Databases (Key: 1644)
Postgres Datatypes

---------- Arellanes Conde Esteban ----------

Contents

1 Homework: Investigate Postgres Datatypes.

Investigate what, which and for what are the Postgres Datatypes along with the SQL Language.

- Forms of representation.
- Representation Capacity.
- Limitations 1 disadvantage.

2 Investigate what are the Postgres Datatypes.

According to the Postgres 17 (Current version) documentation these are all of the Postgres Datatypes available. Also PostgreSQL has a rich set of native data types available to users. Users can add new types to PostgreSQL using the CREATE TYPE command.

Each data type has an external representation determined by its input and output functions. Many of the built-in types have obvious external formats. However, several types are either unique to PostgreSQL, such as geometric paths, or have several possible formats, such as the date and time types. Some of the input and output functions are not invertible, i.e., the result of an output function might lose accuracy when compared to the original input.

Table 1. shows all the built-in general-purpose data types. Most of the alternative names listed in the *"Aliases"* column are the names used internally by PostgreSQL for historical reasons. In addition, some internally used or deprecated types are available, but are not listed here:

2.1 Data Types

Table of Contents of Numeric Types:

1. **Numeric Types**
    - Integer Types
    - Arbitrary Precision Numbers
    - Floating-Point Types
    - Serial Types
2. **Monetary Types**
3. **Character Types**
4. **Binary Data Types**
    - bytea Hex Format
    - bytea Escape Format

5. Date/Time Types

- Date/Time Input
- Date/Time Output
- Time Zones
- Interval Input
- Interval Output

6. Boolean Type

7. Enumerated Types

- Declaration of Enumerated Types
- Ordering
- Type Safety
- Implementation Details

8. Geometric Types

- Points
- Lines
- Line Segments
- Boxes
- Paths
- Polygons
- Circles

9. **Network Address Types**
    - inet
    - cidr
    - inet vs. cidr
    - macaddr
    - macaddr8
10. **Bit String Types**
11. **Text Search Types**
    - tsvector
    - tsquery
12. **UUID Type**
13. **XML Type**
    - Creating XML Values
    - Encoding Handling
    - Accessing XML Values

14. JSON Types

- JSON Input and Output Syntax
- Designing JSON Documents
- jsonb Containment and Existence
- jsonb Indexing
- jsonb Subscripting
- Transforms
- jsonpath Type

15. Arrays

- Declaration of Array Types
- Array Value Input
- Accessing Arrays
- Modifying Arrays
- Searching in Arrays
- Array Input and Output Syntax

16. Domain Types

17. Object Identifier Types

18. `pg_lsn` Type

19. Pseudo-Types

20. Composite Types

- Declaration of Composite Types
- Constructing Composite Values
- Accessing Composite Types
- Modifying Composite Types
- Using Composite Types in Queries
- Composite Type Input and Output Syntax

21. Range Types

- Built-in Range and Multirange Types
- Examples
- Inclusive and Exclusive Bounds
- Infinite (Unbounded) Ranges
- Range Input/Output
- Constructing Ranges and Multiranges
- Discrete Range Types
- Defining New Range Types
- Indexing
- Constraints on Ranges

3 Advantages vs. Disadvantages

One mayor advantage of PostgreSQL is that it has a rich and extensible set of Data Types and supports a wide variety of built-in data types, such as integers, floating-point numbers, strings, arrays, JSON, UUID, XML, geometric types (e.g., points, polygons), network address types (e.g., inet, macaddr), ranges, and more. Additionally, it allows users to define their own custom data types (e.g., composite types, enumerated types). This rich support for different types of data makes PostgreSQL highly flexible for handling diverse applications, from financial databases to GIS (Geographical Information Systems), and big data analysis. The ability to store complex data like JSON or arrays natively within the database is a powerful feature for modern applications.

But in the other hand, the complexity and learning curve PostgreSQL's has along with it's flexibility with data types is a major advantage, it can also be a disadvantage for developers who are new to the database system. The wide variety of data types, including specialized ones like `hstore`, `jsonb`, `pg_lsn`, and custom types, can be overwhelming. It may take more time and effort to understand when and how to use each type effectively. Additionally, understanding how to perform efficient indexing, querying, and optimization on complex data types can also add to the complexity of working with PostgreSQL also taking into account that if it's not correctly set up it could lead to security breaches like RCE or SQLi in previous versions that aren't fixed.

In summary:

Advantage	**Disadvantage**
Wide variety of Datatypes	Complex Learning Curve

Table 1: Cons. - Pros.

4 Representation Capacity

Name	Aliases	Description
bigint	int8	signed eight-byte integer
bigserial	serial8	autoincrementing eight-byte integer
bit [(n)]		fixed-length bit string
bit varying [(n)]	varbit [(n)]	variable-length bit string
boolean	bool	logical Boolean (true/false)
box		rectangular box on a plane
bytea		binary data (“byte array”)
character [(n)]	char [(n)]	fixed-length character string
character varying [(n)]	varchar [(n)]	variable-length character string
cidr		IPv4 or IPv6 network address
circle		circle on a plane
date		calendar date (year, month, day)
double precision	float8	double precision floating-point number (8 bytes)
inet		IPv4 or IPv6 host address
integer	int, int4	signed four-byte integer
interval [fields] [(p)]		time span
json		textual JSON data
jsonb		binary JSON data, decomposed
line		infinite line on a plane
lseg		line segment on a plane
macaddr		MAC (Media Access Control) address
macaddr8		MAC (Media Access Control) address (EUI-64 format)
money		currency amount
numeric [(p, s)]	decimal [(p, s)]	exact numeric of selectable precision
path		geometric path on a plane
time [(p)] [without time zone]		time of day (no time zone)
time [(p)] with time zone	timetz	time of day, including time zone

Table 2

Name	Aliases	Description
timestamp [(p)] [without time zone]		date and time (no time zone)
timestamp [(p)] with time zone	timestamptz	date and time, including time zone
tsquery		text search query
tsvector		text search document
txid_snapshot		user-level transaction ID snapshot (deprecated; see pg_snapshot)
uuid		universally unique identifier
xml		XML data
pg_lsn		PostgreSQL Log Sequence Number
pg_snapshot		user-level transaction ID snapshot
point		geometric point on a plane
polygon		closed geometric path on a plane
real	float4	single precision floating-point number (4 bytes)
smallint	int2	signed two-byte integer
smallserial	serial2	autoincrementing two-byte integer
serial	serial4	autoincrementing four-byte integer
text		variable-length character string

Table 3

5 References

[1] Ramez Elmasri, Shamkant B. Navathe. Fundamentals of Database Systems (Global Edition). Pearson 2017

[2] De Miguel Mart0̆0ednez, Adoraci0̆0f3n, Piattini, Mario, Esperanza, Marcos Dise0̆0f1o de bases de datos relacionales. M0̆0e9xico, Alfaomega, 2000.

[3] Kriegel, Alex; Trukhnov. SQL Bible, second edition, Willey 2008.

[4] Alan Beaulieu, Learning SQL: Generate, Manipulate, and Retrieve Data, OŔeilly, 3rd. Edition - 2020.

[5] Joan Casteel, Oracle 12c: SQL. Cengage Learning, Sep 11, 2015.

[6] Joan Casteel, Oracle 12c: SQL. Cengage Learning, Sep 11, 2015.

[7] PostgreSQL Global Development Group, "PostgreSQL Documentation," `https://www.postgresql.org/docs/`.

[8] IEEE, "Guide to Software Requirements Specifications," IEEE Std 830-1984. `https://chumpolm.files.wordpress.com/2018/09/ieee-std-830-1984.pdf`