DATA_MASK scalar function

Last Updated: 2024-10-30

9.12

You can use the DATA_MASK built-in scalar function to do partial redaction, format preserving masking, strong cryptographic obfuscation, and integrity preserving tokenization. The function produces masked data that can be used by analytics tools to create insights without revealing sensitive data. You use the DATA_MASK function in the masking definition of a CREATE MASK statement.

> **Note:** This scalar function is currently available on Db2® 11.5.9 and later for the following 64-bit Linux platforms:
> – AMD (linuxamd64).
> – Z Linux (linux390x64).
> – Power PC (linuxppcle64).

```
▸▸─ DATA_MASK ── ( ── expression ── , ── mask-type ── , ── mask-parameters ── , ── mask-format ── , ── seed ── ) ─▸◂
```

The schema is SYSIBM.

DATA_MASK

The data type of the returned value is the same as the data type of the expression.

The DATA_MASK scalar function supports only Unicode databases.

> **Note:** The masked output produced by the DATA_MASK scalar function might change as the masking algorithms evolve and the underlying **securehash** function is upgraded or changed.

expression

An expression that returns a value of any built-in data type that is not CLOB, BLOB, DBCLOB, VARGRAPHIC, GRAPHIC, and XML.

mask-type

An expression that specifies masking operation to be performed. See Table 1 for available masking operations. The expression must return a value that is a built-in INTEGER.

> **Note:** If the value is not of type INTEGER, it is implicitly cast to INTEGER before evaluating the function.

Table 1. Valid mask types

INT value	Masking Name	Target Data-Type	Description
9.12 0	REDACT	VARCHAR & CHAR. All other data-types redacted to a default value	Performs full redaction on input with the string value supplied by the `mask-parameters` attribute.
1	REDACT PRESERVE LENGTH	VARCHAR & CHAR. All other data-types redacted to default value.	First character in `mask-parameters` is a redaction character. Masked output is a redaction character repeated for the length of the input value.
2 9.12	SUBSTITUTE	All supported data-types	For VARCHAR and CHAR, seeded secureHash(SHA-256) and following base64 encoding is done. Column length must be at least 43 bytes to avoid collisions. For all other data types, obfuscation is performed. If a seed value is not provided, the default redaction value for the data type is returned.
3 9.12	PARTIAL REDACT	VARCHAR & CHAR. All other data-types redacted to default corresponding value	Advance redaction using patterns that are supplied as regular expressions or a string slice expression by the `mask-parameters` attribute.

INT value	Masking Name	Target Data-Type	Description
4	FORMAT PRESERVING OBFUSCATION	All supported data-types	Masking by obfuscation with input validation. The output format matches the original input format that is specified by the `mask-format` attribute. See Table 4 for all the valid format available. The `mask-format` value is not to be used for `DATE` and `TIMESTAMP` data types. If the input value does not match a known value in the dictionary of the format that is specified by the `mask-format` attribute, the value is redacted with the default value. If the `seed` attribute is empty or null, a random `seed` value is used.
5	DATE AGING	`DATE` & `TIMESTAMP`. All other data-types redacted to default corresponding value	Masks input by aging it. `mask-parameters` controls how the date is aged.
7	IDENTIFIER	STRING & NUMERIC data-types . All other data-types redacted to default corresponding value	Characters are replaced with characters, and digits are replaced with digits. All other characters do not change. If the `seed` attribute is empty or null, a random `seed` value is used.
10	FORMAT PRESERVING FABRICATION	All supported data-types	Same value as the `FORMAT PRESERVING OBFUSCATION` value that is masking by obfuscation but without input validation. If the `seed` attribute is empty or null, a random `seed` value is used.
11	NUMERIC SHIFT	Only for numerical data-types	Masks the input by shifting it. The `mask-parameters` attribute contains a double value that is used as a percentage by which to shift the input value.

Note: For obfuscation and substitution operations involving NULL input values, fabrication and substitution is based on a random input that produces random output.

Table 2. Default values

data-type	Default Value
Numeric data type	`0` *(Note for different numeric data types `0` is represented differently)*
String data type	If column length is less than 10 characters then entire string is redacted with X up to the column length, otherwise returns XXXXXXXXXX
`DATE`	`2001-01-01`
`TIMESTAMP`	`2001-01-01-00.00.00.000000`
`TIME`	`00.00.00`
`BOOLEAN`	`0`

mask-parameters

The expression must return a value that is either a built-in CHAR or VARCHAR data type.

Table 3. mask-type and valid mask-parameters

mask-type	Valid mask-parameters	Example
REDACT	The string value to which the input is to be redacted.	`'*'` or `'X'` or `'XXX'`
REDACT PRESERVE LENGTH	Represents the redaction character. If the string is more than one character, then only the first character is used.	`'*'` or `'X'`

mask-type	Valid mask-parameters	Example
PARTIAL REDACT	A regular expression or slice expression that identifies the parts of input to be redacted. Also identifies the character to be used for redaction	`'SLC=0,3,0,X;SLC=0,4,1,X'` or `'"RGX=^\\d{3}[- ]?(\\d{2})##char=X;SLC=0,3,0'`
FORMAT PRESERVING OBFUSCATION	Only for DATE and TIMESTAMP data types. The value represents a masked output range or time frame.	`'minDate=2001-01-01;maxDate=2099-12-31'`, `'maxDate=2089-12-31'`, `'same=quarter'`, `'same=week'`, `'same=year'` or `'same=month'`
DATE AGING	Controls how the date value is aged. It should be in the format "days=+/-"	`'days=1'` or `'days=-5'`
NUMERIC SHIFT	Contains an integer value that is used as a percentage by which to shift the input value.	`'5'` or `'-65'`
FORMAT PRESERVING FABRICATION	Only for DATE and TIMESTAMP data types. The value represents a masked output range or time frame	`'minDate=2001-01-01;maxDate=2099-12-31'`, `'maxDate=2089-12-31'`, `'same=quarter'`, `'same=week'`, `'same=year'` or `'same=month'`

mask-format

The expression must return a value that is either a built-in CHAR or VARCHAR data type. It should be one of the format from table 4 below that representing the expression.

The following valid formats for FORMAT PRESERVING OBFUSCATION and FORMAT PRESERVING FABRICATION Obfuscation are supported.

A B C D E F G H I J K L M N O P Q R S T U V W

A

- AirportCodes
- AlabamaStateDriversLicense
- AlaskaStateDriversLicense
- AlbertaProvinceDriversLicense
- AmericanExpressCard
- ArizonaStateDriversLicense
- ArkansasStateDriversLicense

B

- BIC
- BritishColumbiaProvinceDriversLicense

C

- CaliforniaStateDriversLicense
- CageCode
- CanadianSIN
- CaPostalCode
- CAProvinceCodes
- CAProvinceNames
- CitiesEnglish
- ColoradoStateDriversLicense
- ConnecticutStateDriversLicense
- CountryCodes_CaseSensitive
- Countries_English CreditCard

D

- Date
- Datetime
- DelawareStateDriversLicense
- DiscoverCard
- DinersClubCard
- DinersClubInternationalCard
- DinersClubUSACanadaCard

- DUNS

E

- EmploymentStatus
- EnglishEmailCaseSensitive
- EnglishNIN
- Ethnicity
- EyeColors

F

- FloridaStateDriversLicense
- Fortune1000
- FrenchINSEE

G

- Gender
- GeorgiaStateDriversLicense
- GermanVehicleRegistration

H

- HairColors
- HawaiiStateDriversLicense
- HealthInsuranceClaimNumber
- Hobbies
- Honorifics
- Hostname

I

- ICD10Codes
- IdahoStateDriversLicense
- IllinoisStateDriversLicense
- IMEI
- INCOTerms
- IndianaStateDriversLicense
- IowaStateDriversLicense
- IP
- IPV6Address
- IrelandEircode
- ISIN
- ISOStateProvinceCode
- ItalianFiscalCode

J

- JapanCreditBureauCard

K

- KansasStateDriversLicense
- KentuckyStateDriversLicense

L

- LanguageList
- Latitude
- Longitude
- LouisianaStateDriversLicense

M

- MacAddress
- MaineStateDriversLicense
- ManitobaProvinceDriversLicense
- MaritalStatus

- MarylandStateDriversLicense
- MassachusettsStateDriversLicense
- MasterCard
- MichiganStateDriversLicense
- MinnesotaStateDriversLicense
- MississippiStateDriversLicense
- MissouriStateDriversLicense
- MontanaStateDriversLicense
- Month

N

- NameSuffix
- NebraskaStateDriversLicense
- NevadaStateDriversLicense
- NewBrunswickProvinceDriversLicense
- NewfoundlandandLabradorProvinceDriversLicense
- NewHampshireDriversLicense
- NewJerseyStateDriversLicense
- NewMexicoStateDriversLicense
- NewYorkStateDriversLicense
- NorthAmericaPhone
- NorthCarolinaStateDriversLicense
- NorthDakotaStateDriversLicense
- NovaScotiaProvinceDriversLicense

O

- OhioStateDriversLicense
- OklahomaStateDriversLicense
- OntarioProvinceDriversLicense
- OregonStateDriversLicense
- Organization

P

- PennsylvaniaStateDriversLicense
- Percent
- PoliticalParties
- PrinceEdwardIslandProvinceDriversLicense

Q

- QuebecProvinceDriversLicense

R

- Relationship
- Religions
- RhodeIslandStateDriversLicense

S

- SaskatchewanProvinceDriversLicense
- SouthCarolinaStateDriversLicense
- SouthDakotaStateDriversLicense
- SpanishNIF
- SSN4
- StateProvinceName

T

- TennesseeStateDriversLicense
- TexasStateDriversLicense

U

- UKPostCode

- UKProvinceCodes
- UniversalProductCode
- USCounty
- USSICCode
- USStateCodes
- USStateNames
- USAddressSimple
- UsaSocialSecurityNumber
- USFirstName
- USLastName
- USNationalDrugCode
- USPersonName
- USRoutingTransitNumber
- USSICCode
- USStateCapitals
- USStreetNames
- USZipCodes
- UtahStateDriversLicense

V

- VehicleIdentificationNumber
- VermontStateDriversLicense
- VirginiaStateDriversLicense
- VisaCard

W

- WashingtonStateDriversLicense
- WashingtonDCStateDriversLicense
- WestVirginiaStateDriversLicense
- WisconsinStateDriversLicense
- WyomingStateDriversLicense

Examples

The following example shows the command syntax for a REDACT operation where `mask-type` has a value of zero, and REDACT PRESERVE LENGTH has a `mask-type` value of 1. The data type is VARCHAR. For this example, the EMPLOYEE table has at least one record where FIRST_NAME is John:

```
SELECT DATA_MASK(FIRST_NAME, 0, 'X','','') FROM EMPLOYEE

1
----------
X

  1 record(s) selected

SELECT DATA_MASK(FIRST_NAME, 0, 'XX','','') FROM EMPLOYEE

1
----------
XX

  1 record(s) selected.


SELECT DATA_MASK(FIRST_NAME, 1, 'X','','') FROM EMPLOYEE

1
----------
XXXX

  1 record(s) selected.
```

The following example shows the command syntax for running a SUBSTITUTE operation where `mask-type` has a value of 2 and DECFLOAT is the input data type:

```
VALUES DATA_MASK(DECFLOAT(12.345),2,'','','0123456789abcdef')

1
```

```
------------------------------------------
                   5.9549086170364273E+307

  1 record(s) selected.
```

The following example shows the command syntax for running a `SUBSTITUTE` operation where `mask-type` has a value of 2 and `TIMESTAMP` is the input data type:

```
VALUES DATA_MASK(TIMESTAMP('2022-09-23 23:21:56'),2,'','','0123456789abcdef')

1
--------------------------
2074-08-26-21.06.57.153000

  1 record(s) selected.
```

The following example shows the command syntax for running a `PARTIAL REDACT` operation where `mask-type` has a value of 3. The example operation uses includes regular and slice expressions that redact the middle and first four characters from the end of the input:

```
VALUES DATA_MASK(VARCHAR('9038-3434-2345'),3,'RGX="^\d{4}[-]?(\d{4})"##char=A;SLC=0,4,1,Y','','')

1
--------------
9038-AAAA-YYYY

  1 record(s) selected.
```

The following example shows the command syntax for running a `FORMAT PRESERVE` operation where `mask-type` has a value of 4 and the input format is `USFirstName`:

```
VALUES DATA_MASK(CAST('John' as VARCHAR(10)), 4, '','USFirstName','0123456789abcdef')

1
----------
winfred

  1 record(s) selected.
```

The following example shows the command syntax for running a `FORMAT PRESERVING OBFUSCATION` operation `mask-type` has a value of 4 and the input format is `ArkansasStateDriversLicense`:

```
VALUES DATA_MASK(VARCHAR('900000000'),4,'','ArkansasStateDriversLicense','0123456789abcdef')

1
---------
982576755

  1 record(s) selected.
```

The following example shows the command syntax for running a `FORMAT PRESERVING OBFUSCATION` operation where `mask-type` has a value of 4 and the input format is `UsaSocialSecurityNumber`:

```
VALUES DATA_MASK(VARCHAR('123-45-6789'),4,'','UsaSocialSecurityNumber','0123456789abcdef')

1
-----------
680-47-4470

  1 record(s) selected.
```

The following example shows the command syntax for running a `FORMAT PRESERVING OBFUSCATION` operation where `mask-type` has a value of 4 and the input format is `TIMESTAMP`:

```
VALUES DATA_MASK(TIMESTAMP('2022-11-23 11:10:23.324'),4,'minDate=2001-01-01;maxDate=2099-12-31','','0123456789abcdef')

1
--------------------------
2068-10-15-03.59.29.174000
```

The following example shows the command syntax for running a `DATE AGING` operation where DATE is the input format and `TIMESTAMP` is the input data type:

```
VALUES DATA_MASK(DATE('2020-01-29'), 5, 'days=2', '', '')

1
----------
2020-01-31

  1 record(s) selected.


VALUES DATA_MASK(TIMESTAMP('2022-11-23 11:10:23.324'), 5, 'days=-2', '', '')

1
--------------------------
2022-11-21-11.10.23.324000

  1 record(s) selected.
```

The following example shows the command syntax for running an `IDENTIFIER` operation:

```
VALUES DATA_MASK(VARCHAR('AB1234'),7,'','','0123456789abcdef')

1
------
TW9980

  1 record(s) selected.

VALUES DATA_MASK(VARCHAR('AB#12@34'),7,'','','0123456789abcdef')

1
--------
TW#99@80

  1 record(s) selected
```

The following example shows the difference between a `FORMAT PRESERVING FABRICATION` operation (mask-type 10) where input validation does not occur and a `FORMAT PRESERVING OBFUSCATION` (mask-type 4) where the inputs are validated:

```
VALUES DATA_MASK(CAST('hfhfhf' as VARCHAR(10)),10,'','USFirstName','0123456789abcdef')

1
----------
kimiko

  1 record(s) selected.

VALUES DATA_MASK(CAST('hfhfhf' as VARCHAR(10)),4,'','USFirstName','0123456789abcdef')

1
----------
XXXXXXXXXX

  1 record(s) select
```