

Tarea 8. Tipos de datos en postgres.

Tipos numéricos

Name	Storage Size	Description	Range
smallint	2 bytes	small-range integer	-32768 to +32767
integer	4 bytes	typical choice for integer	-2147483648 to +2147483647
bigint	8 bytes	large-range integer	-9223372036854775808 to +9223372036854775807
decimal	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
numeric	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
real	4 bytes	variable-precision, inexact	6 decimal digits precision
double precision	8 bytes	variable-precision, inexact	15 decimal digits precision
smallserial	2 bytes	small autoincrementing integer	1 to 32767
serial	4 bytes	autoincrementing integer	1 to 2147483647
bigserial	8 bytes	large autoincrementing integer	1 to 9223372036854775807

Tipos de caracteres

Name	Description
character varying(<i>n</i>), varchar(<i>n</i>)	variable-length with limit
character(<i>n</i>), char(<i>n</i>)	fixed-length, blank padded
text	variable unlimited length

Name	Storage Size	Description
"char"	1 byte	single-byte internal type
name	64 bytes	internal type for object names

Tipos de fecha/tiempo

Name	Storage Size	Description	Low Value	High Value	Resolution
timestamp [(p)] [without time zone]	8 bytes	both date and time (no time zone)	4713 BC	294276 AD	1 microsecond
timestamp [(p)] with time zone	8 bytes	both date and time, with time zone	4713 BC	294276 AD	1 microsecond
date	4 bytes	date (no time of day)	4713 BC	5874897 AD	1 day
time [(p)] [without time zone]	8 bytes	time of day (no date)	00:00:00	24:00:00	1 microsecond
time [(p)] with time zone	12 bytes	time of day (no date), with time zone	00:00:00+1559	24:00:00-1559	1 microsecond
interval [fields] [(p)]	16 bytes	time interval	-178000000 years	178000000 years	1 microsecond

Ejemplos de entrada para fechas

Table 8.10. Date Input

Example	Description
1999-01-08	ISO 8601; January 8 in any mode (recommended format)
January 8, 1999	unambiguous in any datestyle input mode
1/8/1999	January 8 in MDY mode; August 1 in DMY mode
1/18/1999	January 18 in MDY mode; rejected in other modes
01/02/03	January 2, 2003 in MDY mode; February 1, 2003 in DMY mode; February 3, 2001 in YMD mode
1999-Jan-08	January 8 in any mode
Jan-08-1999	January 8 in any mode
08-Jan-1999	January 8 in any mode
99-Jan-08	January 8 in YMD mode, else error
08-Jan-99	January 8, except error in YMD mode
Jan-08-99	January 8, except error in YMD mode
19990108	ISO 8601; January 8, 1999 in any mode
990108	ISO 8601; January 8, 1999 in any mode
1999.008	year and day of year
J2451187	Julian date
January 8, 99 BC	year 99 BC

Tipos de datos monetarios

Table 8.3. Monetary Types

Name	Storage Size	Description	Range
money	8 bytes	currency amount	-92233720368547758.08 to +92233720368547758.07

Acepta la entrada con “\$” de números tanto enteros como decimales, además asume dos posiciones de punto decimal.

Tipos de datos JSON

Sirven para almacenar datos con JavaScriptObjectNotation establece que el texto almacenado aquí cumple con las características de notación de objetos de java script, por tanto existen algunas funciones y operadores específicos para este tipo de datos.

Table 8.23. JSON Primitive Types and Corresponding PostgreSQL Types

JSON primitive type	PostgreSQL type	Notes
string	text	\u0000 is disallowed, as are Unicode escapes representing characters not available in the database encoding
number	numeric	NaN and infinity values are disallowed
boolean	boolean	Only lowercase true and false spellings are accepted
null	(none)	SQL NULL is a different concept

Bibliografía

[1] “PostgreSQL 15.4 documentation”, PostgreSQL Documentation, 10-ago-2023. [En línea]. Disponible en: <https://www.postgresql.org/docs/15/index.html>. [Consultado: 21-sep-2023].