

En esta clase conoceremos los tipos de datos que maneja Redshift, y como ya lo sabes, Redshift está basado en PostgreSQL, de manera que si estás familiarizado con los tipos de datos de PostgreSQL esta clase se te facilitará.

Como otras bases de datos, Redshift maneja datos de tipo numérico, texto, de lógica booleana y estos datos tienen ciertas particularidades.

Tipos de datos para texto:

En estos tipos de datos almacenaremos caracteres y cadenas de texto como pueden ser nombres, direcciones, descripciones, etc.

Lo más importante que debes saber al momento de manejar estos tipos de datos es que el parámetro que le indiques a cada dato no es la longitud, es el número de bytes que trabajará, de manera que un dato VARCHAR(2) no indica que reciba 2 caracteres de longitud, sino que el contenido es de 2 bytes.

```
CREATE TABLE test_tipo_dato (  
dato varchar(2)  
)
```

```
insert into test_tipo_dato values ('AA');
```

```
insert into test_tipo_dato values ('€');
```

Si ejecutas las anteriores sentencias notarás que el primer insert se ejecuta sin ningún problema, “A” es un carácter que solo ocupa 1 byte. De modo que “AA” ocupa un total de 2 bytes. Pero, la segunda sentencia de insert falló y esta solo contiene un carácter “€”, esto ocurre por que este carácter tiene una longitud de 3 bytes por sí solo, de manera que rompe la restricción de tipo de dato VARCHAR(2).

Tipo de dato	Tipo de dato en redshift	Tamaño
CHAR, CHARACTER o NCHAR	Longitud de la cadena, incluidos espacios en blanco anteriores o posteriores (si corresponde).	4.096 bytes
VARCHAR, CHARACTER VARYING o NVARCHAR	4 bytes + bytes totales por caracteres, donde cada carácter puede tener entre 1 y 4 bytes.	65.535 bytes
BPCHAR	Convertido a CHAR(256) de longitud fija.	256 bytes
TEXT	Convertido a VARCHAR(256).	260 bytes

Como ves los tipos de dato BPCHAR son convertidos por Redshift a CHAR y los tipos de dato TEXT a VARCHAR; también es importante tener en cuenta que un dato CHAR de 10, siempre ocupara 10 bytes, y dado el caso que el dato guardado no complete los 10 bytes Redshift lo completa con espacios en blanco, esto no ocurre en el tipo de dato VARCHAR.

Compruébalo tú mismo:

```
create table test_char (  
dato char(10)  
);
```

```
insert into test_char values ('dato');
```

```
select * from test_char;
```

Tipos de datos numéricos:

Estos tipos de datos se deben usar para el manejo de números, en casos como cantidades, medidas, llaves únicas etc, los datos numéricos pueden ser enteros, decimales y de coma flotante.

Enteros:

Tipo de dato	Bytes	Rango
SMALLINT o INT2	2	De -32768 a +32767
INTEGER, INT o INT4	4	De -2147483648 a 2147483647
BIGINT o INT8	8	De -9223372036854775808 a 9223372036854775807

Decimales:

Tipo de dato	Bytes	Rango
DECIMAL o NUMERIC	128	38 Dígitos

Para el manejo de datos decimales, decimal(precisión, escala), debes tener en cuenta que se manejan dos parámetros, la precisión y la escala.

Para explicarte como funciona usemos un ejemplo:

```
create table test_decimal (  
dato decimal(4,2)  
);
```

```
insert into test_decimal values (4581.54);
```

```
insert into test_decimal values (121.7);
```

```
insert into test_decimal values (12.21);
```

```
insert into test_decimal values (12.2);
```

```
insert into test_decimal values (12.21222);
```

Como puedes observar los dos primeros “insert” fallaron, y quizá te estés preguntando, ¿por qué ocurrió esto? si en la sentencia dice 4 enteros y 2 decimales... pero no, el valor de precisión (4 para este ejemplo) indica el número total de dígitos que puede recibir, y escala indica el número de decimales que se trabajarán, de modo que en realidad este dato puede máximo guardar un número de 2 enteros y 2 decimales (4,2), pero ¿y la última sentencia insert que tenía 2 enteros y 5 decimales? ¿por qué funcionó?, bueno Redshift nota que el número de enteros es válido y solo redondea las cifras decimales a 2 dígitos para este caso.

Otros tipos de dato:

Los valores de fecha como date, timestamp, timestampz se manejan como cualquier otra base de datos, también los tipos de dato bool, boolean, los cuales solo pueden recibir un byte de longitud.

A continuación te comparto la tabla que usa Redshift para el manejo de tipos de datos.

Data type	Aliases	Description
SMALLINT	INT2	Signed two-byte integer
INTEGER	INT, INT4	Signed four-byte integer
BIGINT	INT8	Signed eight-byte integer
DECIMAL	NUMERIC	Exact numeric of selectable precision
REAL	FLOAT4	Single precision floating-point number
DOUBLE PRECISION	FLOAT8, FLOAT	Double precision floating-point number
BOOLEAN	BOOL	Logical Boolean (true/false)
CHAR	CHARACTER, NCHAR, BPCHAR	Fixed-length character string
VARCHAR	CHARACTER VARYING, NVARCHAR, TEXT	Variable-length character string with a user-defined limit
DATE		Calendar date (year, month, day)
TIMESTAMP	TIMESTAMP WITHOUT TIME ZONE	Date and time (without time zone)
TIMESTAMPTZ	TIMESTAMP WITH TIME ZONE	Date and time (with time zone)
GEOMETRY		Spatial data

