Creating Tables: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2019

Syntax

- Returning the description of a table:

```
import psycopg2

conn = psycopg2.connect("dbname=dq user=dq")

cur = conn.cursor()

cur.execute('SELECT * FROM users LIMIT 0')

print(cur.description)
```

Concepts

- Using data types will save space on the database server which provides exponentially faster read and writes. In addition, having proper data types will ensure that any errors in the data will be caught and the data can be queried the way you expect.
- The description property outputs column information from the table. Within the column information, you will find the column data type, name, and other meta information.
- Numeric data types that Postgres supports:

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
		autoincrementing	

serial	4 bytes	autoincrementing integer	1 to 2147483647
bigserial	8 bytes	large autoincrementing integer	1 to 9223372036854775807

- `REAL` , `DOUBLE PRECISION` , `DECIMAL`, and `NUMERIC` can store float-like numbers such as: 1.23123, 8973.1235, and 100.00.
- The difference between `REAL` and `DOUBLE PRECISION` is that the `REAL` type is up to 4 byes, whereas the `DOUBLE PRECISION` type is up to 8 bytes.
- The `DECIMAL` type works as follows: The precision value which is the maximum amount of digits before and/or after the decimal point, whereas the scale is the maximum amount of digits after the decimal number where scale must be less than or equal to precision.

- The NUMERIC and DECIMAL types are equivalent in Postgres.
- Corrupted data is unexpected data that has been entered into the data set.
- String-like data types that Postgres supports:

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

- The difference between CHAR(N) and VARCHAR(N) is that CHAR(N) will pad any empty space of a character with whitespace characters while VARCHAR(N) does not.
- The BOOLEAN type can accept any of the following:
- The "true" state: True, 't' 'true', 'y', 'yes', 'no', '1'.
- The "false" state: False, 'f', 'false', 'n', 'no', 'off', '0'.
- Date/Time data types that Postgres supports:

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 / 14 digits
timestamp [(p)] with time zone	8 bytes	both date and time, with time zone	4713 BC	294276 AD	1 microsecond / 14 digits
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 / 14 digits
time [(p)] with time zone	12 bytes	times of day only, with time zone	00:00:00+1459	24:00:00-1459	1 microsecond / 14 digits
interval [fields] [(p)]	16 bytes	time interval	-178000000 years	178000000 years	1 microsecond / 14 digits

Resources

- The cursor class
- PostgreSQL data types