REPORT

과	목 :	데이터베이스이론및실습
제출일자	:	2024.03.25
담당교수	:	권기현
학	과 :	정보통신공학과
학	번 :	201920479
이	름 :	김행복

-- 7-1. SUM 함수를 사용하여 급여 합계 출력하기

SELECT SUM(SAL) FROM EMP;

	SUM(...
1	29025

-- 7-2. SUM 함수를 사용하여 사원 이름과 급여 합계 출력하기

SELECT ENAME, SUM(SAL) FROM EMP;

```
ORA-00937: 단일 그룹의 그룹 함수가 아닙니다
00937. 00000 -  "not a single-group group function"
*Cause:
*Action:
5행, 8열에서 오류 발생
```

-- 7-3. 추가 수당 합계 구하기

SELECT SUM(COMM) FROM EMP;

	SUM(COMM)
1	2200

-- 7-4. 급여 합계 구하기(DISTINCT, ALL) 사용

SELECT SUM(DISTINCT SAL), SUM(ALL SAL), SUM(SAL) FROM EMP;

	SUM(DISTINCTSAL)	SUM(ALLSAL)	SUM(SAL)
1	24775	29025	29025

-- 7-5. EMP 테이블의 데이터 개수 출력하기

SELECT COUNT(*) FROM EMP;

	COUNT(*)
1	14

-- 7-6. 부서 번호가 30번인 직원 수 구하기

SELECT COUNT(*) FROM EMP WHERE DEPTNO=30;

	COUNT(*)
1	6

-- 7-7. COUNT 함수를 사용하여 급여 개수 구하기(DISCINCT, ALL) 사용

```
SELECT COUNT(DISTINCT SAL), COUNT(ALL SAL), COUNT(SAL) FROM EMP;
```

	COUNT(DISTINCTSAL)	COUNT(ALLSAL)	COUNT(SAL)
1	12	14	14

-- 7-8. COUNT 함수를 사용하여 추가 수당 열 개수 출력하기

```
SELECT COUNT(COMM) FROM EMP;
```

	COUNT(COMM)
1	4

-- 7-9. COUNT 함수와 IS NOT NULL을 사용하여 추가 수당 열 개수 출력하기

```
SELECT COUNT(COMM) FROM EMP WHERE COMM IS NOT NULL;
```

	COUNT(COMM)
1	4

-- 7-10. 부서 번호가 10번인 사원들의 최대 급여 출력하기

```
SELECT MAX(SAL) FROM EMP WHERE DEPTNO=10;
```

	MAX(SAL)
1	5000

-- 7-11. 부서 번호가 10번인 사원들의 최소 급여 출력하기

```
SELECT MIN(SAL) FROM EMP WHERE DEPTNO=10;
```

	MIN(SAL)
1	1300

-- 7-12. 부서 번호가 20인 사원의 입사일 중 제일 최근 입사일 출력하기

```
SELECT MAX(HIREDATE) FROM EMP WHERE DEPTNO=20;
```

	MAX(HIREDATE)
1	1983/01/12

-- 7-13. 부서 번호가 20인 사원의 입사일 중 제일 오래된 입사일 출력하기

```
SELECT MIN(HIREDATE) FROM EMP WHERE DEPTNO=20;
```

	MIN(HIREDATE)
1	1980/12/17

-- 7-14. 부서 번호가 30인 사원들의 평균 급여 출력하기

```
SELECT AVG(SAL) FROM EMP WHERE DEPTNO=30;
```

	AVG(SAL)
1	1566.6666666666666666666666666666666666667

-- 7-15. DISTINCT로 중복을 제거한 급여 열의 평균 급여 구하기

```
SELECT AVG(DISTINCT SAL) FROM EMP WHERE DEPTNO=30;
```

	AVG(DISTINCTSAL)
1	1630

-- 7-16. 집합 연산자를 사용하여 각 부서별 평균 급여 출력하기

```
SELECT AVG(SAL), '10' AS DEPTNO FROM EMP WHERE DEPTNO=10
UNION ALL
SELECT AVG(SAL), '20' AS DEPTNO FROM EMP WHERE DEPTNO=20
UNION ALL
SELECT AVG(SAL), '30' AS DEPTNO FROM EMP WHERE DEPTNO=30;
```

	AVG(SAL)	DEPTNO
1	2916.6666666666666666666666666666666666667	10
2	2175	20
3	1566.6666666666666666666666666666666666667	30

-- 7-17. GROUP BY를 사용하여 부서별 평균 급여 출력하기

```
SELECT AVG(SAL), DEPTNO FROM EMP GROUP BY DEPTNO;
```

	AVG(SAL)	DEPTNO
1	2175	20
2	1566.6666666666666666666666666666666666667	30
3	2916.6666666666666666666666666666666666667	10

-- 7-18. 부서 번호 및 직책별 평균 급여로 정렬하기

SELECT DEPTNO, JOB, AVG(SAL) FROM EMP GROUP BY DEPTNO, JOB ORDER BY DEPTNO, JOB;

	DEPTNO	JOB	AVG(SAL)
1	10	CLERK	1300
2	10	MANAGER	2450
3	10	PRESIDENT	5000
4	20	ANALYST	3000
5	20	CLERK	950
6	20	MANAGER	2975
7	30	CLERK	950
8	30	MANAGER	2850
9	30	SALESMAN	1400

-- 7-19. GROUP BY절에 없는 열을 SELECT절에 포함했을 경우

SELECT ENAME, DEPTNO, AVG(SAL) FROM EMP GROUP BY DEPTNO;

```
ORA-00979: GROUP BY 표현식이 아닙니다.
00979. 00000 -  "not a GROUP BY expression"
*Cause:
*Action:
60행, 8열에서 오류 발생
```

-- 7-20. GROUP BY 절에 HAVING 절을 사용하여 출력하기

SELECT DEPTNO, JOB, AVG(SAL) FROM EMP

GROUP BY DEPTNO, JOB

HAVING AVG(SAL) >= 2000

ORDER BY DEPTNO, JOB;

	DEPTNO	JOB	AVG(SAL)
1	10	MANAGER	2450
2	10	PRESIDENT	5000
3	20	ANALYST	3000
4	20	MANAGER	2975
5	30	MANAGER	2850

-- 7-21. HAVING 절 대신 WHERE 절을 잘못 사용했을 경우

```
SELECT DEPTNO, JOB, AVG(SAL) FROM EMP

WHERE AVG(SAL) >= 2000

GROUP BY DEPTNO, JOB

ORDER BY DEPTNO, JOB;
```

```
ORA-00934: 그룹 함수는 허가되지 않습니다
00934. 00000 -  "group function is not allowed here"
*Cause:
*Action:
70행, 7열에서 오류 발생
```

-- 7-22. WHERE절을 사용하지 않고 HAVING 절만 사용한 경우

```
SELECT DEPTNO, JOB, AVG(SAL) FROM EMP

GROUP BY DEPTNO, JOB

HAVING AVG(SAL) >= 2000

ORDER BY DEPTNO, JOB;
```

	DEPTNO	JOB	AVG(SAL)
1	10	MANAGER	2450
2	10	PRESIDENT	5000
3	20	ANALYST	3000
4	20	MANAGER	2975
5	30	MANAGER	2850

-- 7-23. WHERE 절과 HAVING 절 모두 사용한 경우

```
SELECT DEPTNO, JOB, AVG(SAL) FROM EMP
WHERE SAL <= 3000
GROUP BY DEPTNO, JOB
HAVING AVG(SAL) >= 2000
ORDER BY DEPTNO, JOB;
```

	DEPTNO	JOB	AVG(SAL)
1	10	MANAGER	2450
2	20	ANALYST	3000
3	20	MANAGER	2975
4	30	MANAGER	2850

-- 7-24. 기존 GROUP BY 절만 사용한 그룹화

```
SELECT DEPTNO, JOB, COUNT(*), MAX(SAL), SUM(SAL), AVG(SAL) FROM EMP
GROUP BY DEPTNO, JOB
ORDER BY DEPTNO, JOB;
```

	DEPTNO	JOB	COUNT(*)	MAX(SAL)	SUM(SAL)	AVG(SAL)
1	10	CLERK	1	1300	1300	1300
2	10	MANAGER	1	2450	2450	2450
3	10	PRESIDENT	1	5000	5000	5000
4	20	ANALYST	2	3000	6000	3000
5	20	CLERK	2	1100	1900	950
6	20	MANAGER	1	2975	2975	2975
7	30	CLERK	1	950	950	950
8	30	MANAGER	1	2850	2850	2850
9	30	SALESMAN	4	1600	5600	1400

-- 7-25. ROLLUP 함수를 적용한 그룹화

```
SELECT DEPTNO, JOB, COUNT(*), MAX(SAL), SUM(SAL), AVG(SAL) FROM EMP

GROUP BY ROLLUP(DEPTNO, JOB);
```

	DEPTNO	JOB	COUNT(*)	MAX(SAL)	SUM(SAL)	AVG(SAL)
1	20	CLERK	2	1100	1900	950
2	30	SALESMAN	4	1600	5600	1400
3	20	MANAGER	1	2975	2975	2975
4	30	MANAGER	1	2850	2850	2850
5	10	MANAGER	1	2450	2450	2450
6	20	ANALYST	2	3000	6000	3000
7	10	PRESIDENT	1	5000	5000	5000
8	30	CLERK	1	950	950	950
9	10	CLERK	1	1300	1300	1300
10	20	(null)	5	3000	10875	2175
11	30	(null)	6	2850	9400	1566.6666666666666666666666666666666666667
12	10	(null)	3	5000	8750	2916.6666666666666666666666666666666666667
13	(null)	(null)	14	5000	29025	2073.2142857142857142857142857142857142857

-- 7-26. CUBE 함수를 적용한 그룹화

```
SELECT DEPTNO, JOB, COUNT(*), MAX(SAL), SUM(SAL), AVG(SAL) FROM EMP

GROUP BY CUBE(DEPTNO, JOB)

ORDER BY DEPTNO, JOB;
```

	DEPTNO	JOB	COUNT(*)	MAX(SAL)	SUM(SAL)	AVG(SAL)
1	10	CLERK	1	1300	1300	1300
2	10	MANAGER	1	2450	2450	2450
3	10	PRESIDENT	1	5000	5000	5000
4	10	(null)	3	5000	8750	2916.6666666666666666666666666666666666667
5	20	ANALYST	2	3000	6000	3000
6	20	CLERK	2	1100	1900	950
7	20	MANAGER	1	2975	2975	2975
8	20	(null)	5	3000	10875	2175
9	30	CLERK	1	950	950	950
10	30	MANAGER	1	2850	2850	2850
11	30	SALESMAN	4	1600	5600	1400
12	30	(null)	6	2850	9400	1566.6666666666666666666666666666666666667
13	(null)	ANALYST	2	3000	6000	3000
14	(null)	CLERK	4	1300	4150	1037.5

-- 7-27. DEPTNO를 먼저 그룹화 한 후 ROLLUP 함수에 JOB 지정하기

```
SELECT DEPTNO, JOB, COUNT(*) FROM EMP

GROUP BY DEPTNO, ROLLUP(JOB);
```

	DEPTNO	JOB	COUNT(*)
1	20	CLERK	2
2	30	SALESMAN	4
3	20	MANAGER	1
4	30	MANAGER	1
5	10	MANAGER	1
6	20	ANALYST	2
7	10	PRESIDENT	1
8	30	CLERK	1
9	10	CLERK	1
10	20	(null)	5
11	30	(null)	6
12	10	(null)	3

-- 7-28. JOB을 먼저 그룹화 한 후 ROLLUP 함수에 DEPTNO 지정하기

```
SELECT DEPTNO, JOB, COUNT(*) FROM EMP

GROUP BY JOB, ROLLUP(DEPTNO);
```

	DEPTNO	JOB	COUNT(*)
1	20	CLERK	2
2	30	SALESMAN	4
3	20	MANAGER	1
4	30	MANAGER	1
5	10	MANAGER	1
6	20	ANALYST	2
7	10	PRESIDENT	1
8	30	CLERK	1
9	10	CLERK	1
10	(null)	CLERK	4
11	(null)	SALESMAN	4
12	(null)	MANAGER	3
13	(null)	ANALYST	2
14	(null)	PRESIDENT	1

-- 7-29. GROUPING SETS 함수를 사용하여 열별로 그룹으로 묶어 출력하기

```
SELECT DEPTNO, JOB, COUNT(*) FROM EMP
GROUP BY GROUPING SETS(DEPTNO, JOB)
ORDER BY DEPTNO, JOB;
```

	DEPTNO	JOB	COUNT(*)
1	10	(null)	3
2	20	(null)	5
3	30	(null)	6
4	(null)	ANALYST	2
5	(null)	CLERK	4
6	(null)	MANAGER	3
7	(null)	PRESIDENT	1
8	(null)	SALESMAN	4

-- 7-30. DEPTNO, JOB 열의 그룹화 결과 여부를 GROUPING 함수로 확인하기

```
SELECT DEPTNO, JOB, COUNT(*), MAX(SAL), SUM(SAL), AVG(SAL),
GROUPING(DEPTNO),
GROUPING(JOB)
FROM EMP
GROUP BY CUBE(DEPTNO, JOB)
ORDER BY DEPTNO, JOB;
```

	DEPTNO	JOB	COUNT(*)	MAX(SAL)	SUM(SAL)	AVG(SAL)	GROUPING(DEPTNO)	GROUPING(JOB)
1	10	CLERK	1	1300	1300	1300	0	0
2	10	MANAGER	1	2450	2450	2450	0	0
3	10	PRESIDENT	1	5000	5000	5000	0	0
4	10	(null)	3	5000	8750	2916.6666666666666666666666666666666666667	0	1
5	20	ANALYST	2	3000	6000	3000	0	0
6	20	CLERK	2	1100	1900	950	0	0
7	20	MANAGER	1	2975	2975	2975	0	0
8	20	(null)	5	3000	10875	2175	0	1
9	30	CLERK	1	950	950	950	0	0
10	30	MANAGER	1	2850	2850	2850	0	0
11	30	SALESMAN	4	1600	5600	1400	0	0
12	30	(null)	6	2850	9400	1566.6666666666666666666666666666666666667	0	1
13	(null)	ANALYST	2	3000	6000	3000	1	0
14	(null)	CLERK	4	1300	4150	1037.5	1	0

-- 7-31. DECODED문으로 GROUPING 함수를 적용하여 결과 표기하기

```
SELECT DECODE(GROUPING(DEPTNO), 1, 'ALL_DEPT', DEPTNO) AS DEPTNO,
       DECODE(GROUPING(JOB), 1, 'ALL_JOB', JOB) AS JOB,
       COUNT(*), MAX(SAL), SUM(SAL), AVG(SAL)
FROM EMP
GROUP BY CUBE(DEPTNO, JOB)
ORDER BY DEPTNO, JOB;
```

	DEPTNO	JOB	COUNT(*)	MAX(SAL)	SUM(SAL)	AVG(SAL)
1	10	ALL_JOB	3	5000	8750	2916.66666666666666666666666666666666667
2	10	CLERK	1	1300	1300	1300
3	10	MANAGER	1	2450	2450	2450
4	10	PRESIDENT	1	5000	5000	5000
5	20	ALL_JOB	5	3000	10875	2175
6	20	ANALYST	2	3000	6000	3000
7	20	CLERK	2	1100	1900	950
8	20	MANAGER	1	2975	2975	2975
9	30	ALL_JOB	6	2850	9400	1566.66666666666666666666666666666666667
10	30	CLERK	1	950	950	950
11	30	MANAGER	1	2850	2850	2850
12	30	SALESMAN	4	1600	5600	1400
13	ALL_DEPT	ALL_JOB	14	5000	29025	2073.214285714285714285714285714285714286
14	ALL_DEPT	ANALYST	2	3000	6000	3000
15	ALL_DEPT	CLERK	4	1300	4150	1037.5
16	ALL_DEPT	MANAGER	3	2975	8275	2758.333333333333333333333333333333333333
17	ALL_DEPT	PRESIDENT	1	5000	5000	5000
18	ALL_DEPT	SALESMAN	4	1600	5600	1400

-- 7-32. DEPTNO, JOB을 함께 명시한 GROUPING_ID 함수 사용하기

```
SELECT DEPTNO, JOB, COUNT(*), SUM(SAL), GROUPING(DEPTNO), GROUPING(JOB),
GROUPING_ID(DEPTNO, JOB) FROM EMP
GROUP BY CUBE(DEPTNO, JOB)
ORDER BY DEPTNO, JOB;
```

	DEPTNO	JOB	COUNT(*)	SUM(SAL)	GROUPING(DEPTNO)	GROUPING(JOB)	GROUPING_ID(DEPTNO,JOB)
1	10	CLERK	1	1300	0	0	0
2	10	MANAGER	1	2450	0	0	0
3	10	PRESIDENT	1	5000	0	0	0
4	10	(null)	3	8750	0	1	1
5	20	ANALYST	2	6000	0	0	0
6	20	CLERK	2	1900	0	0	0
7	20	MANAGER	1	2975	0	0	0
8	20	(null)	5	10875	0	1	1
9	30	CLERK	1	950	0	0	0
10	30	MANAGER	1	2850	0	0	0
11	30	SALESMAN	4	5600	0	0	0
12	30	(null)	6	9400	0	1	1
13	(null)	ANALYST	2	6000	1	0	2
14	(null)	CLERK	4	4150	1	0	2
15	(null)	MANAGER	3	8275	1	0	2
16	(null)	PRESIDENT	1	5000	1	0	2
17	(null)	SALESMAN	4	5600	1	0	2
18	(null)	(null)	14	29025	1	1	3

-- 7-33. GROUP BY절로 그룹화하여 부서 번호와 사원 이름 출력하기

```
SELECT DEPTNO, ENAME
FROM EMP
GROUP BY DEPTNO, ENAME;
```

	DEPTNO	ENAME
1	20	SMITH
2	30	ALLEN
3	30	WARD
4	20	JONES
5	30	MARTIN
6	30	BLAKE
7	10	CLARK
8	20	SCOTT
9	10	KING
10	30	TURNER
11	20	ADAMS
12	30	JAMES
13	20	FORD
14	10	MILLER

-- 7-34. 부서별 사원 이름을 나란히 출력하기

```
SELECT DEPTNO, LISTAGG(ENAME, ', ') WITHIN GROUP(ORDER BY SAL DESC) AS ENAMES
FROM EMP
GROUP BY DEPTNO;
```

	DEPTNO	ENAMES
1	10	KING, CLARK, MILLER
2	20	FORD, SCOTT, JONES, ADAMS, SMITH
3	30	BLAKE, ALLEN, TURNER, MARTIN, WARD, JAMES

-- 7-35. 부서별, 직책별로 그룹화하여 최고 급여 데이터 출력하기

```
SELECT DEPTNO, JOB, MAX(SAL) FROM EMP
GROUP BY DEPTNO, JOB
ORDER BY DEPTNO, JOB;
```

	DEPTNO	JOB	MAX(SAL)
1	10	CLERK	1300
2	10	MANAGER	2450
3	10	PRESIDENT	5000
4	20	ANALYST	3000
5	20	CLERK	1100
6	20	MANAGER	2975
7	30	CLERK	950
8	30	MANAGER	2850
9	30	SALESMAN	1600

-- 7-36. PIVOT 함수를 사용하여 직책별, 부서별 최고 급여를 2차원 표 형태로 출력하기

```
SELECT * FROM (SELECT DEPTNO, JOB, SAL FROM EMP)
PIVOT(MAX(SAL) FOR DEPTNO IN (10,20,30))
ORDER BY JOB;
```

	JOB	10	20	30
1	ANALYST	(null)	3000	(null)
2	CLERK	1300	1100	950
3	MANAGER	2450	2975	2850
4	PRESIDENT	5000	(null)	(null)
5	SALESMAN	(null)	(null)	1600

-- 7-37. PIVOT 함수를 사용하여 부서별, 직책별 최고 급여를 2차원 표 형태로 출력하기

```
SELECT * FROM (SELECT JOB, DEPTNO, SAL FROM EMP)
PIVOT(MAX(SAL) FOR JOB IN('CLERK' AS CLERK,
                          'SALESMAN' AS SALESMAN,
                          'PRESIDENT' AS PRESIDENT,
                          'MANAGER' AS MANAGER,
                          'ANALYST' AS ANALYST))
ORDER BY DEPTNO;
```

	DEPTNO	CLERK	SALESMAN	PRESIDENT	MANAGER	ANALYST
1	10	1300	(null)	5000	2450	(null)
2	20	1100	(null)	(null)	2975	3000
3	30	950	1600	(null)	2850	(null)

-- 7-38. DECODE문을 활용하여 PIVOT 함수와 같은 출력 구현하기

```
SELECT DEPTNO,
        MAX(DECODE(JOB, 'CLERK', SAL)) AS "CLERK",
        MAX(DECODE(JOB, 'SALESMAN', SAL)) AS "SALESMAN",
        MAX(DECODE(JOB, 'PRESIDENT', SAL)) AS "PRESIEDENT",
        MAX(DECODE(JOB, 'MANAGER', SAL)) AS "MANAGER",
        MAX(DECODE(JOB, 'ANALYST', SAL)) AS "ANALYST"
FROM EMP
GROUP BY DEPTNO
ORDER BY DEPTNO;
```

	DEPTNO	CLERK	SALESMAN	PRESIEDENT	MANAGER	ANALYST
1	10	1300	(null)	5000	2450	(null)
2	20	1100	(null)	(null)	2975	3000
3	30	950	1600	(null)	2850	(null)

```
-- 7-39. UNPIVOT 함수를 사용하여 열로 구분된 그룹을 행으로 출력하기
SELECT * FROM (SELECT DEPTNO,
                      MAX(DECODE(JOB, 'CLERK', SAL)) AS "CLERK",
                      MAX(DECODE(JOB, 'SALESMAN', SAL)) AS "SALESMAN",
                      MAX(DECODE(JOB, 'PRESIDENT', SAL)) AS "PRESIDENT",
                      MAX(DECODE(JOB, 'MANAGER', SAL)) AS "MANAGER",
                      MAX(DECODE(JOB, 'ANALYST', SAL)) AS "ANALYST"
               FROM EMP
               GROUP BY DEPTNO
               ORDER BY DEPTNO)
UNPIVOT ( SAL FOR JOB IN (CLERK, SALESMAN, PRESIDENT, MANAGER, ANALYST))
ORDER BY DEPTNO, JOB;
```

	DEPTNO	JOB	SAL
1	10	CLERK	1300
2	10	MANAGER	2450
3	10	PRESIDENT	5000
4	20	ANALYST	3000
5	20	CLERK	1100
6	20	MANAGER	2975
7	30	CLERK	950
8	30	MANAGER	2850
9	30	SALESMAN	1600