**What are the all kinds of Data Types in PostgreSQL?**

|  |  |  |
| --- | --- | --- |
| **Datatypes** | **Description** | **Example** |
| **SMALLINT** | Stores small-range integer values (-32,768 to 32,767) | 32767 |
| **INTEGER** | Standard integer type (-2^31 to 2^31-1) | 2147483647 |
| **BIGINT** | Large-range integer (-2^63 to 2^63-1) | 9223372036854775807 |
| **DECIMAL(p,s)** | Exact numeric type with fixed precision and scale | DECIMAL(10,2) = 12345.67 |
| **NUMERIC(p,s)** | Same as DECIMAL, used for financial calculations | NUMERIC(8,3) = 123.456 |
| **REAL** | Floating point number (4 bytes) | 3.14 |
| **DOUBLE PRECISION** | Floating point number (8 bytes) | 2.718281828 |
| **SERIAL** | Auto-incrementing integer (4 bytes) | 1, 2, 3, ... |
| **BIGSERIAL** | Auto-incrementing integer (8 bytes) | 1, 2, 3, ... |
| **CHAR(n)** | Fixed-length character string | ABC' |
| **VARCHAR(n)** | Variable-length character string | Hello, World!' |
| **TEXT** | Unlimited length character string | Lorem ipsum dolor sit amet' |
| **BOOLEAN** | Stores TRUE or FALSE | TRUE or FALSE |
| **DATE** | Stores a calendar date | 2024-02-07' |
| **TIME** | Stores time without time zone | 14:30:00' |
| **TIMESTAMP** | Stores date and time | 2024-02-07 14:30:00' |
| **TIMESTAMPTZ** | Stores date and time with time zone | 2024-02-07 14:30:00+05:30' |
| **INTERVAL** | Stores a time interval | 2 years 3 months' |
| **BYTEA** | Stores binary data | E'\\xDEADBEEF' |
| **UUID** | Stores a universally unique identifier | 550e8400-e29b-41d4-a716-446655440000' |
| **JSON** | Stores JSON data | {"name": "John", "age": 30}' |
| **JSONB** | Stores binary JSON data (faster queries) | {"name": "Alice", "city": "NY"}' |
| **ARRAY** | Stores an array of elements | {1,2,3,4,5}' |
| **HSTORE** | Stores key-value pairs | "name"=>"John", "age"=>"30"' |
| **CIDR** | Stores network address (IP) | 192.168.1.0/24' |
| **INET** | Stores an IP address | 192.168.1.1' |
| **MACADDR** | Stores a MAC address | 08:00:2b:01:02:03' |
| **TSVECTOR** | Full-text search vector | "'hello':1 'world':2" |
| **TSQUERY** | Full-text search query | hello & world' |
| **XML** | Stores XML data | <note><to>Tove</to></note>' |
| **POINT** | Stores a geometric point | (1.5, 2.5)' |
| **LINE** | Stores a geometric line | {(1,2), (3,4)}' |
| **LSEG** | Stores a line segment | [(1,2), (3,4)]' |
| **BOX** | Stores a rectangular box | ((1,2), (3,4))' |
| **PATH** | Stores a geometric path | ((1,2), (3,4), (5,6))' |
| **POLYGON** | Stores a polygon shape | ((1,2), (3,4), (5,6), (1,2))' |
| **CIRCLE** | Stores a circle | <(3,3),5>' |

**Creating Databases and Tables, CRUD Operations: CREATE, READ, UPDATE, DELETE**

Simple creation of table:

|  |
| --- |
| create table postgress\_learning (  username varchar (50),  age int,  email varchar (100),  gender CHAR (1),  salary DECIMAL (5,2),  id int,  Primary key(id)  ); |

Inserting into table:

|  |
| --- |
| INSERT INTO postgress\_learning (username, age, email, gender, salary, id) VALUES  ('Sanskar', 23, 'sanskardebnath2019@example.com', 'M', 500.00, 1),  ('Tripti', 20, 'tripti.m@example.com', 'F', 650.50, 2),  ('Rahul', 32, 'rahul.kapoor@example.com', 'M', 800.75, 3),  ('Priya', 24, 'priya.singh@example.com', 'F', 450.25, 4),  ('Anika', 29, 'anika.roy@example.com', 'F', 700.00, 5),  ('Avinash', 24, 'Avinash@example.com', 'M', 900.00, 6); |

Reading from the table

|  |
| --- |
| * select \* from postgress\_learning; * **select username, age from postgress\_learning;**        * **select username as Uname, age as difference from postgress\_learning;** |

Update query

|  |
| --- |
| --update postgress\_learning set username = 'SDN' where id = 1;  select \* from postgress\_learning where id =1;    **Extra**: Use returning \* after the where condition in update query to see what is updated;  **update postgress\_learning set username = 'SANSKAR DEBNATH' where id = 1**  **returning \*;**  **Updating multiple rows with update query:**  update postgress\_learning set age = age + 5 where gender = 'M' returning \*; |

Delete query:

|  |
| --- |
| Delete from postgress\_learning where id = 1;    **Caution : Don’t use Delete query without the where condition, else it will remove all the data (Rows)**  **As I execute it without where clause then this happened with me.** |

**Basic Queries: ORDER BY, LIMIT**

|  |
| --- |
| * select \* from postgress\_learning order by id DESC; * select \* from postgress\_learning order by salary DESC, id ASC; |

|  |
| --- |
| select \* from postgress\_learning order by salary DESC, id ASC LIMIT(2);  --it will display only two records out of all. |