**Open PostgreSQL**

psql -U postgres

Password for user postgres:

psql (17.3)

**Create Database**

postgres=# CREATE DATABASE my\_ToDoList\_API\_Database;

**View List of Databases**

postgres=# \l

List of databases

Name | Owner | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges

--------------------------+----------+----------+-----------------+---------+-------+--------+-----------+-----------------------

my\_todolist\_api\_database | postgres | UTF8 | libc | en-US | en-US | | |

postgres | postgres | UTF8 | libc | en-US | en-US | | |

template0 | postgres | UTF8 | libc | en-US | en-US | | | =c/postgres +

| | | | | | | | postgres=CTc/postgres

template1 | postgres | UTF8 | libc | en-US | en-US | | | =c/postgres +

| | | | | | | | postgres=CTc/postgres

(4 rows)

**Connect to Database**

postgres=# \c my\_todolist\_api\_database

You are now connected to database "my\_todolist\_api\_database" as user "postgres".

**Create Table within Database**

my\_todolist\_api\_database=# CREATE TABLE tasks (

my\_todolist\_api\_database(# id SERIAL PRIMARY KEY,

my\_todolist\_api\_database(# title VARCHAR(255) NOT NULL,

my\_todolist\_api\_database(# description TEXT,

my\_todolist\_api\_database(# completed BOOLEAN DEFAULT FALSE

my\_todolist\_api\_database(# );

CREATE TABLE

**Change Columns within Table**

my\_todolist\_api\_database=# ALTER TABLE tasks

my\_todolist\_api\_database-# DROP COLUMN title;

ALTER TABLE

my\_todolist\_api\_database=# ALTER TABLE tasks

my\_todolist\_api\_database-# DROP COLUMN description;

ALTER TABLE

my\_todolist\_api\_database=# ALTER TABLE tasks

my\_todolist\_api\_database-# ADD COLUMN name VARCHAR(255) NOT NULL;

ALTER TABLE

**View Table within Database**

my\_todolist\_api\_database=# \dt

List of relations

Schema | Name | Type | Owner

--------+-------+-------+----------

public | tasks | table | postgres

(1 row)

**View Tasks table format within Database**

my\_todolist\_api\_database=# \d tasks

Table "public.tasks"

Column | Type | Collation | Nullable | Default

-------------+------------------------+-----------+----------+-----------------------------------

id | integer | | not null | nextval('tasks\_id\_seq'::regclass)

title | character varying(255) | | not null |

description | text | | |

completed | boolean | | | false

Indexes:

"tasks\_pkey" PRIMARY KEY, btree (id)

**View Tasks in the Table using a Query**

SELECT \* FROM tasks;

**Delete all Tasks from Table using a Query**

DELETE FROM tasks;

**Create a Function to reset task ID if no tasks are present in table**

CREATE OR REPLACE FUNCTION reset\_task\_id()

RETURNS TRIGGER AS $$

BEGIN

-- Check if the tasks table is empty

IF NOT EXISTS (SELECT 1 FROM tasks) THEN

-- Reset the id sequence to 1

PERFORM setval('tasks\_id\_seq', 1, false);

END IF;

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

**Create a Trigger to execute ID reset function**

CREATE TRIGGER trigger\_reset\_task\_id

AFTER DELETE ON tasks

FOR EACH STATEMENT

EXECUTE FUNCTION reset\_task\_id();

**Quit PostgreSQL**

my\_todolist\_api\_database=# \q