

1 REM Author : Henry  
2 REM Date : 2024.06.12  
3 REM Objective :  
4 REM Environment : Ubuntu Server 22.04 LTS, MySQL Workbench 8.0 CE, MySQL Community  
Server 8.0.37-0ubuntu0.22.04.3 (ubuntu)

5  
6 REM 여러 행(그룹, 다중행) 함수

#### 7 1. 개념

- 8 1) 그룹 함수는 행 집합에 작용하여 그룹 당 하나의 결과를 생성한다.
- 9 2) 이 집합은 전체 테이블 또는 그룹으로 분류된 테이블이다.
- 10 3) 여러 행을 바탕으로 하나의 결과 값을 도출해 내기 위해 사용하는 함수
- 11 4) 다중행 함수를 사용한 SELECT 절에는 기본적으로 여러 행이 결과로 나올 수 있는 열을  
함께 사용할 수 없다.

12 -SELECT ename, SUM(sal)  
13 FROM emp;

14

15

#### 16 2. 종류

- 17 1) AVG
- 18 2) COUNT
- 19 3) MAX
- 20 4) MIN
- 21 5) SUM
- 22 6) STDDEV
- 23 7) VARIANCE

24

25

#### 26 3. 사용 지침

- 27 1) DISTINCT를 지정하면 함수는 중복되지 않는 값만 검토하고 ALL을 지정하면 중복 값을  
포함한 모든 값을 검토한다. 기본은 ALL
- 28 2) 인수에 대한 데이터 유형은 CHAR, VARCHAR, NUMBER 또는 DATE이며 expression 형식으로  
나열됨
- 29 3) COUNT(\*)를 제외한 모든 그룹 함수는 NULL 값을 무시

30

31

#### 32 4. AVG

- 33 1) Returns the average value of expr.
- 34 2) The DISTINCT option can be used to return the average of the distinct values of expr.
- 35 3) Syntax  
36 AVG([DISTINCT | ALL ] expression)
- 37 4) expression 값의 평균
- 38 5) NULL 무시

39

40 SELECT AVG(sal), MAX(sal), MIN(sal), SUM(sal)  
41 FROM emp  
42 WHERE job LIKE 'SALES%';

43

44 SELECT AVG(comm)  
45 FROM emp;

46

47 SELECT AVG(NVL(comm, 0))

48 FROM emp;

49

50

## 51 5. COUNT

52 1>Returns a count of the number of non-NULL values of expression.

53 2)Syntax

54 COUNT( { \* | [DISTINCT | ALL ] expression } )

55 3)행 수, expression은 NULL을 제외한 값을 계산

56 4)\*을 사용하면 중복 행 및 NULL 값을 가진 행을 포함하여 선택한 행 모두를 계산

57

58 SELECT COUNT(\*)

59 FROM emp

60 WHERE deptno = 30;

61

62 SELECT COUNT(comm)

63 FROM emp

64 WHERE deptno = 30;

65

66 SELECT COUNT(DISTINCT (deptno))

67 FROM emp;

68

69

## 70 6. MAX

71 1>Returns the maximum value in a set of values.

72 2)Syntax

73 MAX([DISTINCT | ALL ] expression)

74 3)expression의 최대값이며 NULL 값을 무시

75

76

## 77 7. MIN

78 1>Returns the minimum value in a set of values.

79 2)Syntax

80 MIN([DISTINCT | ALL ] expression)

81 3)expression의 최소값이며 NULL 값을 무시

82

83 SELECT MIN(hiredate), MAX(hiredate)

84 FROM emp;

85

86 SELECT MIN(ename), MAX(ename)

87 FROM emp;

88

89

## 90 8. SUM

91 1)Calculates the sum of a set of values.

92 2)Syntax

93 SUM(expression)

94 3)NULL 값 무시

95

96

## 97 9. STDDEV

98 1>Returns the population standard deviation of value.

2)Syntax

**STDDEV**(**[DISTINCT | ALL ]** expression)

3)NULL 값을 무시

4)분산의 제곱근

**SELECT STDDEV**(sal)

**FROM** emp;

## 10. VARIANCE

1>Returns the population standard **variance of value**.

2)Syntax

**VARIANCE**(**[DISTINCT | ALL ]** expression)

3)NULL 값을 무시

4)편차 제곱의 평균

## REM GROUP BY

1. 지금까지는 테이블을 하나의 대형 정보 그룹으로 취급했음

2. 테이블 정보를 더 작은 그룹으로 나눠야 할 경우 **GROUP BY**절을 사용

3. **GROUP BY** 절을 사용하여 테이블 행을 그룹으로 나눈 후 그룹 함수를 사용하여 각 그룹에 대한 요약 정보를 반환 가능

4. 지침

1)GROUP BY절에 열을 포함시켜야 한다.

2)GROUP BY절엔 열 별칭을 사용할 수 없다.

3)기본적으로 **GROUP BY**목록에 포함된 열은 오름차순으로 정렬된다. 무시하려면 **ORDER BY** 사용

5. Syntax

**SELECT**

**FROM**

**WHERE**

**GROUP BY;**

-SELECT 절

1)SELECT 절에서 **GROUP** 함수(복수행 함수)와 **column**이 같이 기술되면, 반드시 **GROUP BY** 절이 기술되어야 한다.

**SELECT** ename, sal, **MAX**(sal)

**FROM** emp

**WHERE** sal = **MAX**(sal);

**SELECT** deptno, **MAX**(sal)

**FROM** emp

**GROUP BY** deptno

**ORDER BY** deptno;

**SELECT** deptno, **MAX**(sal), **MIN**(sal), **SUM**(sal), **AVG**(sal)

**FROM** emp

```
147 GROUP BY deptno
148 ORDER BY deptno DESC;
```

149  
150 2) 그러나 **SELECT** 절에 복수행함수만 기술되고, **column** 을 사용하지 않았다면 **GROUP BY** 를 필요로 하지 않는다.

```
151
152 SELECT MAX(sal), MIN(sal), SUM(sal), AVG(sal)
153 FROM emp
```

154  
155 3) Multiple **Grouping**  
156 -부서별, 업무별로 그룹하여 결과를 부서번호, 업무, 인원수, 급여의 평균, 급여의 합을 구하시오.

```
157
158 SELECT deptno, job, COUNT(*), AVG(sal), SUM(sal)
159 FROM emp
160 GROUP BY deptno, job
161 ORDER BY deptno ASC, job DESC;
```

162  
163 4) 여러 열을 기준으로 분류  
164 -하나 이상의 **GROUP BY** 열 나열  
165 -열 순서에 따라 결과의 기본 정렬 순서를 결정

```
166
167 SELECT deptno, job, SUM(sal)
168 FROM emp
169 GROUP BY deptno, job;
```

170  
171  
172  
173 REM **HAVING**

174 1. **WHERE** 절에서는 복수행 함수를 사용할 수 없다.

```
175
176 SELECT deptno, COUNT(*), SUM(sal)
177 FROM emp
178 WHERE COUNT(*) >= 4
179 GROUP BY deptno;
```

180  
181  
182 2. **GROUP BY**의 조건절은 **HAVING** 이다.

```
183
184 SELECT deptno, COUNT(*), SUM(sal)
185 FROM emp
186 GROUP BY deptno
187 HAVING COUNT(*) >= 4;
```

188  
189 -사원테이블에서 업무별 급여의 평균이 3000불 이상인 업무에 대해, 업무명, 평균급여, 급여의 합을 구하시오.

```
190 SELECT job, AVG(sal), SUM(sal)
191 FROM emp
192 GROUP BY job
193 HAVING AVG(sal) >= 3000;
```

194

-사원테이블에서 전체 월급이 5000불을 초과하는 각 업무에 대해 업무이름과 월 급여의 합계를 출력하라. 단, 판매원은 제외하고 월급여 합계의 내림차순으로 출력하라.

```
SELECT job, SUM(sal)
FROM emp
WHERE job NOT LIKE 'SA%'
GROUP BY job
HAVING SUM(sal) > 5000
ORDER BY SUM(sal) DESC;
```

```
SELECT deptno, AVG(sal)
FROM emp
GROUP BY deptno;
```

3. GROUP BY 열은 SELECT 목록에 포함시키지 않아도 된다. BUT 별 의미 없음.

```
SELECT AVG(sal)
FROM emp
GROUP BY deptno;
```

4. ORDER BY 절 사용 가능

```
SELECT deptno, AVG(sal)
FROM emp
GROUP BY deptno
ORDER BY AVG(sal);
```

REM HAVING 절을 사용한 분류된 행을 포함 또는 제외

1. SQL-92 버전 및 이전 버전에서는 SELECT 목록의 열 또는 표현식 중 집계 함수가 아닌 것은 GROUP BY 절에 포함시켜야 한다.

```
SELECT deptno, COUNT(ename)
FROM emp; --> SQL92 및 이전버전에서는 Error, 이후 버전은 가능
```

--SQL92 및 이전버전에서 수정

```
SELECT deptno, COUNT(ename)
FROM emp
GROUP BY deptno;
```

2. 그룹 결과 제외 : HAVING 절

1) WHERE 를 사용하여 행을 제한하는 것과 같이 HAVING 절을 사용하여 그룹을 제한  
2) 그룹 함수의 결과를 기반으로 행을 제한할 경우 GROUP BY 절 및 HAVING 절이 모두 있어야

3) 주의할 점 : WHERE 절로 그룹을 제한할 수 없음

```
SELECT deptno, MAX(sal)
FROM emp
GROUP BY deptno
HAVING MAX(sal) > 2900;
```

```
243
244 SELECT deptno, AVG(sal)
245 FROM emp
246 WHERE AVG(sal) > 2000
247 GROUP BY deptno; ==> Error
```

```
248
249 SELECT deptno, AVG(sal)
250 FROM emp
251 GROUP BY deptno
252 HAVING AVG(sal) > 2000;
```

```
253
254 SELECT deptno, COUNT(*), SUM(sal)
255 FROM emp
256 GROUP BY deptno
257 HAVING COUNT(*) > 2;
```

```
258
259
260
```

```
261 REM ROLLUP
```

```
262 1. GROUP BY절과 함께 사용
```

```
263 2. GROUP BY절에 의해서 그룹핑 된 집합 결과에 대해 좀 더 상세한 정보를 반환하는 기능을
수행
```

```
264 3. 보통 SELECT 절에 ROLLUP을 사용함으로써 보통의 SELECT된 데이터와 그 데이터의 총계를
구할 수 있다.
```

```
265
```

```
266 SELECT job, SUM(sal)
267 FROM emp
268 GROUP BY job;
```

```
269
```

```
270 --ROLLUP을 사용해서 직무별로 급여 합계와 총계를 구한다.
```

```
271 SELECT job, SUM(sal)
272 FROM emp
273 GROUP BY job
274 WITH ROLLUP;
```

```
275
```

```
276 4. GROUP BY 칼럼이 두 개 이상인 경우 합계 및 소계까지 계산되어 표시된다.
```

```
277 SELECT job, deptno, SUM(sal)
278 FROM emp
279 GROUP BY job, deptno
280 WITH ROLLUP;
```