Exploring Postgres Internals: Takeaways

Syntax

- Getting all the tables within the public schema of a Postgres database:

```
conn = psycopg2.connect(dbname='dq', user='hud_admin', password='eRqg123EEkl')
cur = conn.cursor()
cur.execute("""
    SELECT table_name FROM information_schema.tables
    WHERE table_schema='public' ORDER BY table_name
""")
```

- Converting the name of a table to a Postgres string:

```
from psycopg2.extensions import AsIs
table_name = "state_info"
proper_interpolation = cur.mogrify("SELECT * FROM %s LIMIT 0", [AsIs(table_name)])
cur_execute(proper_interpolation)
```

- Converting a dictionary into a JSON string:

```
import json
d = {'int': 1, 'list': [1, 2, 3], 'dictionary': {'k': 1}}
json_string = json.dumps(d, indent=4)
```

- Converting a JSON string into a dictionary:

```
import json
json_string = '{"int": 1, "list": [1, 2, 3], "dictionary": {"k": 1}}'
dictionary = json.loads(json_string)
```

Concepts

- In every Postgres engine, there are a set of internal tables Postgres uses to manage its entire structure. These contain all the information about data, names of tables, and types stored in a Postgres database.
- We can use the `pg_tables` table to get a high-level overview of what tables are stored in the database.
- The `pg_catalog.pg_tables` structure is as follows:

Name	Type	Description
schemaname	name	Name of schema containing table
tablename	name	Name of table
tableowner	name	Name of table's owner

tablespace	name	Name of tablespace containing table (null if default for database)

hasindexes boolean True if table has (or recently had) any indexes hasrules boolean True if table has (or once had) rules hastriggers boolean TTrue if table has (or once had) triggers rowsecurity boolean True if row security is enabled on the table

- In Postgres, schemas are used as a namespace for tables with the distinct purpose of separating them into isolated groups or sets within a database.
- `AsIs` keeps the valid SQL representation of a non-string quoted instead of converting it.
- Using an internal table, we can accurately map the types for every column in a table.
- The JSON format is an open standard for writing dictionary-like data into a text file that is easy for a human to read.

Resources

- pg_catalog.pg_tables description
- The Information schema
- System catalogs
- pg_type table description
- pg_aggregate table description
- JSON format
- json Python library