

```

1 REM Author :
2 REM Date :
3 REM Objective : 4. Aggregate Function
4 REM Environment : Ubuntu Server 20.04 LTS, HeidiSQL 12.0, MySQL Community Server 8.0.29
5
6 REM 여러 행(그룹) 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
11
12 2. 종류
13 1) AVG
14 2) COUNT
15 3) MAX
16 4) MIN
17 5) SUM
18 6) STDDEV
19 7) VARIANCE
20
21
22 3. 사용 지침
23 1) DISTINCT를 지정하면 함수는 중복되지 않는 값만 검토하고 ALL을 지정하면 중복 값을 포함한 모든 값을 검토한다.
24 기본은 ALL
25 2) 인수에 대한 데이터 유형은 CHAR, VARCHAR, NUMBER 또는 DATE이며 expression 형식으로 나열됨
26 3) COUNT(*)를 제외한 모든 그룹 함수는 NULL 값을 무시
27
28 4. AVG
29 1) Returns the average value of expr.
30 2) The DISTINCT option can be used to return the average of the distinct values of expr.
31 3) Syntax
32 AVG([DISTINCT | ALL ] expression)
33 4) expression 값의 평균
34 5) NULL 무시
35
36 SELECT AVG(sal), MAX(sal), MIN(sal), SUM(sal)
37 FROM emp
38 WHERE job LIKE 'SALES%';
39
40 SELECT AVG(comm)
41 FROM emp;
42
43 SELECT AVG(NVL(comm, 0))
44 FROM emp;
45
46
47 5. COUNT
48 1) Returns a count of the number of non-NULL values of expression.
49 2) Syntax
50 COUNT( { * | [DISTINCT | ALL ] expression } )
51 3) 행 수, expression은 NULL을 제외한 값을 계산
52 4) *을 사용하면 중복 행 및 NULL 값을 가진 행을 포함하여 선택한 행 모두를 계산
53
54 SELECT COUNT(*)
55 FROM emp
56 WHERE deptno = 30;
57
58 SELECT COUNT(comm)
59 FROM emp
60 WHERE deptno = 30;
61
62 SELECT COUNT(DISTINCT (deptno))
63 FROM emp;
64
65
66 6. MAX

```

```

67 1)Returns the maximum value in a set of values.
68 2)Syntax
69 MAX([DISTINCT | ALL ] expression)
70 3)expression의 최대값이며 NULL 값을 무시
71
72
73 7. MIN
74 1)Returns the minimum value in a set of values.
75 2)Syntax
76 MIN([DISTINCT | ALL ] expression)
77 3)expression의 최소값이며 NULL 값을 무시
78
79 SELECT MIN(hiredate), MAX(hiredate)
80 FROM emp;
81
82 SELECT MIN(ename), MAX(ename)
83 FROM emp;
84
85
86 8. SUM
87 1)Calculates the sum of a set of values.
88 2)Syntax
89 SUM(expression)
90 3)NULL 값 무시
91
92
93 9. STDDEV
94 1)Returns the population standard deviation of value.
95 2)Syntax
96 STDDEV([DISTINCT | ALL ] expression)
97 3)NULL 값을 무시
98 4)분산의 제곱근
99
100 SELECT STDDEV(sal)
101 FROM emp;
102
103
104 10. VARIANCE
105 1)Returns the population standard variance of value.
106 2)Syntax
107 VARIANCE([DISTINCT | ALL ] expression)
108 3)NULL 값을 무시
109 4)편차 제곱의 평균
110
111
112
113 REM GROUP BY
114 1. 지금까지는 테이블을 하나의 대형 정보 그룹으로 취급했음
115 2. 테이블 정보를 더 작은 그룹으로 나눠야 할 경우 GROUP BY절을 사용
116 3. GROUP BY 절을 사용하여 테이블 행을 그룹으로 나눈 후 그룹 함수를 사용하여 각 그룹에 대한 요약 정보를 반환 가능
117 4. 지침
118 1)GROUP BY절에 열을 포함시켜야 한다.
119 2)GROUP BY절엔 열 별칭을 사용할 수 없다.
120 3)기본적으로 GROUP BY목록에 포함된 열은 오름차순으로 정렬된다. 무시하려면 ORDER BY사용
121 5. Syntax
122
123 SELECT
124 FROM
125 WHERE
126 GROUP BY;
127
128
129 -SELECT 절
130 1)SELECT 절에서 GROUP 함수(복수행 함수)와 column이 같이 기술되면, 반드시 GROUP BY 절이 기술되어야 한다.
131
132 SELECT ename, sal, MAX(sal)

```

```
133 FROM emp
134 WHERE sal = MAX(sal);
```

```
135
136 SELECT deptno, MAX(sal)
137 FROM emp
138 GROUP BY deptno
139 ORDER BY deptno;
```

```
140
141 SELECT deptno, MAX(sal), MIN(sal), SUM(sal), AVG(sal)
142 FROM emp
143 GROUP BY deptno
144 ORDER BY deptno DESC;
```

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

```
145
146 SELECT MAX(sal), MIN(sal), SUM(sal), AVG(sal)
147 FROM emp
```

### 3) Multiple Grouping

-부서별, 업무별로 그룹하여 결과를 부서번호, 업무, 인원수, 급여의 평균, 급여의 합을 구하시오.

```
150
151 SELECT deptno, job, COUNT(*), AVG(sal), SUM(sal)
152 FROM emp
153 GROUP BY deptno, job
154 ORDER BY deptno ASC, job DESC;
```

### 4) 여러 열을 기준으로 분류

- 하나 이상의 **GROUP BY** 열 나열
- 열 순서에 따라 결과의 기본 정렬 순서를 결정

```
155
156 SELECT deptno, job, SUM(sal)
157 FROM emp
158 GROUP BY deptno, job;
```

## REM HAVING

1. **WHERE** 절에서는 복수행 함수를 사용할 수 없다.

```
159
160 SELECT deptno, COUNT(*), SUM(sal)
161 FROM emp
162 WHERE COUNT(*) >= 4
163 GROUP BY deptno;
```

2. **GROUP BY**의 조건절은 **HAVING** 이다.

```
164
165 SELECT deptno, COUNT(*), SUM(sal)
166 FROM emp
167 GROUP BY deptno
168 HAVING COUNT(*) >= 4;
```

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

```
169
170 SELECT job, AVG(sal), SUM(sal)
171 FROM emp
172 GROUP BY job
173 HAVING AVG(sal) >= 3000;
```

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

```
174
175 SELECT job, SUM(sal)
176 FROM emp
177 WHERE job NOT LIKE 'SA%'
178 GROUP BY job
179 HAVING SUM(sal) > 5000
180 ORDER BY SUM(sal) DESC;
```

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

202  
203  
204 3. **GROUP BY** 절은 **SELECT** 목록에 포함시키지 않아도 된다. BUT 별 의미 없음.

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

```
208
209
210 4. ORDER BY 절 사용 가능
211 SELECT deptno, AVG(sal)
212 FROM emp
213 GROUP BY deptno
214 ORDER BY AVG(sal);
```

```
215
216
217
218 REM HAVING 절을 사용한 분류된 행을 포함 또는 제외
```

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

```
220
221 SELECT deptno, COUNT(ename)
222 FROM emp; --> SQL92 및 이전버전에서는 Error, 이후 버전은 가능
```

```
223
224 --SQL92 및 이전버전에서 수정
225 SELECT deptno, COUNT(ename)
226 FROM emp
227 GROUP BY deptno;
```

```
228
229
230 2. 그룹 결과 제외 : HAVING 절
231 1) WHERE 를 사용하여 행을 제한하는 것과 같이 HAVING 절을 사용하여 그룹을 제한
232 2) 그룹 함수의 결과를 기반으로 행을 제한할 경우 GROUP BY 절 및 HAVING 절이 모두 있어야
233 3) 주의할 점 : WHERE 절로 그룹을 제한할 수 없음
```

```
234
235 SELECT deptno, MAX(sal)
236 FROM emp
237 GROUP BY deptno
238 HAVING MAX(sal) > 2900;
```

```
239
240 SELECT deptno, AVG(sal)
241 FROM emp
242 WHERE AVG(sal) > 2000
243 GROUP BY deptno; ==> Error
```

```
244
245 SELECT deptno, AVG(sal)
246 FROM emp
247 GROUP BY deptno
248 HAVING AVG(sal) > 2000;
```

```
249
250 SELECT deptno, COUNT(*), SUM(sal)
251 FROM emp
252 GROUP BY deptno
253 HAVING COUNT(*) > 2;
```

```
254
255
256
257 REM ROLLUP
```

258 1. **GROUP BY** 절과 함께 사용  
259 2. **GROUP BY** 절에 의해서 그룹핑 된 집합 결과에 대해 좀 더 상세한 정보를 반환하는 기능을 수행  
260 3. 보통 **SELECT** 절에 **ROLLUP** 을 사용함으로써 보통의 **SELECT** 된 데이터와 그 데이터의 총계를 구할 수 있다.

```
261
262 SELECT job, SUM(sal)
263 FROM emp
264 GROUP BY job;
```

265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276

--ROLLUP을 사용해서 직무별로 급여 합계와 총계를 구한다.

```
SELECT job, SUM(sal)
FROM emp
GROUP BY job
WITH ROLLUP;
```

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

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