User and Database Management: Takeaways

Syntax

- Connecting to the Postgres server with a secured user:

```
import psycopg2
conn = psycopg2.connect(dbname='dq', user='postgres', password='abc123')
```

- Creating a user with a password that can create other users and databases:

```
conn = psycopg2.connect(dbname='dq', user = 'postgres', password='password')
cur = conn.cursor()
cur.execute("CREATE USER data_viewer WITH CREATEUSER CREATEDB PASSWORD 'secret';")
conn.commit()
```

- Revoking privileges from a user on a table:

```
conn = psycopg2.connect(dbname='dq', user='dq')
cur = conn.cursor()
cur.execute("REVOKE ALL ON users FROM data_viewer;")
conn.commit()
```

- Granting the privileges to a user on a table:

```
conn = psycopg2.connect(dbname='dq', user='dq')
cur = conn.cursor()
cur.execute("GRANT SELECT ON users TO data_viewer;")
conn.commit()
```

- Targeting all tables inside a given schema:

```
REVOKE DELETE ON ALL TABLES IN schema_name FROM user_name;
GRANT SELECT ON ALL TABLES IN schema_name TO user_name;
```

- Creating a group:

```
CREATE GROUP readonly NOLOGIN;
```

- Assigning a user to a group:

```
GRANT readonly TO data_viewer;
```

- Creating a database with a specified owner:

```
CREATE DATABASE my_database OWNER postgres;
```

- Listing users:

```
SELECT * FROM pg_user;
```

- Listing table privileges:

```
SELECT grantor, grantee, privilege_type
FROM information_schema.table_privileges;
```

- Turning on autocommit mode instead of manually executing a transaction block:

```
conn = psycopg2.connect(dbname='dq', user='dq')
conn.autocommit = True
```

Concepts

- A superuser is like an administrator who has full access to the Postgres engine and can issue any command on every database, table, and user. You must enter a password when using a superuser to connect to the Postgres server.

- Privileges are rules that allow a user to run commands such as `SELECT`, `INSERT`, and `DELETE`. Server owners or database superusers can grant or revoke privileges.

- Not revoking certain privileges allows unaware users to issue incorrect commands and destroy the entire database.

- The most common practice when creating users is to create them, revoke all privileges, then choose the privileges you want to grant to them.

Resources

- [User privileges](#)
- [Groups](#)
- [Schemas](#)