제 3 장
검색 (SELECT)

목차

3.1 SELECT

●데이터를 검색하기 위한 명령어가 SELECT이며, 형식은 다음과 같다.

```
SELECT [DISTINCT] column-commalist
        FROM table-names
        [WHERE predicate]
        [GROUP BY column-commalist   [HAVING predicate]]
        [ORDER BY column-commalist]
```

3.1 SELECT

● 테이블의 구조를 알기 위해 DESCRIBE 명령어를 사용한다.

SQL> describe student;

NAME	NULLABLE	TYPE	DEFAULT	COMMENT
STU_NO	NOT NULL	NUMBER(9)		
STU_NAME		VARCHAR2(12)		
STU_DEPT		VARCHAR2(20)		
STU_GRADE		NUMBER(1)		
STU_CLASS		CHAR(1)		
STU_GENDER		CHAR(1)		
STU_HEIGHT		NUMBER(5,2)		
STU_WEIGHT		NUMBER(5,2)		

3.1 SELECT

● 테이블의 구조를 알기 위해 DESCRIBE 명령어를 사용한다.

SQL> describe enrol;

NAME	NULLABLE	TYPE	DEFAULT	COMMENT
SUB_NO	NOT NULL	CHAR(3)		
STU_NO	NOT NULL	NUMBER(9)		
ENR_GRADE		NUMBER(3)		

SQL> describe subject;

NAME	NULLABLE	TYPE	DEFAULT	COMMENT
SUB_NO	NOT NULL	CHAR(3)		
SUB_NAME		VARCHAR2(30)		
SUB_PROF		VARCHAR2(12)		
SUB_GRADE		NUMBER(1)		
SUB_DEPT		VARCHAR2(20)		

3.1 SELECT

●테이블에 있는 모든 데이터 검색

```
SQL> select *
  2  from student;
```

STU_NO	STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS	STU_GENDER	STU_HEIGHT	STU_WEIGHT
20153075	옥한빛	기계	1	C	M	177	80
20153088	이태연	기계	1	C	F	162	50
20143054	유가인	기계	2	C	F	154	47
20152088	조민우	전기전자	1	C	M	188	90
20142021	심수정	전기전자	2	A	F	168	45
20132003	박희철	전기전자	3	B	M		63
20151062	김인중	컴퓨터정보	1	B	M	166	67
20141007	진현무	컴퓨터정보	2	A	M	174	64
20131001	김종헌	컴퓨터정보	3	C	M		72
20131025	옥성우	컴퓨터정보	3	A	F	172	63

3.1 SELECT

● 특정 열의 내용 검색

```
SQL> select stu_no, stu_name
  2  from student;
```

STU_NO	STU_NAME
20153075	옥한빛
20153088	이태연
20143054	유가인
20152088	조민우
20142021	심수정
20132003	박희철
20151062	김인중
20141007	진현무
20131001	김종헌
20131025	옥성우

3.1 SELECT

●중복 행 제거

```
SQL> select stu_dept
  2  from student;
```

STU_DEPT
기계
기계
기계
전기전자
전기전자
전기전자
컴퓨터정보
컴퓨터정보
컴퓨터정보
컴퓨터정보

```
SQL> select distinct stu_dept
  2  from student;
```

STU_DEPT
전기전자
기계
컴퓨터정보

3.1 SELECT

●중복 행 제거

```
SQL> select distinct stu_grade, stu_class
  2  from student;
```

STU_GRADE	STU_CLASS
1	B
1	C
3	A
2	C
2	A
3	B
3	C

3.1 SELECT

●수식을 포함한 검색

```
SQL> select stu_no, sub_no, enr_grade, enr_grade+10
  2  from enrol;
```

STU_NO	SUB_NO	ENR_GRADE	ENR_GRADE+10
20131001	101	80	90
20131001	104	56	66
20132003	106	72	82
20152088	103	45	55
20131025	101	65	75
20131025	104	65	75
20151062	108	81	91
20143054	107	41	51
20153075	102	66	76
20153075	105	56	66
20153088	102	61	71
20153088	105	78	88

3.1 SELECT

● 결과열에 별칭(Alias) 부여하기

```
SQL> select stu_no as ID, stu_name as name
  2  from student;
```

ID	NAME
20153075	옥한빛
20153088	이태연
20143054	유가인
20152088	조민우
20142021	심수정
20132003	박희철
20151062	김인중
20141007	진현무
20131001	김종헌
20131025	옥성우

3.1 SELECT

●연결 연산자

```
SQL> select stu_dept || stu_name as 학과성명
  2  from student;
```

학과성명
기계옥한빛
기계이태연
기계유가인
전기전자조민우
전기전자심수정
전기전자박희철
컴퓨터정보김인중
컴퓨터정보진현무
컴퓨터정보김종헌
컴퓨터정보옥성우

3.1 SELECT

●연결 연산자

```
SQL> select stu_dept || ',' || stu_name || '입니다' as 학과성명
  2  from student;
```

학과성명
기계,옥한빛입니다
기계,이태연입니다
기계,유가인입니다
전기전자,조민우입니다
전기전자,심수정입니다
전기전자,박희철입니다
컴퓨터정보,김인중입니다
컴퓨터정보,진현무입니다
컴퓨터정보,김종헌입니다
컴퓨터정보,옥성우입니다

3.2 조건 검색

●WHERE절 사용하기

- ➢일반적으로 비교연산자를 사용한다.
- ➢이때 사용되는 비교연산자는 =, <, >, <=, >=, <>의 6가지이다.
- ➢특히 '<>'는 '!=', '^='를 사용할 수도 있다.

```
SQL> select stu_name, stu_dept, stu_grade, stu_class
  2  from student
  3  where stu_dept = '컴퓨터정보';
```

STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS
김인중	컴퓨터정보	1	B
진현무	컴퓨터정보	2	A
김종헌	컴퓨터정보	3	C
옥성우	컴퓨터정보	3	A

3.2 조건 검색

●논리 연산자

➢논리 연산자 NOT, AND, OR를 사용하여 여러 개의 조건을 결합하여 표현할 수 있다.

```
SQL> select stu_name, stu_dept, stu_grade, stu_class
  2  from student
  3  where stu_dept = '컴퓨터정보' and stu_grade = 2;
```

STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS
진현무	컴퓨터정보	2	A

3.2 조건 검색

● 범위조건

➢ WHERE절에 BETWEEN ~ AND을 사용하여 검색할 수 있다.

```
SQL> select *
  2  from student
  3  where stu_weight between 60 and 70;
```

STU_NO	STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS	STU_GENDER	STU_HEIGHT	STU_WEIGHT
20132003	박희철	전기전자	3	B	M		63
20151062	김인중	컴퓨터정보	1	B	M	166	67
20141007	진현무	컴퓨터정보	2	A	M	174	64
20131025	옥성우	컴퓨터정보	3	A	F	172	63

3.2 조건 검색

```
SQL> select *
  2  from student
  3  where stu_no between '20090001' and '20099999';
```

STU_NO	STU_ NAME	STU_ DEPT	STU_ GRADE	STU_ CLASS	STU_ GENDER	STU_ HEIGHT	STU_ WEIGHT
20141007	진현무	컴퓨터정보	2	A	M	174	64
20142021	심수정	전기전자	2	A	F	168	45
20143054	유가인	기계	2	C	F	154	47

3.2 조건 검색

●LIKE를 이용한 검색

➢데이터의 일부를 알고 있을 경우, 와일드카드 문자와 함께 사용한다.

```
SQL> select stu_no, stu_name, stu_dept
  2  from student
  3  where stu_name like '김%';
```

STU_NO	STU_NAME	STU_DEPT
20151062	김인중	컴퓨터정보
20131001	김종헌	컴퓨터정보

●오라클에서 LIKE와 같이 사용하는 와일드카드 문자는 다음과 같다.

기호	의 미
%	0개 이상의 문자
_	1개의 문자

3.2 조건 검색

```
SQL> select stu_no, stu_name, stu_dept
  2  from student
  3  where stu_name like '_수%';
```

STU_NO	STU_NAME	STU_DEPT
20142021	심수정	전기전자

```
SQL> select *
  2  from student
  3  where stu_no like '2009%';
```

STU_NO	STU_ NAME	STU_ DEPT	STU_ GRADE	STU_ CLASS	STU_ GENDER	STU_ HEIGHT	STU_ WEIGHT
20143054	유가인	기계	2	C	F	154	47
20142021	심수정	전기전자	2	A	F	168	45
20141007	진현무	컴퓨터정보	2	A	M	174	64

3.2 조건 검색

●널(NULL) 값 처리

➢ 널 값이 존재할 수 있으며, 이는 때때로 자료처리에 있어서 문제를 발생시킨다.

```
SQL> select stu_no, stu_name, stu_height
  2  from student;
```

STU_NO	STU_NAME	STU_HEIGHT
20153075	옥한빛	177
20153088	이태연	162
20143054	유가인	154
20152088	조민우	188
20142021	심수정	168
20132003	박희철	
20151062	김인중	166
20141007	진현무	174
20131001	김종헌	
20131025	옥성우	172

➢ 신장(stu_height)열에 널 값이 나타남.

3.2 조건 검색

●NULL값을 가지는 연산

```
SQL> select stu_name, stu_height/30.46
  2  from student;
```

STU_NAME	STU_HEIGHT/30.48
옥한빛	5.807086614
이태연	5.31496063
유가인	5.052493438
조민우	6.167979003
심수정	5.511811024
박희철	
김인중	5.446194226
진현무	5.708661417
김종헌	
옥성우	5.643044619

3.2 조건 검색

●데이터에 널 값의 존재 여부 질의문

```
SQL> select stu_no, stu_name, stu_height
  2  from student
  3  where stu_height is null;
```

STU_NO	STU_NAME	STU_HEIGHT
20132003	박희철	
20131001	김종헌	

3.2 조건 검색

● 널 값이 아닌 행의 결과

```
SQL> select stu_no, stu_name, stu_height
  2  from student
  3  where stu_height is not null;
```

STU_NO	STU_NAME	STU_HEIGHT
20153075	옥한빛	177
20153088	이태연	162
20143054	유가인	154
20152088	조민우	188
20142021	심수정	168
20151062	김인중	166
20141007	진현무	174
20131025	옥성우	172

3.2 조건 검색

●IN 연산자

➢ 여러 개 조건 값 중 하나만 만족하는 행을 처리할 경우 사용한다.

```
SQL> select stu_no, stu_name
  2  from student
  3  where stu_dept in ('컴퓨터정보', '기계');
```

STU_NO	STU_NAME
20153075	옥한빛
20153088	이태연
20143054	유가인
20151062	김인중
20141007	진현무
20131001	김종헌
20131025	옥성우

stu_dept = '컴퓨터정보' or
stu_dept = '기계'

3.3 검색 결과의 순서화 (정렬)

●정렬(SORT) : 데이터를 어떤 기준에 의해 나열하는 것

➢기준(key)
- 기본키(primary key)
- 보조키(secondary kerys)

➢나열방법
- 오름차순(ascending) : 생략가능
- 내림차순(descending)

===> ORDER BY

3.3 검색 결과의 순서화 (정렬)

```
SQL> select stu_no, stu_name
  2  from student
  3  order by stu_no;
```

STU_NO	STU_NAME
20131001	김종헌
20131025	옥성우
20132003	박희철
20141007	진현무
20142021	심수정
20143054	유가인
20151062	김인중
20152088	조민우
20153075	옥한빛
20153088	이태연

3.3 검색 결과의 순서화 (정렬)

● 학생들의 신상을 학번의 내림차순으로 검색하는 질의문

```
SQL> select stu_no, stu_name
  2  from student
  3  order by stu_no desc;
```

STU_NO	STU_NAME
20153088	이태연
20153075	옥한빛
20152088	조민우
20151062	김인중
20143054	유가인
20142021	심수정
20141007	진현무
20132003	박희철
20131025	옥성우
20131001	김종헌

3.3 검색 결과의 순서화 (정렬)

● 별칭이 붙어 있는 열을 기준으로 정렬하는 질의문

```
SQL> select stu_no, stu_name, stu_dept, stu_weight-5 as target
  2  from student
  3  order by target;
```

STU_NO	STU_NAME	STU_DEPT	TARGET
20142021	심수정	전기전자	40
20143054	유가인	기계	42
20153088	이태연	기계	45
20132003	박희철	전기전자	58
20131025	옥성우	컴퓨터정보	58
20141007	진현무	컴퓨터정보	59
20151062	김인중	컴퓨터정보	62
20131001	김종헌	컴퓨터정보	67
20153075	옥한빛	기계	75
20152088	조민우	전기전자	85

3.3 검색 결과의 순서화 (정렬)

● 열의 순서번호를 이용하여 정렬하는 질의문

```
SQL> select stu_no, stu_name, stu_dept, stu_weight-5 as target
  2  from student
  3  order by 4;
```

STU_NO	STU_NAME	STU_DEPT	TARGET
20142021	심수정	전기전자	40
20143054	유가인	기계	42
20153088	이태연	기계	45
20132003	박희철	전기전자	58
20131025	옥성우	컴퓨터정보	58
20141007	진현무	컴퓨터정보	59
20151062	김인중	컴퓨터정보	62
20131001	김종헌	컴퓨터정보	67
20153075	옥한빛	기계	75
20152088	조민우	전기전자	85

3.3 검색 결과의 순서화 (정렬)

●산술식의 열의 이름을 이용하여 정렬하는 질의문

```
SQL> select stu_no, stu_name, stu_dept, stu_weight-5 as target
  2  from student
  3  order by stu_weight-5;
```

STU_NO	STU_NAME	STU_DEPT	TARGET
20142021	심수정	전기전자	40
20143054	유가인	기계	42
20153088	이태연	기계	45
20132003	박희철	전기전자	58
20131025	옥성우	컴퓨터정보	58
20141007	진현무	컴퓨터정보	59
20151062	김인중	컴퓨터정보	62
20131001	김종헌	컴퓨터정보	67
20153075	옥한빛	기계	75
20152088	조민우	전기전자	85

3.3 검색 결과의 순서화 (정렬)

●여러 개의 열을 기준으로 정렬하는 질의문

```
SQL> select stu_no, stu_name, stu_dept, stu_weight-5 as target
  2  from student
  3  order by stu_dept, target;
```

STU_NO	STU_NAME	STU_DEPT	TARGET
20143054	유가인	기계	42
20153088	이태연	기계	45
20153075	옥한빛	기계	75
20142021	심수정	전기전자	40
20132003	박희철	전기전자	58
20152088	조민우	전기전자	85
20131025	옥성우	컴퓨터정보	58
20141007	진현무	컴퓨터정보	59
20151062	김인중	컴퓨터정보	62
20131001	김종헌	컴퓨터정보	67

3.3 검색 결과의 순서화 (정렬)

● 오름차순과 내림차순이 혼용된 질의문

```
SQL> select stu_no, stu_name, stu_dept, stu_weight-5 as target
  2  from student
  3  order by stu_dept, stu_weight-5 desc;
```

STU_NO	STU_NAME	STU_DEPT	TARGET
20153075	옥한빛	기계	75
20153088	이태연	기계	45
20143054	유가인	기계	42
20152088	조민우	전기전자	85
20132003	박희철	전기전자	58
20142021	심수정	전기전자	40
20131001	김종헌	컴퓨터정보	67
20151062	김인중	컴퓨터정보	62
20141007	진현무	컴퓨터정보	59
20131025	옥성우	컴퓨터정보	58

3.3 검색 결과의 순서화 (정렬)

● SELECT절에 포함되지 않는 열을 이용하여 정렬하는 질의문

```
SQL> select stu_no, stu_name, stu_dept, stu_weight-5 as target
  2  from student
  3  order by stu_height;
```

STU_NO	STU_NAME	STU_DEPT	TARGET
20143054	유가인	기계	42
20153088	이태연	기계	45
20151062	김인중	컴퓨터정보	62
20142021	심수정	전기전자	40
20131025	옥성우	컴퓨터정보	58
20141007	진현무	컴퓨터정보	59
20153075	옥한빛	기계	75
20152088	조민우	전기전자	85
20132003	박희철	전기전자	58
20131001	김종헌	컴퓨터정보	67

3.4 함수(Function)

●함수란 ?

- 하나 이상의 인수를 전달받아 처리한 결과 값을 함수의 이름을 변수처럼 활용하여 반환해 주는 프로그램 모듈

- 단일 행 함수는 함수가 정의된 SQL문장이 실행 될 때 각각의 행에 대해 수행되며 각 행별로 하나의 결과 값을 반환함

- 그룹함수는 데이터를 그룹화하고 그룹 각각에 대한 결과를 반환하며, GROUP BY 절을 사용함

3.4 함수(Function)

● 단일행 함수

- 인수의 데이터 타입에 따라

  - ✓숫자함수
  - ✓문자함수
  - ✓날짜함수
  - ✓형변환함수
  - ✓일반함수

- 함수의 활용에서 함수는 인수의 수 및 각 인수의 역할이 중요함으로 인수를 바꾸어가며 충분한 실습이 필요함

3.4 함수(Function)

●숫자 함수

➢숫자 인수를 사용하는 함수

함 수	설 명
ROUND(인수1,인수2)	인수1의 값을 인수2의 자리로 반올림하여 반환
TRUNC(인수1,인수2)	인수1의 값을 인수2 자리까지 유지하고, 나머지는 절삭하여 반환
MOD(인수1, 인수2)	인수1 값을 인수2 값으로 나눈 나머지를 반환
ABS(인수)	인수의 절대값은 반환
FLOOR(인수)	소숫점 이하 자리를 절삭하여 반환

3.4 함수(Function)

●ROUND함수

```
SQL> select round(345.678), round(345.678, 0),
  2  round(345.678, 1), round(345.678, -1)
  3  from dual;
```

ROUND(345.678)	ROUND(345.678,0)	ROUND(345.678,1)	ROUND(345.678,-1)
346	346	345.7	350

3.4 함수(Function)

●문자 함수

➢인수를 문자로 하는 함수

함 수	설 명
LOWER(인수)	인수1을 모두 소문자로 변환하여 반환
UPPER(인수)	인수1을 모두 대문지로 변환하여 반환
INITCAP(인수)	인수1 단어의 첫 번째 문자를 대문자로 변환하여 반환
CONCAT(인수1,인수2)	두개의 문자 인수를 연결하여 반환
SUBSTR(인수1,인수2,인수3,인수4)	문자열인수1의 일부분을 추출하여 반환
LENGTH(인수)	문자인수의 길이를 반환
INSTR(인수1,인수2,인수3,인수4)	문자인수 중 특정 문자의 절대위치를 반환
LPAD(인수1,인수2,인수3)	자릿수를 지정하고 빈 공간을 특정 문자로 왼쪽부터 채워서 문자열을 반환
RPAD(인수1,인수2,인수3)	자릿수를 지정하고 빈 공간을 특정 문자로 오른쪽부터 채워서 문자열을 반환

3.4 함수(Function)

●UPPER함수

```
SQL> select upper('korea')
  2  from dual;
```

UPPER('KOREA')
korea

3.4 함수(Function)

●날짜 함수

➢기본 날짜 형식은 'DD-MON-RR'형식(RR형식 : 21세기와 20세기 데이터의 처리가 가능)

함 수	설 명
SYSDATE	시스템의 오늘 날짜를 반환
날짜 연산	날짜에 +, - 연산을 함
MONTHS_BETWEEN(인수1, 인수2)	인수1, 2의 날수 차이를 반환
NEXT_DAY(인수1,인수2)	인수1에서 가장 가까운 인수2의 요일을 반환
ADD_MONTH(인수1, 인수2)	인수1에 인수2의 달을 더하여 반환
LAST_DAY(인수1)	인수1이 속한 달의 마지막날을 반환
ROUND(인수1,인수2)	인수1의 값을 인수2를 기준으로 반올림하여 반환
TRUNC(인수1)	인수1의 값을 인수2를 기준으로 절사하여 반환

3.4 함수(Function)

```
SQL> select sysdate
  2  from dual;
```

SYSDATE
2016-07-26 15:49

- SYSDATE 함수는 시스템의 현재 날짜를 반환한다.

```
SQL> select next_day(sysdate, '토')
  2  from dual;
```

NEXT_DAY(SYSDATE,'WED')
2016-07-27 16:09

- 결과에서처럼 현재의 날짜를 기준으로 다음에 오늘 수요일을 구하게 된다.
- 이때 요일에 해당하는 숫자를 이용할 수 있다.
- 일-1,월-2, 화-3, 수-4, 목-5, 금-6, 토-7

3.4 함수(Function)

```
SQL> select round(sysdate, 'MON')
  2  from dual;
```

ROUND(SYSDATE,'MON')
2016-08-01

- 날짜도 숫자와 같이 반올림을 할 수 있다.
- 예에서 보듯이 두 번째 인수를 MON으로 하였을 경우 달을 기준으로 반올림하여 결과를 보여준다.
- 두 번째 인수를 조절하면 년, 월, 일 등을 기준으로 반올림할 수 있다.

3.4 함수(Function)

●변환 함수

➢데이터의 형을 변환함

함 수	내 용
TO_NUMBER	문자 데이터를 숫자 데이터로 변환
TO_DATE	문자 데이터를 날짜 데이터로 변환
TO_CHAR	숫자, 날짜 데이터를 문자 데이터로 변환

●TO_CHAR 함수

➢TO_CHAR 함수는 주로 출력에 형식을 지정하기 위해 사용되며, 날짜형, 숫자형 모든 데이터에 사용한다.

3.4 함수(Function)

●TO_CHAR 함수

➢ TO_CHAR 함수는 주로 출력에 형식을 지정하기 위해 사용되며, 날짜형, 숫자형 모든 데이터에 사용한다.

```
SQL> select empno, ename,
                    to_char(hiredate, 'yyyy-mm') as 입사년월
  2  from emp;
```

EMPNO	ENAME	입사년월
7369	SMITH	1980-12
7499	ALLEN	1981-02
7521	WARD	1981-02
7566	JONES	1981-04
7654	MARTIN	1981-09
7698	BLAKE	1981-05
7782	CLARK	1981-06
7788	SCOTT	1987-04
7839	KING	1981-11
.....	.....	.....

3.4 함수(Function)

●TO_NUMBER 함수

➢TO_NUMBER 함수는 숫자 형태의 문자를 숫자로 변환할 때 사용한다.

```
SQL> select to_char(to_number(1234.5678), '9999.999')
  2  from dual;
```

TO_CHAR(TO_NUMBER(1234.5678),'9999.999')
1234.568

```
SQL> select to_char(to_number(1234.5678), '999.999')
  2  from dual;
```

TO_CHAR(TO_NUMBER(1234.5678),'999.999')
########

3.4 함수(Function)

●TO_DATE 함수

➢ TO_DATE 함수는 날짜 형태의 문자를 날짜로 변환할 때 사용한다.

```
SQL> select empno, ename
  2  from emp
  3  where hiredate = to_date('1980-12-17','yy-mm-dd');
```

EMPNO	ENAME
7369	SMITH

3.4 함수(Function)

- 일반 함수
- NVL 함수
  - 인수 1이 널이면, 인수2를 아니면 인수를 반환함

```
SQL> select nvl(stu_height, 0)
  2  from student;
```

NVL(STU_HEIGHT,0)
177
162
154
188
168
0
166
174
0
172

3.4 함수(Function)

●NVL2 함수

➢ 인수 1이 널이 아니면 인수 2를 널이면 인수 3을 반환함

```
SQL> select ename, sal, comm, nvl2(comm, sal+comm, sal)
  2  from emp;
```

ENAME	SAL	COMM	NVL2(COMM,SAL+COMM,SAL)
SMITH	800		800
ALLEN	1600	300	1900
WARD	1250	500	1750
JONES	2975		2975
MARTIN	1250	1400	2650
BLAKE	2850		2850
CLARK	2450		2450
SCOTT	3000		3000
KING	5000		5000
TURNER	1500	0	1500
ADAMS	1100		1100
JAMES	950		950
FORD	3000		3000
MILLER	1300		1300

3.4 함수(Function)

- NULLIF 함수
  - 인수 1과 인수 2의 값을 비교하여 그 값이 같으면 NULL을 아니면 인수1의 값을 반환함

```
SQL> select nvl(nullif('A', 'A'), '널 값')
  2  from dual;
```

NVL(NULLIF('A','A'),'널 값')
널 값

3.4 함수(Function)

●COALESCE 함수

- 인수 1의 값이 널값이 아니면 인수1의 값을, 널값이면 인수 2의 값을 검사하여 널 값이 아니면 인수 2의 값을 반환하고, 아니면 인수 3의 값을 검사하여 값을 반환함

```
SQL> select coalesce(null, null, 10, 100, null)
  2  from dual;
```

COALESCE(NULL,NULL,10,100,NULL)
10

3.4 함수(Function)

●CASE 함수 조건에 따른 처리

```
SELECT column-names
  CASE WHEN condition-1 THEN statement-1,
       WHEN condition-2 THEN statement-2,
                    .....
       WHEN condition-n THEN statement-n,
       ELSE statement
   END
FROM table-name;
```

3.4 함수(Function)

●CASE 함수

```
SQL> select empno, ename, sal,
  2       case job when 'SALESMAN' then sal * 1.1
  3                when 'CLERK' then sal * 1.15
  4                when 'MANAGER' then sal * 1.2
  5                else sal
  6       end as 급여인상
  7   from emp;
```

EMPNO	ENAME	SAL	급여인상
7369	SMITH	800	920
7499	ALLEN	1600	1760
7521	WARD	1250	1375
7566	JONES	2975	3570
7654	MARTIN	1250	1375
7698	BLAKE	2850	3420
7782	CLARK	2450	2940
7788	SCOTT	3000	3000
.....	.....	.....	.....

3.4 함수(Function)

●DECODE 함수

```
DECODE ( column-name, condition-1, statement-1,
                      condition-2, statement-2,
                      ..................
                      condition-n, statement-n,
         statement)
```

3.4 함수(Function)

● DECODE 함수

```
SQL> select empno, ename, job, sal,
  2  decode(job, 'SALESMAN', sal * 1.1,
  3  'CLERK' , sal * 1.15,
  4  'MANAGER', sal * 1.2,
  5  sal) as 인상된급여
  6  from emp;
```

EMPNO	ENAME	JOB	SAL	인상된급여
7369	SMITH	CLERK	800	960
7499	ALLEN	SALESMAN	1600	1760
7521	WARD	SALESMAN	1250	1375
7566	JONES	MANAGER	2975	3867.5
7654	MARTIN	SALESMAN	1250	1375
7698	BLAKE	MANAGER	2850	3705
7782	CLARK	MANAGER	2450	3185
7788	SCOTT	ANALYST	3000	3000
7839	KING	PRESIDENT	5000	5000
.....	.....	.....	.....	.....

3.4 함수(Function)

●그룹함수

- ➢여러 행에 대한 연산 즉 평균, 개수 등의 결과값을 반환하는 함수
- ➢SELECT문에서 GROUP BY절을 사용함

●그룹함수의 종류

함 수	기 능
COUNT( )	조건을 만족하는 열의 데이터 값들의 개수를 반환
COUNT(*)	모든 행의 개수를 반환
SUM( )	조건을 만족하는 열의 데이터 값들의 합을 반환
AVG( )	조건을 만족하는 열의 데이터 값들의 평균을 반환
MAX( )	조건을 만족하는 열의 데이터 값들 중 최댓값을 반환
MIN( )	조건을 만족하는 열의 데이터 값들 중 최솟값을 반환
STDDEV( )	조건을 만족하는 열의 데이터 값들의 표준편차를 반환
VARIANCE( )	조건을 만족하는 열의 데이터 값들의 분산 값을 반환

3.4 함수(Function)

● MAX와 MIN함수

```
SQL> select max(enr_grade), min(enr_grade)
  2  from enrol;
```

MAX(ENR_GRADE)	MIN(ENR_GRADE)
81	41

```
SQL> select min(stu_weight), max(stu_weight)
  2  from student
  3  where stu_dept = '기계';
```

MIN(STU_WEIGHT)	MAX(STU_WEIGHT)
47	80

3.4 함수(Function)

●COUNT 함수

```
SQL> select count(*), count(stu_height)
  2  from student;
```

COUNT(*)	COUNT(STU_HEIGHT)
10	8

```
SQL> select count(stu_dept), count(distinct stu_dept)
  2  from student;
```

COUNT(STU_DEPT)	COUNT(DISTINCTSTU_DEPT)
10	3

3.4 함수(Function)

●SUM과 AVG함수

```
SQL> select sum(stu_weight), to_char(avg(stu_weight), '9999.99')
  2  from student
  3  where stu_dept = '컴퓨터정보';
```

SUM(STU_WEIGHT)	TO_CHAR(AVG(STU_WEIGHT),'9999.99')
266	66.50

```
SQL> select count(*) as 학생, sum(stu_height) as 신장합,
  2  count(stu_height) "해당학생수", avg(stu_height) "평균신장"
  3  from student;
```

학생	신장합	해당학생수	평균신장
10	1361	8	170.125

3.4 함수(Function)

● 단일행을 이용한 GROUP BY절

```
SQL> select stu_dept, avg(stu_weight)
  2  from student
  3  group by stu_dept;
```

STU_DEPT	AVG(STU_WEIGHT)
전기전자	66
기계	59
컴퓨터정보	66.5

3.4 함수(Function)

●단일행을 이용한 GROUP BY절

```
SQL> select stu_dept, count(*)
  2  from student
  3  where stu_weight >= 50
  4  group by stu_dept;
```

STU_DEPT	COUNT(*)
전기전자	2
기계	2
컴퓨터정보	4

3.4 함수(Function)

●다중열 GROUP BY절

```
SQL> select stu_dept, stu_grade, count(*)
  2  from student
  3  group by stu_dept, stu_grade;
```

STU_DEPT	STU_GRADE	COUNT(*)
기계	2	1
기계	1	2
전기전자	1	1
컴퓨터정보	1	1
컴퓨터정보	3	2
컴퓨터정보	2	1
전기전자	2	1
전기전자	3	1

3.4 함수(Function)

●HAVING절 사용

➢그룹함수를 적용한 결과에 다시 조건을 부여할 때는 HAVING 절을 사용한다.

```
SQL> select stu_grade, avg(stu_height)
  2  from student
  3  where stu_dept = '기계'
  4  group by stu_grade having avg(stu_height) >= 160;
```

STU_GRADE	AVG(STU_HEIGHT)
1	169.5

3.4 함수(Function)

● HAVING절 사용

```
SQL> select stu_dept, max(stu_height)
  2  from student
  3  group by stu_dept having max(stu_height) >= 175;
```

STU_DEPT	MAX(STU_HEIGHT)
전기전자	188
기계	177

```
SQL> select to_char(max(avg(stu_height)),'999.99')
  2  from student
  3  group by stu_dept;
```

TO_CHAR(MAX(AVG(STU_HEIGHT)),'999.99')
178.00

3장을 마치며……

Q & A