PERFORM COMPUTATIONS WITH DATA

PERFORM COMPUTATIONS WITH DATA

산술 및 SQL 함수를 사용하여 다양한 방법으로 데이터를 수정하고 조회한다.

- 숫자 및 날짜 값을 사용하여 계산 수행
- NULL 값을 포함하는 계산 처리
- 숫자, 날짜 및 문자 값을 수정
- 날짜 값을 다양한 방식으로 표시
- 행 그룹에 대한 계산 조회 및 수행

Arithmetic Operators	
+	더하기
-	빼기
*	곱하기
/	나누기

PERFORM COMPUTATIONS WITH NUMBERS

직무가 **'MANAGER'**인 사원에 대해 **$500**의 급여 인상을 계산 후 사원 이름, 급여, 인상된 급여**(NEW SALARY)**를 출력하라.

Example

```
SQL> SELECT ename, sal, sal + 500 "NEW SALARY"
   FROM s_emp
   WHERE job = 'MANAGER'
   ORDER BY  sal + 500;
```

```
ENAME           SAL NEW SALARY
---------- ---------- ----------
CLARK           2450       2950
BLAKE           2850       3350
JONES           2975       3475

3 rows selected.
```

PERFORM COMPUTATIONS WITH NUMBERS

부서 번호가 **20**인 사원에 대해 급여의 **10%**를 보너스를 지급 후, 사원 이름, 급여, 보너스**(BONUS)**를 출력하라.

Example

```
SQL> SELECT ename, sal, sal * 0.1 BONUS
  2 FROM s_emp
  3 WHERE deptno = 20
  4 ORDER BY sal * 0.1;
```

```
ENAME          SAL     BONUS
---------- ---------- ----------
SMITH          800       80
ADAMS          1100      110
JONES          2975     297.5
SCOTT          3000      300
FORD           3000      300

5 rows selected.
```

PERFORM COMPUTATIONS WITH NUMBERS

Rules of Precedence

산술 연산자 우선 순위는 다음과 같다.

1. 곱셈과 나누기 **(*, /)**

2. 덧셈과 빼기 **(+, -)**

Example

부서 번호 **10**인 사원에 대해 연간 보상을 지급한다. 이름, 급여, 연간 보상**(ANNUAL COMPENSATION)**을 출력하라. 연간 보상은 급여에 **$100**를 상여 후 **12**를 곱한다.

```
SQL> SELECT ename, sal, (sal + 100) * 12 "ANNUAL COMPENSATION"
  2 FROM s_emp
  3 WHERE deptno = 10;
```

```
ENAME          SAL ANNUAL COMPENSATION
---------- ---------- -------------------
CLARK          2450              30600
KING           5000              61200

2 rows selected.
```

PERFORM COMPUTATIONS WITH NUMBERS

ROUND

ROUND(*num1*, *num2*) – *num1*을 소수점 아래 *num2* 위치에서 반올림한 값을 반환

Example

직무가 **CLERK**인 사원에게 급여를 근무 일수로 나눈 성과급을 지급한다. 이름, 급여, 성과급**(PERFORMANCE PAY)**을 출력하라. 근무 일수는 **22**일, 성과급은 소수점 둘째 자리에서 반올림 한다.

```
SQL> SELECT ename, sal, ROUND(sal/22, 2)
  2 FROM s_emp
  3 WHERE job = 'CLERK';
```

```
ENAME          SAL ROUND(SAL/22,0)
---------- ---------- ---------------
SMITH         800          36.36
ADAMS         1100          50
JAMES         950          43.18
```

PERFORM COMPUTATIONS WITH NUMBERS

TRUNC

TRUNC(*num1*, *num2*) – *num1*을 소수점 아래 *num2* 위치에서 버림한 값을 반환

Example

부서 번호가 **20**인 사원에게 급여를 근무 일수로 나눈 성과급을 지급한다. 이름, 급여, 성과급**(PERFORMANCE PAY)**을 출력하라. 근무 일수는 **30**일, 성과급은 소수점 둘째 자리에서 버림 한다.

```
SQL> SELECT ename, sal, TRUNC(sal /30, 3)
  2 FROM s_emp
  3 WHERE deptno = 20;
```

```
ENAME           SAL TRUNC(SAL/30,2)
---------- ---------- ---------------
SMITH           800          26.66
JONES          2975          99.16
SCOTT          3000            100
ADAMS          1100          36.66
FORD           3000            100
```

PERFORM COMPUTATIONS WITH NUMBERS

MOD

MOD(*num1, num2*) – *num1*을 *num2*로 나눈 나머지를 반환하는 함수

Example

사원 번호가 짝수인 사원의 정보를 모두 출력하라.

```
SQL> SELECT *
  2 FROM s_emp
  3 WHERE MOD(empno, 2) = 0;
```

```
 EMPNO ENAME      JOB            MGR HIREDATE                                                    SAL       COMM     DEPTNO
---------- ---------- --------- ---------- ---------------------------------------------------------------- ---------- ---------- ----------
      3790 GOODMAN                          2022/10/09                                                  2000
      7566 JONES      MANAGER        7839 1981/04/01                                                  2975                 20
      7654 MARTIN     SALESMAN       7698 1981/09/10                                                  1250       1400        30
      7698 BLAKE      MANAGER        7839 1981/05/01                                                  2850                 30
      7782 CLARK      MANAGER        7839 1981/05/09                                                  2450                 10
      7788 SCOTT      ANALYST        7566 1982/12/22                                                  3000                 20
      7844 TURNER     SALESMAN       7698 1981/08/21                                                  1500          0        30
      7876 ADAMS      CLERK          7788 1983/01/15                                                  1100                 20
      7900 JAMES      CLERK          7698 1981/12/11                                                  950                  30
      7902 FORD       ANALYST        7566 1981/12/11                                                  3000                 20

10 rows selected.
```

PERFORM COMPUTATIONS WITH NUMBERS

NULL

알 수 없는 결측값, 잘못된 값, **NULL**을 포함하는 식을 **NULL**로 나타낸다.

Example

부서 번호가 **30**인 사원에게 급여와 커미션을 더한 금액을 지급한다. 사원 이름, 급여, 커미션, 추가 금액을 출력하라.

```
SQL> SELECT ename, sal, comm, sal + comm
  2 FROM s_emp
  3 WHERE deptno = 30;
```

```
ENAME          SAL      COMM   SAL+COMM
---------- ---------- ---------- ----------
ALLEN          1600       300       1900
WARD           1250       500       1750
MARTIN          1250      1400       2650
BLAKE          2850
TURNER          1500         0       1500
JAMES          950
```

PERFORM COMPUTATIONS WITH NUMBERS

NVL

NVL(*expr1, expr2*) – *expr1*이 NULL이 아니면 *expr1*을 반환하고, *expr1*이 NULL이면 *expr2*를 반환하는 함수

Example

부서 번호가 30인 사원에게 급여와 커미션을 더한 금액을 지급한다. 사원 이름, 급여, 커미션, 추가 금액을 출력하라. 단, 커미션이 없으면 0으로 출력하라

```
SQL> SELECT ename, sal, comm, NVL(sal + comm, 0)
  2 FROM s_emp
  3 WHERE deptno = 30;
```

```
ENAME          SAL      COMM NVL(SAL+COMM,0)
---------- ---------- ---------- ---------------
ALLEN          1600       300          1900
WARD           1250       500          1750
MARTIN          1250      1400          2650
BLAKE          2850                     0
TURNER          1500         0          1500
JAMES          950                      0
```

PERFORM COMPUTATIONS WITH DATES

산술 연산자를 이용해 날짜를 계산할 수 있다.

Date Expression
date + number
date – number
date - date

Example

급여가 **2,000** 이상인 사원의 견습 기간 종료일을 확인한다. 이름, 입사 날짜, 견습 기간 종료일을 출력하라. 견습 기간은 입사 후 **90**일까지이다.

```
SQL> SELECT ename, hiredate, hiredate + 90
  2 FROM s_emp
  3 WHERE sal >= 2000;
```

```
ENAME      HIREDATE                                    HIREDATE+90
---------- ---------------------------------------------------------- ---------------------------------------------------------------
JONES      1981/04/01                                  1981/06/30
BLAKE      1981/05/01                                  1981/07/30
CLARK      1981/05/09                                  1981/08/07
SCOTT      1982/12/22                                  1983/03/22
KING       1981/11/17                                  1982/02/15
FORD       1981/12/11                                  1982/03/11
```

PERFORM COMPUTATIONS WITH DATES

SYSDATE

SYSDATE 함수를 이용해 오늘 날짜를 출력할 수 있다.

Example

부서 번호 **20**인 사원의 이름과 근무 기간을 일, 월로 출력한다. 단, 월은 **30**일로 계산한다

```
SQL> SELECT ename, SYSDATE - hiredate DAYS, (SYSDATE - hiredate) / 30 MONTHS
  2 FROM s_emp
  3 WHERE deptno = 20;
```

```
ENAME          DAYS     MONTHS
---------- ---------- ----------
SMITH      15280.3261 509.344204
JONES      15167.3261 505.577537
SCOTT      14537.3261 484.577537
ADAMS      14513.3261 483.777537
FORD       14913.3261 497.110871

5 rows selected.
```

PERFORM COMPUTATIONS WITH DATES

ADD_MONTHS

ADD_MONTHS(*date, integer*) – date에 integer만큼의 달을 더한 결과를 구하는 함수

Example

직무가 MANAGER인 사원의 이름과 입사 날짜에 6개월을 더한 값을 출력한다.

```
SQL> SELECT ename, ADD_MONTHS(hiredate, 6)
  2 FROM s_emp
  3 WHERE job = 'MANAGER';
```

```
ENAME      ADD_MONTHS(HIREDATE,6)
---------- ------------------------------------------------------------
JONES      1981/10/01
BLAKE      1981/11/01
CLARK      1981/11/09

3 rows selected.
```

PERFORM COMPUTATIONS WITH DATES

LAST_DAY

LAST_DAY(*date*) - 특정 일자에 해당하는 월의 마지막 일자를 표시하는 함수

Example

현재 달의 마지막 일자를 출력하라

```
SQL> SELECT LAST_DAY(SYSDATE)
  2 FROM dual;
```

```
LAST_DAY(SYSDATE)
-----------------------------------------------------------
2022/10/31

1 row selected.
```

NEXT_DAY

NEXT_DAY(*date*, *str*) – date와 가장 가까운 다음 요일 str의 날짜를 반환하는 함수

PERFORM COMPUTATIONS WITH DATES

MONTS_BETWEEN

MONTHS_BETWEEN(*date1*, *date2*) – date1과 date2의 개월 차를 구하는 함수. 두 날짜 사이의 일수를 31로 나눈 값을 반환

Example

직무가 **SALESMAN**인 사원의 이름, 입사일, 근무 개월을 출력하라

```
SQL> SELECT LAST_DAY(SYSDATE)
  2 FROM dual;
```

```
ENAME      HIREDATE          MONTHS_BETWEEN(SYSDATE,HIREDATE)
---------- ----------------------------------------------------------------------------------------

ALLEN      1981/02/11        499.979055

WARD       1981/02/23        499.591958

MARTIN     1981/09/10        493

TURNER     1981/08/21        493.656474


4 rows selected.
```

REFORMAT DATES

날짜 형식

Element	Description	Element	Description
DD	1개월 중 몇 번째 날인지 출력(1- 31)	YYYY	연도를 네 자리로 출력 (2022)
DY	축약 표기한 요일을 출력 (ex : THU)	HH:MI:SS	시간, 분, 초 (10:00:00)
DAY	요일을 출력 (ex : MONDAY)	Q	분기를 출력 (1-4)
MM	달을 출력 (1-12)	HH24	시간을 출력 (0 – 23)
MON	축약된 달 이름을 출력 (ex : DEC)	RM	달을 로마 숫자로 출력 (I-XII)
MONTH	달을 출력한다 (ex : NOVEMBER)	AM or PM	오전 또는 오후
YY	연도를 두 자리로 출력 (22)	TZH or THM	시간대에서 시간 or 분 출력

REFORMAT DATES

TO_CHAR

TO_CHAR(*date_value*, *format_mask*) – date_value를 format_mask 형식에 따라 문자열로 변환 하는 함수

Example

직무가 **MANAGER**인 사원의 이름, 입사일을 **YY**년 **MM**월 **DD**일 형식으로 출력하라

```
SQL> SELECT ename, TO_CHAR(hiredate, 'MM/YY')
   FROM s_emp
   WHERE job =  'MANAGER';
```

```
ENAME      HIREDATE           TO_CHAR(HIREDATE,'YY"년"MM"월"DD"일"')
---------- -------------------- ----------------------------------------
JONES      1981/04/01          81 년 04 월 01
BLAKE      1981/05/01          81 년 05 월 01
CLARK      1981/05/09          81 년 05 월 09

3 rows selected.
```

REFORMAT DATES

TO_DATE

TO_CHAR(*character_value*, *format_mask*) – character_value를 format_mask 형식에 따라 Date로 변환 하는 함수

Example

S_EMP 테이블에 신입사원을 추가한다. 입사일은 900708이다

```
SQL> INSERT INTO s_emp(empno, ename, hiredate, deptno)
  2 VALUES (8371, 'MARU', TO_DATE('19900708', 'YYYYMMDD'), 40);

1 row inserted.
```

```
 EMPNO ENAME      JOB          MGR HIREDATE              SAL    COMM     DEPTNO
---------- ---------- --------- ---------- -------------------- ---------- ---------- -------------------- ----------
8371       MARU                            1990/07/08                                                      40

1 row selected.
```

MANIPULATE CHARACTER STRINGS

||

접합 연산자로 문자열을 결합 시킨다.

Example

부서 번호 **20**인 사원의 사원 번호와 이름을 합쳐 출력한다.

```
SQL> SELECT empno || ' ' || ename "ID AND EMPLOYEE"
  2 FROM s_emp
  3 WHERE deptno = 20;
```

```
ID AND EMPLOYEE
--------------------------------------------------------------------------------
7369 SMITH
7566 JONES
7788 SCOTT
7876 ADAMS
7902 FORD

5 rows selected.
```

MANIPULATE CHARACTER STRINGS

문자 함수

NAME : Delhi Sports

Function	EXAMPLE	RESULT
INITCAP	INITCAP (NAME)	Delhi Sports
UPPER	UPPER (NAME)	DELHI SPORTS
LOWER	LOWER (NAME)	delhi sports
SUBSTR	SUBSTR (NAME, 1, 4)	Delh
LENGTH	LENGTH (NAME)	12

MANIPULATE CHARACTER STRINGS

INITCAP

INITCAP – 첫 글자만 대문자로 변환한다.

Example

직무가 **CLERK**인 사원의 이름, 직무를 출력한다. 이름은 첫 글자는 대문자, 나머지는 소문자로 출력한다.

```
SQL> SELECT INITCAP(ename) NAME, job
  2 FROM s_emp
  3 WHERE job = 'CLERK';
```

```
NAME       JOB
---------- ---------
Smith      CLERK
Adams      CLERK
James      CLERK

3 rows selected.
```

MANIPULATE CHARACTER STRINGS

UPPER

UPPER - 모든 문자를 대문자로 변환한다.

Example

INITCAP 함수를 이용해 출력한 직원 이름을 UPPER 함수를 이용해 대문자로 변환한다.

```
SQL> SELECT UPPER(INITCAP(ename)) NAME, job
  2 FROM s_emp
  3 WHERE job = 'CLERK';
```

```
NAME       JOB
---------- ---------
SMITH      CLERK
ADAMS      CLERK
JAMES      CLERK

3 rows selected.
```

MANIPULATE CHARACTER STRINGS

LOWER

LOWER - 모든 문자를 소문자로 변환한다.

Example

부서 번호 **30**인 직원 이름, 직무를 출력한다. **LOWER** 함수를 이용해 모두 소문자로 변환한다.

```
SQL> SELECT LOWER(ename), LOWER(job)
  2 FROM s_emp
  3 WHERE deptno = 30;
```

```
LOWER(ENAME) LOWER(JOB)
------------ ----------
allen        salesman
ward         salesman
martin       salesman
blake        manager
turner       salesman
james        clerk

6 rows selected.
```

MANIPULATE CHARACTER STRINGS

SUBSTR

SUBSTR(*char*, *m* [, *n*] – char 내 m 번째 위치로부터 n 길이의 문자열을 추출하는 함수. n이 지정되지 않으면 마지막까지 추출

Example

부서 번호 10인 직원 이름을 뒤에서 첫 번째에서 2개까지 출력한다.

```
SQL>  SELECT ename, SUBSTR(ename, 1, 2)
  2  FROM s_emp
  3 WHERE deptno = 10;
```

```
ENAME      SUBSTR(ENAME,1,2)
---------- -----------------
CLARK      CL
KING       KI

2 rows selected.
```

MANIPULATE CHARACTER STRINGS

LENGTH

LENGTH – 문자열의 길이를 반환하는 함수

Example

부서 번호 **30**인 직원 이름과 이름의 길이를 출력하라.

```
SQL> SELECT ename, LENGTH(ename)
  2 FROM s_emp
  3 WHERE deptno = 30;
```

```
ENAME      LENGTH(ENAME)
---------- -------------
ALLEN               5
WARD                4
MARTIN               6
BLAKE               5
TURNER               6
JAMES               5
```

이름 길이가 **6**이상인 직원의 이름을 출력하라

```
SQL> SELECT ename
  2 FROM s_emp
  3 WHERE LENGTH(ename) >= 6;
```

```
ENAME
----------
MARTIN
TURNER
```

PERFORM SUMMARY COMPUTATIONS

문자 함수

NAME : Delhi Sports

Function	Description
AVG	평균값
MAX	최댓값
MIN	최솟값
SUM	합
COUNT	로우의 개수를 세는 함수

MANIPULATE CHARACTER STRINGS

AVG SUM
MIN MAX

Example

직무가 SALESMAN 인 사원들의 급여 평균(AVERAGE), 최댓값(MAXIMUM), 최솟값(MINIMUM), 합(SUM)을 출력하라

```
SQL> COLUMN average FORMAT $99,999.99
SQL> COLUMN maximum FORMAT $99,999.99
SQL> COLUMN minimum FORMAT $99,999.99
SQL> COLUMN sum FORMAT $99,999.99
SQL> SELECT AVG(sal) average, MAX(sal) maximum, MIN(sal) minimum, SUM(sal) sum
  2 FROM s_emp
  3 WHERE job = 'SALESMAN';
```

```
 AVERAGE     MAXIMUM     MINIMUM        SUM
----------- ----------- ----------- -----------
  $1,400.00   $1,600.00   $1,250.00   $5,600.00

1 row selected.
```

MANIPULATE CHARACTER STRINGS

COUNT

Example

S_EMP 테이블에 있는 총 사원의 수를 구하라

```
SQL> SELECT COUNT(*)
  2 FROM s_emp;
```

```
 COUNT(*)
----------
        14

1 row selected.
```

커미션을 받는 사원의 수를 구하라

```
SQL> SELECT COUNT(comm) "Employees with Comm"
  2 FROM s_emp;
```

```
Employees with Comm
-------------------
                  4

1 row selected.
```

GROUP ROWS TOGETHER

부서 번호가 30인 사원은 6명으로 6번 표시가 된다. **GROUP BY** 절을 사용하면 각 부서에 대해 한 줄로 출력할 수 있다.

```
SQL> SELECT empno, ename, deptno
  2 FROM s_emp
  3 WHERE deptno = 30;

 EMPNO ENAME          DEPTNO
---------- ---------- ----------
      7499 ALLEN            30
      7521 WARD              30
      7654 MARTIN             30
      7698 BLAKE            30
      7844 TURNER             30
      7900 JAMES            30

6 rows selected.
```

```
SQL> SELECT deptno, COUNT(*) NUBMER
    FROM s_emp
    WHERE deptno = 30
    GROUP BY deptno;

 DEPTNO     NUBMER
---------- ----------
        30          6

1 row selected.
```

GROUP ROWS TOGETHER

GROUP BY 및 HAVING 절이 있는 행 그룹에 대해 요약 결과를 출력한다.

Syntax

```
SELECT column_name
FROM table_name
WHERE condition
GROUP BY group_by_expression
```

where	*group_by_expression*	그룹화가 되는 기준의 열을 지정

Example

S_EMP 테이블에 있는 직무를 출력한다.

```
SQL> SELECT job, COUNT(*) "Number"
  2 FROM s_emp
  3 GROUP BY job;
```

```
JOB         Number
--------- ----------
CLERK            3
ANALYST          2
PRESIDENT        1
SALESMAN         4
MANAGER          3
```

GROUP ROWS TOGETHER

Example

부서 번호에 따른 직원 수 출력

```
SQL> SELECT deptno, COUNT(*) "Head Count"
  2 FROM s_emp
  3 GROUP BY deptno;
```

```
 DEPTNO Head Count
---------- ----------
        10          2
        40          1
        20          5
        30          6
```

GROUP BY 절 없이 정규 열과 그룹 함수를 함께 사용하면 안된다.

```
SQL> SELECT deptno, COUNT(*) "Employees Within Titles"
  2  FROM s_emp;
TBR-8038: Expression is not in a GROUP BY clause.
```

GROUP ROWS TOGETHER

둘 이상의 **GROUP BY** 칼럼을 나열하여 그룹 및 하위 그룹에 대한 결과를 출력한다.

Example

```
SQL> SELECT deptno, job, COUNT(*) "Employees Within Titles"
  2 FROM s_emp
  3 WHERE deptno = 30
  4 GROUP BY deptno, job;
```

```
 DEPTNO JOB       Employees Within Titles
---------- --------- -----------------------
        30 CLERK                          1
        30 SALESMAN                       4
        30 MANAGER                        1
```

```
SQL> SELECT  job, deptno, COUNT(*) "Employees Within Titles"
   2 FROM s_emp
   3 WHERE deptno = 30
   4 GROUP BY job, deptno ;
```

```
JOB          DEPTNO Employees Within Titles
--------- ---------- -----------------------
CLERK            30                        1
SALESMAN         30                        4
MANAGER          30                        1
```

DISPLAY SPECIFIC GROUPS

특정 행 또는 특정 그룹 출력

Syntax

```
SELECT column_name [,column_name]
FROM table_name
WHERE condition
GROUP BY group_by_expression
HAVING condition
```

where	*condition*	지정된 조건이 참(TRUE)인 그룹만 반환

DISPLAY SPECIFIC GROUPS

직원이 세 명 이상인 부서의 급여 평균과 직원 수를 출력하라

Example

```
SQL> SELECT deptno, AVG(sal) average, COUNT(*) "Number of Employees"
  2 FROM s_emp
  3 GROUP BY deptno
  4 HAVING COUNT(*) >= 3;
```

```
 DEPTNO     AVERAGE Number of Employees
---------- ----------- -------------------
      20   $2,175.00                 5
      30   $1,566.67                 6

2 rows selected.
```

DISPLAY SPECIFIC GROUPS

각 직무 별 급여의 합이 **5,000**보다 큰 직무와 급여 합을 출력하라. 급여의 합은 내림차순으로 정렬

Example

```
SQL> SELECT job, SUM(sal) sum
  2 FROM s_emp
  3 GROUP BY job
  4 HAVING SUM(sal) > 5000
  5 ORDER BY SUM(sal) DESC;
```

```
JOB             SUM
--------- -----------
MANAGER     $8,275.00
ANALYST     $6,000.00
SALESMAN    $5,600.00

3 rows selected.
```