL	1996-10-04 00:00:00.000	1000	NULL	20
2	104	e4	chief	1008	2003-09-02 00:00:00.000	600	NULL	20
3	113	e13	chief	1003	2002-10-09 00:00:00.000	560	NULL	20
4	107	e7	chief	1008	2004-01-08 00:00:00.000	520	NULL	10
5	108	e8	senior	1003	2004-03-08 00:00:00.000	500	0	30
6	103	e3	section	1005	2005-02-10 00:00:00.000	500	100	30
7	110	e10	section	1003	2005-04-07 00:00:00.000	500	NULL	10

5등에 동률이 있는 경우, 모두 출력



SELECT절과 WHERE절



순위 함수

01 TOP() 함수

TOP(n) PERCENT의 활용

- 정렬 기준에서 **특정 비율**까지만 보고 싶을 때 활용
- 상위 n%까지만 출력
 - ▶ WITH TIES와도 함께 사용이 가능

```
SELECT TOP(n) PERCENT [WITH TIES] 속성명  
...  
ORDER BY 속성명
```



SELECT절과 WHERE절



순위 함수



TOP() 함수

TOP(n) PERCENT의 활용

예 | 사원들 중 급여 기준 5등까지만 결과로 출력하시오.

```
USE MagicCorp
GO

select top(20) PERCENT *
from employee
order by salary desc
```

SELECT top(20) * PERCENT *
from employee
order by salary desc 입력 후 출력

100 %

결과 메시지

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO
1	109	e9	ceo	NULL	1996-10-04 00:00:00,000	1000	NULL	20
2	104	e4	chief	108	2003-09-02 00:00:00,000	600	NULL	20
3	113	e13	chief	103	2002-10-09 00:00:00,000	560	NULL	20



SELECT절과 WHERE절



순위 함수

01 TOP() 함수



TOP() 함수의 한계

결과 수를 제한

등수 확인 불가



Q

각 튜플에 등수 표시가 필요할 때, 사용할 수 있는 함수는?

- RANK 함수



SELECT절과 WHERE절



순위 함수

02 RANK() 함수

정의

- 속성 기준 오름차순(ASC)또는 내림차순(DESC)으로 정렬된 상태에 대하여 등수를 지정할 때 사용

RANK 함수 OVER (ORDER BY 속성명 [ASC|DESC])

특징

- 동렬에 대하여 동일 등수 배정
- 비연속식 등수 배정

예 | 1, 2, 2, 4,...



SELECT절과 WHERE절



순위 함수



RANK() 함수

예 | 사원에 대하여 이름, 급여, 급여에 대한 내림차순 RANK() 값을 출력하시오.

```
USE MagicCorp
GO

select ename, salary, rank() over(order by salary desc) as rank
from employee
```

select ename, salary, rank()
over(order by salary desc) as rank
from employee 입력 후 출력

	ename	salary	rank
1	e9	1000	1
2	e4	600	2
3	e13	560	3
4	e7	520	4
5	e8	500	5
6	e3	500	5
7	e10	500	5
8	e6	480	8
9	e5	450	9



SELECT절과 WHERE절



순위 함수

03

DENSE_RANK() 함수

```
SELECT 속성명, DENSE_RANK () OVER (ORDER BY 속성명)
```

특징

- 동렬에 대하여 동일 등수 배정
- 연속식 등수 배정

예 | 1, 2, 2, 3, ...



SELECT절과 WHERE절



순위 함수



03 DENSE_RANK() 함수

예 | 사원에 대하여 이름, 급여, 급여에 대한 DENSE_RANK() 값을 출력하시오.

```
USE MagicCorp
GO
select ename, salary, dense_rank() over(order by salary desc) as rank
from employee
```

select ename, salary, dense_rank()
over(order by salary desc) as rank
from employee 입력 후 출력

	ename	salary	rank
1	e9	1000	1
2	e4	600	2
3	e13	560	3
4	e7	520	4
5	e8	500	5
6	e3	500	5
7	e10	500	5
8	e6	480	6
9	e5	450	7



SELECT절과 WHERE절



순위 함수

04 ROW_NUMBER() 함수

SELECT 속성명, ROW_NUMBER () OVER (ORDER BY 속성명)

특징

- 동렬에 대하여 **임의** 등수 배정
- **연속식** 등수 배정

예 | 1, 2, 3, 4, ...



SELECT절과 WHERE절



순위 함수

04

ROW_NUMBER() 함수

예 | 사원에 대하여 이름, 급여, 급여에 대한 ROW_NUMBER() 값을 출력하시오.

```
USE MagicCorp
GO

select ename, salary, ROW_NUMBER() over(order by salary desc) as rank
from employee
```

select ename, salary, ROW_NUMBER()
over(order by salary desc) as rank
from employee 입력 후 출력

	ename	salary	rank
1	e9	1000	1
2	e4	600	2
3	e13	560	3
4	e7	520	4
5	e8	500	5
6	e3	500	6
7	e10	500	7
8	e6	480	8
9	e5	450	9



SELECT절과 WHERE절



순위 함수

05 NTILE(n) 함수

I 정의

- 전체 튜플을 num개로 균등 분할하여 **순위를 지정**할 때 사용

예 |

- 결과 튜플이 20개이고 num이 10이면,
1등 2개, 2등 2개, ...,
10등 2개로 등수 지정
- 결과 튜플 수가 num으로 나누어 떨어지지 않으면
1등부터 추가적으로 배정
➡ 결과 튜플이 22개이고 num이 10이면 1등 3개,
2등 3개, 3등 2개, ..., 10등 2개로 등수 지정

- 예 | 사원에 대하여 이름, 급여, 급여에 대한 내림차순으로 5등분 하여
등분순위를 출력하시오.

USE MagicCorp
60

```
select ename, salary, NTILE(5) over(order by salary desc) as rank
from employee
```

select ename, salary, NTILE(5)
over(order by salary desc) as rank
from employee 입력 후 출력

	ename	salary	rank
1	e9	1000	1
2	e4	600	1
3	e13	560	1
4	e7	520	2
5	e8	500	2
6	e3	500	2
7	e10	500	3
8	e6	480	3
9	e5	450	3



SELECT절과 WHERE절



순위 함수

06 그룹별 순위 지정

기존 RANK 함수 문법

- 전체 결과에 대한 속성값을 기준으로 등수를 지정

Q

특정 그룹별 순위 지정은 어떻게 할까?

» 예 | 부서별로 구분해서 각 부서 내 봉급 순위를 구하시오.



PARTITION BY절 활용

PARTITION BY절

- 튜플들을 속성값에 따라 그룹핑
- 각 그룹에 대하여 순위 함수를 적용

- `RANK() over (PARTITION BY dno ORDER BY salary desc)`



DNO별로 분류 후, 분류된 소그룹에서 salary 기준으로 내림차순하고 순위를 나타내시오.



SELECT절과 WHERE절



순위 함수



그룹별 순위 지정

PARTITION BY절

예 | DNO별로 분류하고 각 분류된 소그룹에서 salary기준 내림차순하고, | 순위를 나타내시오.

```
USE MagicCorp
GO
select ename, salary, dno,
       rank() over(partition by dno order by salary desc) as rank_dept
from employee
```

select ename, salary, dno,
rank() over(partition by dno order by salary desc)
as rank_dept
from employee 입력 후 출력

5	e4	600	20	2
6	e13	560	20	3
7	e12	300	20	4
8	e1	300	20	4
9	e8	500	30	1
10	e3	500	30	1
11	e6	480	30	3
12	e5	450	30	4
13	e11	280	30	5
14	e2	250	30	6



SELECT절과 WHERE절



순위 함수

06 그룹별 순위 지정

PARTITION BY절과 WHERE절

- 그룹별 특정 등수의 정보를 보고 싶을 때 WHERE절을 같이 활용

RANK() over (PARTITION BY dno ORDER BY salary desc)
AS 속성명
...
WHERE 속성명 = 등수

예 | 각 부서에서 급여 순위가 2등인 사원의 부서번호, 이름, 급여를 출력하시오.

```
USE MagicCorp
GO

select *
FROM
  (SELECT dno, ename, salary, rank() over(partition by dno order by salary desc) as rank_val
   FROM EMPLOYEE) as temp
WHERE rank_val =2
```

결과				
	dno	ename	salary	rank_val
1	10	e10	500	2
2	20	e4	600	2

Select *
from (select dno, ename, salary,
rank() over(partition by dno
order by salary desc) as rank_val
from employee) as temp 입력 후 출력



SELECT절과 WHERE절



순위 함수

07 그룹별 집단 함수의 적용

- SELECT 집단 함수 ~ GROUP BY~
- PARTITION BY를 이용해서도 그룹별 집단 함수 적용이 가능

SELECT 집단 함수() OVER (PARTITION BY 속성명)

예 | PARTITION BY를 이용하여 부서별 급여의 평균을 출력하시오.

```
USE MagicCorp
GO

select dno,
       AVG(salary) over(partition by dno ) as avg_sal_dept
from employee
```

Select dno,
AVG(salary)over(partition by dno) as avg_sal_dept
from employee) 입력 후 출력

	dno	avg_sal_dept
1	10	423
2	10	423
3	10	423
4	20	552
5	20	552
6	20	552
7	20	552
8	20	552
9	30	410



1 SELECT절과 WHERE절

✓ SELECT절

- * : 모든 속성을 의미
- AS : 질의 결과의 속성명을 바꾸어 나타나게 함

✓ WHERE절

- 비교 연산자: 문자, 숫자, 날짜의 크기나 순서 비교
- 논리 연산자: 여러 개의 조건을 결합
- BETWEEN: 검색 조건의 상한과 하한 지정
- IN(a, b, c, ...): 속성 값이 a, b, c, ... 중 하나라도 일치하면 참
- IS NULL: 속성 값이 NULL인 튜플을 검색 ↔ IS NOT NULL

2 복잡한 데이터 검색

✓ 집단연산자: 테이블의 전체 행을 하나 이상의 칼럼을 기준으로 그룹화 후, 그룹별 통계 값을 출력

- 집단연산자의 종류: 합계(SUM), 평균(AVG), 최대값(MIN), 최소값(MAX), 표준편차(STDEV)
- GROUP BY와 HAVING
- GROUP BY: 특정 속성을 기준으로 테이블 전체를 그룹으로 나누기 위한 절
- HAVING: 그룹에 대한 제약 조건 기술

✓ 순위 함수

- TOP(n): 제한된 수의 결과만 검색
- RANK: 각 튜플에 등수를 지정