



COLLATION_KEY

Contents

COLLATION_KEY.	1
------------------------	---

COLLATION_KEY

The COLLATION_KEY function returns a VARBINARY string that represents the collation key of the expression argument, in the specified collation.

➡➡COLLATION_KEY(—*string-expression*—,—*collation-name*— length—)➡➡

The schema is SYSIBM.

The results of COLLATION_KEY for two strings can be binary compared to determine their order within the specified *collation-name*. For the comparison to be meaningful, the results that are used must be from the same *collation-name*.

string-expression

An expression for which the collation key is determined. The expression must return a value that is a built-in character string, graphic string, numeric, or datetime data type. Numeric and datetime data types are supported through implicit casting. The expression must not be a FOR BIT DATA subtype (SQLSTATE 429BM). If the expression is a CLOB, numeric, or datetime data type, the expression is cast to VARCHAR before the function is evaluated. If the expression is a DBCLOB, it is cast to VARGRAPHIC before the function is evaluated. If *string-expression* is not in UTF-16, this function converts the code page of *string-expression* to UTF-16. If the result of the code page conversion contains at least one substitution character, this function returns a collation key of the UTF-16 string with the substitution character or characters. In such cases, the warning flag SQLWARN8 in the SQLCA is set to 'W'.

collation-name

An expression that specifies the collation to use when the collation key is determined. The expression must return a value that is a CHAR or VARCHAR. In a Unicode database, the expression can also be a GRAPHIC or VARGRAPHIC. The expression must be a constant (SQLSTATE 428I9). The value of *collation-name* is not case-sensitive and must be one of the Unicode Collation Algorithm-based collations or language-aware collations for Unicode data (SQLSTATE 42704).

length

An expression that specifies the length attribute of the result in bytes. The expression must return a value that is a built-in numeric data type, CHAR, or VARCHAR data type. In a Unicode database, the expression can also be a GRAPHIC or VARGRAPHIC data type. CHAR, VARCHAR, GRAPHIC, and VARGRAPHIC are supported through implicit casting. If the expression is not an INTEGER, it is cast to INTEGER before the function is evaluated. The value must be 1 - 32 672 (SQLSTATE 428I5). The expression must be a constant (SQLSTATE 428I9).

If a value for *length* is not specified, the length of the result is determined as described in the following table:

Table 1. Determining the result length

Data type of <i>string-expression</i>	Result data type length
CHAR(<i>n</i>) or VARCHAR(<i>n</i>)	Minimum of 12 <i>n</i> bytes and 32 672 bytes

Table 1. Determining the result length (continued)

Data type of <i>string-expression</i>	Result data type length
GRAPHIC(<i>n</i>) or VARCHARGRAPHIC(<i>n</i>)	Minimum of 12 <i>n</i> bytes and 32 672 bytes

Regardless of whether *length* is specified, if the length of the collation key is longer than the length of the result data type, an error is returned (SQLSTATE 42815). The actual result length of the collation key is approximately six times the length of *string-expression* after it is converted to UTF-16.

If *string-expression* is an empty string, the result is a valid collation key that can have a nonzero length.

If any argument can be null, the result can be null; if any argument is null, the result is the null value.

Examples

1. The following query orders employees by their surnames by using the language-aware collation for German in code page 923:

```
SELECT FIRSTNAME, LASTNAME
FROM EMPLOYEE
ORDER BY COLLATION_KEY (LASTNAME, 'SYSTEM_923_DE')
```

2. The following query uses a culturally correct comparison to find the departments of employees in the province of Québec:

```
SELECT E.WORKDEPT
FROM EMPLOYEE AS E INNER JOIN SALES AS S
ON COLLATION_KEY(E.LASTNAME, 'CLDR181_LFR') =
   COLLATION_KEY(S.SALES_PERSON, 'CLDR181_LFR')
WHERE S.REGION = 'Quebec'
```