UNAM FACULTAD DE INGENIERIA

Bases de Datos

Tarea 06

Martínez García José Eduardo

POSTGRES DATA TYPES

PostgreSQL has a rich set of native data types available to users. Users can add new types to PostgreSQL using the CREATE TYPE command [1]. PostgreSQL supports the following data types:

Postgres numeric data types

INTEGERS

1. **SMALLINT**: This is a 2-byte signed integer, with a range of -32768 to +32767 [2].

2. **INTEGER**: This is a 4-byte signed integer, with a range of -2147483648 to +2147483647 [2].

3. **BIGINT**: This is an 8-byte signed integer, with a range of -9223372036854775808 to +9223372036854775807 [2].

```
CREATE TABLE employees (
  id INTEGER PRIMARY KEY,
  name TEXT,
  age INTEGER,
  salary NUMERIC(10, 2)
);
--[2]
```

FLOATING POINT NUMBER

1. **float(n):** This is a floating-point number whose precision, is at least, n, up to a maximum of 8 bytes [3].

2. **real** or **float8:** This is a 4-byte floating-point number [3].

3. **numeric** or **numeric(p,s):** This is a real number with p digits with s number after the decimal point. This numeric(p,s) is the exact number [3].

```
CREATE TABLE orders (
  id SERIAL PRIMARY KEY,
  customer TEXT,
  amount NUMERIC(10, 2)
);
--[2]
```

Postgres character data types

1. **CHAR(n):** Is the fixed-length character with space padded. If you insert a string that is shorter than the length of the column, PostgreSQL pads spaces. If you insert a string that is longer than the length of the column, PostgreSQL will issue an error [3].

2. **VARCHAR(n):** Is the variable-length character string. The VARCHAR(n) allows you to store up to n characters. PostgreSQL does not pad spaces when the stored string is shorter than the length of the column [3].

3. **TEXT:** Is the variable-length character string. Theoretically, text data is a character string with unlimited length [3].

```
CREATE TABLE employees (
  employee_id serial PRIMARY KEY,
  last_name char(20),
  first_name varchar(20),
  hire_date date
);
--[2]
```

Postgres temporal data types

1. **DATE:** Stores the dates only [3].

2. **TIME:** Stores the time of day values [3].

3. **TIMESTAMP:** Stores both date and time values [3].

4. **TIMESTAMPTZ:** Is a timezone-aware timestamp data type. It is the abbreviation for timestamp with the time zone [3].

5. **INTERVAL:** Stores periods [3].

```
CREATE TABLE orders (
  order_id serial PRIMARY KEY,
  order_date date,
  customer_id int,
  checkin_time time,
  quantity int,
  created_at timestamp,
  duration interval,
  total decimal(8,2)
);
--[2]
```

Postgres Arrays

```
CREATE TABLE users (
  user_id SERIAL PRIMARY KEY,
  name VARCHAR(50) NOT NULL,
  emails TEXT[] NOT NULL
);
--[2]
```

Postgres special data types

1. **box:** A rectangular box [3].
2. **line:** A set of points [3].
3. **point:** A geometric pair of numbers [3].
4. **lseg:** A line segment [3].
5. **polygon:** A closed geometric [3].
6. **inet:** An IP4 address [3].
7. **macaddr:** A MAC address [3].

Postgres Boolean data types

1. **1, yes, y, t, true** values are converted to true [3].

2. **0, no, false, f** values are converted to false [3].

```
CREATE TABLE orders (
  order_id SERIAL PRIMARY KEY,
  customer_name VARCHAR(50) NOT NULL,
  order_date DATE NOT NULL,
  shipped BOOLEAN DEFAULT FALSE
);
--[2]
```

Postgres JSON data types

The JSON data type allows you to store JSON (JavaScript Object Notation) data in a column. This can be useful when you need to store complex data structures that can be easily serialized and deserialized [2].

```
CREATE TABLE products (
  product_id SERIAL PRIMARY KEY,
  name VARCHAR(50) NOT NULL,
  attributes JSON NOT NULL
);
--[2]
```

In this example, the “attributes” column is of type JSON and stores a JSON object containing various attributes for the product.

Postgres XML data types

The XML data type allows you to store XML (Extensible Markup Language) data in a column. This can be useful when you need to store and query XML data in your database [2].

```
CREATE TABLE books (
  book_id SERIAL PRIMARY KEY,
  title VARCHAR(50) NOT NULL,
  content XML NOT NULL
);
--[2]
```

References

- [1] “Chapter 8. Data types,” PostgreSQL Documentation, Feb. 20, 2025. https://www.postgresql.org/docs/current/datatype.html

- [2] J. Richman, “PostgreSQL data types explained with examples,” *Estuary*, Feb. 28, 2025. https://estuary.dev/blog/postgresql-data-types/

- [3] Neon, “PostgreSQL data types,” *Neon*, May 08, 2024. https://neon.tech/postgresql/postgresql-tutorial/postgresql-data-types