

Funciones analíticas VARIANCE, VAR_POP y VAR_SAMP

by admin - viernes, octubre 02, 2020

https://dbandtech.com/funciones-analiticas-variance-var_pop-y-var_samp/

Este artículo ofrece una descripción general de las funciones analíticas **VARIANCE**, **VAR_POP** y **VAR_SAMP**. Si es nuevo en las funciones analíticas, probablemente debería leer primero esta introducción a las funciones analíticas.

- [Preparar](#)
- [VARIANCE, VAR_POP y VAR_SAMP como funciones agregadas](#)
- [Función analítica VARIANCE](#)
- [Función analítica VAR_POP](#)
- [Función analítica VAR_SAMP](#)
- [Enlaces rapido](#)

Preparar

Los ejemplos de este artículo requieren la siguiente tabla.

```
--DROP TABLE emp PURGE;
```

```
CREATE TABLE emp (  
empno    NUMBER(4) CONSTRAINT pk_emp PRIMARY KEY,  
ename    VARCHAR2(10),  
job      VARCHAR2(9),  
mgr      NUMBER(4),  
hiredate DATE,  
sal      NUMBER(7,2),  
comm     NUMBER(7,2),  
deptno   NUMBER(2)  
);  
INSERT INTO emp VALUES (7369, 'SMITH', 'CLERK', 7902, to_date('17-12-1980'  
, 'dd-mm-yyyy'), 800, NULL, 20);  
INSERT INTO emp VALUES (7499, 'ALLEN', 'SALESMAN', 7698, to_date('20-2-1981'  
, 'dd-mm-yyyy'), 1600, 300, 30);  
INSERT INTO emp VALUES (7521, 'WARD', 'SALESMAN', 7698, to_date('22-2-1981'  
, 'dd-mm-yyyy'), 1250, 500, 30);  
INSERT INTO emp VALUES (7566, 'JONES', 'MANAGER', 7839, to_date('2-4-1981'  
, 'dd-mm-yyyy'), 2975, NULL, 20);  
INSERT INTO emp VALUES (7654, 'MARTIN', 'SALESMAN', 7698, to_date('28-9-1981'  
, 'dd-mm-yyyy'), 1250, 1400, 30);
```

```
INSERT INTO emp VALUES (7698, 'BLAKE', 'MANAGER', 7839, to_date('1-5-1981',
, 'dd-mm-yyyy'), 2850, NULL, 30);
INSERT INTO emp VALUES (7782, 'CLARK', 'MANAGER', 7839, to_date('9-6-1981',
, 'dd-mm-yyyy'), 2450, NULL, 10);
INSERT INTO emp VALUES (7788, 'SCOTT', 'ANALYST', 7566, to_date('13-JUL-87',
, 'dd-mm-rr')-85, 3000, NULL, 20);
INSERT INTO emp VALUES (7839, 'KING', 'PRESIDENT', NULL, to_date('17-11-1981',
, 'dd-mm-yyyy'), 5000, NULL, 10);
INSERT INTO emp VALUES (7844, 'TURNER', 'SALESMAN', 7698, to_date('8-9-1981',
, 'dd-mm-yyyy'), 1500, 0, 30);
INSERT INTO emp VALUES (7876, 'ADAMS', 'CLERK', 7788, to_date('13-JUL-87',
, 'dd-mm-rr')-51, 1100, NULL, 20);
INSERT INTO emp VALUES (7900, 'JAMES', 'CLERK', 7698, to_date('3-12-1981',
, 'dd-mm-yyyy'), 950, NULL, 30);
INSERT INTO emp VALUES (7902, 'FORD', 'ANALYST', 7566, to_date('3-12-1981',
, 'dd-mm-yyyy'), 3000, NULL, 20);
INSERT INTO emp VALUES (7934, 'MILLER', 'CLERK', 7782, to_date('23-1-1982',
, 'dd-mm-yyyy'), 1300, NULL, 10);
COMMIT;
```

VARIANCE, VAR_POP y VAR_SAMP como funciones agregadas

Las funciones agregadas VARIANCE, VAR_POP y VAR_SAMP se utilizan para calcular la varianza, la varianza de la población y la varianza de la muestra de un conjunto de datos, respectivamente. Como funciones agregadas, reducen el número de filas, de ahí el término "aggregate". Si los datos no están agrupados, convertimos las 14 filas de la tabla EMP en una sola fila con los valores agregados.

```
SELECT VARIANCE(sal) AS variance_sal,
       VAR_POP(sal) AS var_pop_sal,
       VAR_SAMP(sal) AS var_samp_sal FROM emp;
VARIANCE_SAL VAR_POP_SAL VAR_SAMP_SAL
-----
1398313.87 1298434.31 1398313.87
SQL>
```

Podemos obtener más granularidad de la información al incluir una cláusula GROUP BY. En el siguiente ejemplo, vemos los valores por departamento.

```
SELECT deptno,
       VARIANCE(sal) AS variance_sal,
       VAR_POP(sal) AS var_pop_sal,
```

```

VAR_SAMP(sal) AS var_samp_sal
FROM emp GROUP BY deptno ORDER BY deptno;
DEPTNO VARIANCE_SAL VAR_POP_SAL VAR_SAMP_SAL
-----
10      3585833.33    2390555.56    3585833.33
20         1261875      1009500        1261875
30      446666.667    372222.222    446666.667
SQL>

```

En ambos casos, hemos agregado los datos para obtener los valores, devolviendo menos filas de las que comenzamos. Las funciones analíticas nos permiten devolver estos valores agregados conservando los datos de la fila original.

Función analítica VARIANCE

Si hay más de un registro en la muestra después de descartar los valores nulos, la función VARIANCE devuelve el resultado de la función VAR_SAMP, la varianza de la muestra. Si solo hay una fila en la muestra después de descartar los valores nulos, la función VARIANCE devuelve el valor "0". Si no hay registros en el conjunto después de descartar los valores nulos, el valor devuelto es NULL.

La descripción básica de la función analítica VARIANCE se muestra a continuación. Order_by_clause y windowing_clause no están permitidos. La cláusula analítica se describe con más detalle aquí.

```
VARIANCE([ DISTINCT | ALL ] expr) [ OVER (analytic_clause) ]
```

El uso de una cláusula OVER vacía convierte la función VARIANCE en una función analítica. La falta de una cláusula de partición significa que todo el conjunto de resultados se trata como una sola partición, por lo que obtenemos la variación del salario de todos los empleados, así como todos los datos originales.

```

SELECT empno ,
       ename ,
       deptno ,
       sal ,
       VARIANCE(sal) OVER () AS variance_sal
FROM   emp
ORDER BY deptno;
EMPNO ENAME          DEPTNO          SAL VARIANCE_SAL
-----
7782 CLARK            10              2450 1398313.87
7839 KING             10              5000 1398313.87
7934 MILLER          10              1300 1398313.87
7566 JONES           20              2975 1398313.87
7902 FORD             20              3000 1398313.87
7876 ADAMS           20              1100 1398313.87

```

```

7369 SMITH          20          800    1398313.87
7788 SCOTT          20         3000    1398313.87
7521 WARD           30         1250    1398313.87
7844 TURNER         30         1500    1398313.87
7499 ALLEN          30         1600    1398313.87
7900 JAMES          30          950    1398313.87
7698 BLAKE          30         2850    1398313.87
7654 MARTIN         30         1250    1398313.87
    
```

SQL>

Agregar la cláusula de partición nos permite mostrar la variación del salario por departamento, junto con los datos de los empleados para cada departamento.

```

SELECT empno ,
       ename ,
       deptno ,
       sal ,
       VARIANCE(sal) OVER (PARTITION BY deptno) AS variance_sal_by_dept
FROM emp;
    
```

EMPNO	ENAME	DEPTNO	SAL	VARIANCE_SAL_BY_DEPT
7782	CLARK	10	2450	3585833.33
7839	KING	10	5000	3585833.33
7934	MILLER	10	1300	3585833.33
7566	JONES	20	2975	1261875
7902	FORD	20	3000	1261875
7876	ADAMS	20	1100	1261875
7369	SMITH	20	800	1261875
7788	SCOTT	20	3000	1261875
7521	WARD	30	1250	446666.667
7844	TURNER	30	1500	446666.667
7499	ALLEN	30	1600	446666.667
7900	JAMES	30	950	446666.667
7698	BLAKE	30	2850	446666.667
7654	MARTIN	30	1250	446666.667

SQL>

Función analítica VAR_POP

La función VAR_POP devuelve la varianza de la población después de descartar los valores nulos. La función realiza el siguiente cálculo.

```
SUM((expr - (SUM(expr) / COUNT(expr)))2) / COUNT(expr)
```

La descripción básica de la función analítica VAR_POP se muestra a continuación.

```
VAR_POP(expr) [ OVER (analytic_clause) ]
```

El uso de una cláusula OVER vacía convierte la función VAR_POP en una función analítica. La falta de una cláusula de partición significa que todo el conjunto de resultados se trata como una sola partición, por lo que obtenemos la variación de población del salario de todos los empleados después de descartar los valores nulos, así como todos los datos originales.

```
SELECT empno ,
       ename ,
       deptno ,
       sal ,
       VAR_POP(sal) OVER ( ) AS var_pop_salFROM emp;
EMPNO ENAME          DEPTNO          SAL  VAR_POP_SAL
-----
7369 SMITH            20              800  1298434.31
7499 ALLEN            30             1600  1298434.31
7521 WARD              30             1250  1298434.31
7566 JONES             20             2975  1298434.31
7654 MARTIN            30             1250  1298434.31
7698 BLAKE             30             2850  1298434.31
7782 CLARK             10             2450  1298434.31
7788 SCOTT             20             3000  1298434.31
7839 KING              10             5000  1298434.31
7844 TURNER            30             1500  1298434.31
7876 ADAMS             20             1100  1298434.31
7900 JAMES             30              950  1298434.31
7902 FORD              20             3000  1298434.31
7934 MILLER            10             1300  1298434.31
SQL>
```

Agregar la cláusula de partición nos permite mostrar la variación de la población del salario por departamento, junto con los datos de los empleados para cada departamento.

```
SELECT empno ,
       ename ,
       deptno ,
```

```
    sal ,
    VAR_POP(sal) OVER (PARTITION BY deptno) AS var_pop_by_deptFROM
emp;
```

EMPNO	ENAME	DEPTNO	SAL	VAR_POP_BY_DEPT
7782	CLARK	10	2450	2390555.56
7839	KING	10	5000	2390555.56
7934	MILLER	10	1300	2390555.56
7566	JONES	20	2975	1009500
7902	FORD	20	3000	1009500
7876	ADAMS	20	1100	1009500
7369	SMITH	20	800	1009500
7788	SCOTT	20	3000	1009500
7521	WARD	30	1250	372222.222
7844	TURNER	30	1500	372222.222
7499	ALLEN	30	1600	372222.222
7900	JAMES	30	950	372222.222
7698	BLAKE	30	2850	372222.222
7654	MARTIN	30	1250	372222.222

SQL>

Función analítica VAR_SAMP

La función VAR_SAMP devuelve la varianza de la muestra después de descartar los valores nulos. Si no hay registros en el conjunto después de descartar los valores nulos, el valor devuelto es NULL. La función realiza el siguiente cálculo.

$$\frac{(\text{SUM}(\text{expr} - (\text{SUM}(\text{expr}) / \text{COUNT}(\text{expr})))^2)}{(\text{COUNT}(\text{expr}) - 1)}$$

La descripción básica de la función analítica VAR_SAMP se muestra a continuación.

```
VAR_SAMP(expr) [ OVER (analytic_clause) ]
```

El uso de una cláusula OVER vacía convierte la función VAR_SAMP en una función analítica. La falta de una cláusula de partición significa que todo el conjunto de resultados se trata como una sola partición, por lo que obtenemos la varianza de muestra del salario de todos los empleados después de descartar los valores nulos, así como todos los datos originales.

```
SELECT empno ,
       ename ,
```

```

deptno,
sal,
VAR_SAMP(sal) OVER () AS var_samp_salFROM emp;
EMPNO ENAME          DEPTNO          SAL VAR_SAMP_SAL
-----
7369 SMITH            20              800 1398313.87
7499 ALLEN            30             1600 1398313.87
7521 WARD             30             1250 1398313.87
7566 JONES            20             2975 1398313.87
7654 MARTIN           30             1250 1398313.87
7698 BLAKE            30             2850 1398313.87
7782 CLARK            10             2450 1398313.87
7788 SCOTT            20             3000 1398313.87
7839 KING             10             5000 1398313.87
7844 TURNER           30             1500 1398313.87
7876 ADAMS            20             1100 1398313.87
7900 JAMES            30              950 1398313.87
7902 FORD             20             3000 1398313.87
7934 MILLER           10             1300 1398313.87

```

SQL>

Agregar la cláusula de partición nos permite mostrar la variación de muestra del salario por departamento, junto con los datos de los empleados para cada departamento.

```

SELECT empno,
       ename,
       deptno,
       sal,
       VAR_SAMP(sal) OVER (PARTITION BY deptno) AS var_samp_by_deptFROM
emp;
EMPNO ENAME          DEPTNO          SAL VAR_SAMP_BY_DEPT
-----
7782 CLARK            10             2450 3585833.33
7839 KING             10             5000 3585833.33
7934 MILLER           10             1300 3585833.33
7566 JONES            20             2975 1261875
7902 FORD             20             3000 1261875
7876 ADAMS            20             1100 1261875
7369 SMITH            20              800 1261875
7788 SCOTT            20             3000 1261875
7521 WARD             30             1250 446666.667
7844 TURNER           30             1500 446666.667
7499 ALLEN            30             1600 446666.667
7900 JAMES            30              950 446666.667

```

7698	BLAKE	30	2850	446666.667
7654	MARTIN	30	1250	446666.667

SQL>

Enlaces rapido

Para más información, ver:

- [VARIANCE](#)
- [VAR_POP](#)
- [VAR_SAMP](#)

PDF generated by Kalin's PDF Creation Station