# Intro to Postgres: Takeaways



## Syntax

- Connecting to a database using psycopg2:

```
import psycopg2
conn = psycopg2.connect("dbname=postgres user=postgres")
```

- Creating a table:

```
CREATE TABLE tableName(
   column1 dataType1 PRIMARY KEY,
   column2 dataType2,
   column3 dataType3,
   ...
);
```

- Dropping a table from a database:

```
DROP TABLE tableName
```

OR

```
DROP TABLE IF EXISTS tableName
```

- Inserting values using psycopg2:

```
import psycopg2
conn = psycopg2.connect("dbname=dq user=dq")
cur = conn.cursor()
insert_query = "INSERT INTO users VALUES {}".format("(10, 'hello@dataquest.io', 'Some
Name', '123 Fake St.')")
cur.execute(insert_query)
conn.commit()
```

OR

```
import psycopg2
conn = psycopg2.connect("dbname=dq user=dq")
cur = conn.cursor()
cur.execute("INSERT INTO users VALUES (%s, %s, %s, %s)", (10, 'hello@dataquest.io',
'Some Name', '123 Fake St.'))
conn.commit()
```

- Deleting data from a table:

```
DELETE from tableName
```

- Loading in a file using psycopg2:

```
conn = psycopg2.connect('dbname=postgres user=postgres')

cur = conn.cursor()

 # sample_file.csv has a header row.

with open('sample_file.csv', 'r') as f:

    # Skip the header row.

    next(f)

    cur.copy_from(f, 'sample_table', sep=',')
```

- Returning the first result:

```
cur.fetchone()
```

- Returning each row in a table:

```
cur.fetchall()
```

## Concepts

- Data engineers needs to have the skills to build a data pipeline that connects all the pieces of the data ecosystem together and keep it running.

- The parts of a data pipeline are the following:
    - Collecting
    - Short-Term Storage
    - Processing
    - Long-Term Storage
    - Presenting
- Relational databases are the most common storage used for web content, large business storage, and for data platforms.
- Postgres (or PostgreSQL) is one of the biggest open source relational databases.
- Postgres is one of the best options for data analysts.
- Postgres is a more robust engine that is implemented as a server. Postgres can also handle multiple connections and can implement more advanced querying features.
- `psycopg2` is an open source library that implements the Postgres protocol to connect to our Postgres server.
- SQL transactions prevent loss of data by ensuring all queries in a transaction block are executed at the same time. If any transactions fail then the whole group fails, and no changes are made to the database.
- A new transaction will automatically be created when we open a Connection in `psycopg2` .
- When a commit is called, the PostgreSQL engine will run all the queries at once. Not calling a commit or rollback will cause the transaction to stay in a pending state, and the changes will not be made.

## Resources

- Comparison of Relational Databases
- Psycopg2 documentation