**Student Name : Adeshina Ayodele**

**Task 1 –CRUD Operations**

**Exercise 1 - Create Table**

CREATE TABLE courses\_submit2 (

course\_id serial primary key,

course\_name varchar(60),

course\_author varchar(40),

course\_status varchar(10) check (course\_status in('published', 'draft', 'inactive')),

course\_published\_dt date

);

**Exercise 2 - Inserting Data**

select \* from courses\_submit2

INSERT INTO courses\_submit2 (course\_name, course\_author, course\_status, course\_published\_dt)

VALUES ('Programming using Python','Bob Dillon','published','2020-09-30');

INSERT INTO courses\_submit2 (course\_name, course\_author, course\_status, course\_published\_dt)

VALUES

('Data Engineering using Python', 'Bob Dillon', 'published', '2020-07-15'),

('Data Engineering using Scala', 'Elvis Presley', 'draft', NULL),

('Programming using Scala', 'Elvis Presley', 'published', '2020-05-12'),

('Programming using Java', 'Mike Jack', 'inactive', '2020-08-10'),

('Web Applications - Python Flask', 'Bob Dillon', 'inactive', '2020-07-20'),

('Web Applications - Java Spring', 'Mike Jack', 'draft', NULL),

('Pipeline Orchestration - Python', 'Bob Dillon', 'draft', NULL),

('Streaming Pipelines - Python', 'Bob Dillon', 'published', '2020-10-05'),

('Web Applications - Scala Play', 'Elvis Presley', 'inactive', '2020-09-30'),

('Web Applications - Python Django', 'Bob Dillon', 'published', '2020-06-23'),

('Server Automation - Ansible', 'Uncle Sam', 'published', '2020-07-05');

**Exercise 3 - Updating Data**

UPDATE courses\_submit2

SET course\_status = 'published',

course\_published\_dt = CURRENT\_DATE

WHERE course\_status = 'draft'

AND (course\_name ILIKE '%Python%' OR course\_name ILIKE '%Scala%');

**To check the table**

select \* from courses\_submit2

SELECT \* FROM courses\_submit2 WHERE course\_name ILIKE '%Python%' OR course\_name ILIKE '%Scala%';

**Exercise 4 - Deleting Data**

DELETE FROM courses\_submit2

WHERE course\_status NOT IN ('draft', 'published');

SELECT course\_author, COUNT(\*) AS published\_count

FROM courses\_submit2

WHERE course\_status = 'published'

GROUP BY course\_author

ORDER BY published\_count DESC;

**Task 2**

**Create a new database with the name: Netflix\_movies**

Create database Netflix\_movies;

**Create a table with the name: Movies**

CREATE TABLE Movies (

movie\_id SERIAL PRIMARY KEY,

title VARCHAR(250),

director VARCHAR(100),

year INTEGER,

length\_minutes INTEGER

);

**To check the table**

select \* from Movies

**Create a table with the name: BoxOffice**

CREATE TABLE BoxOffice (

movie\_id INTEGER PRIMARY KEY,

rating DECIMAL(3,1),

domestic\_sales BIGINT,

international\_sales BIGINT

);

**To check the table**

select \* from BoxOffice

**To insert into the table Movies with data given.**

INSERT INTO Movies (movie\_id, title, director, year, length\_minutes)

VALUES

(1, 'Toy Story', 'John Lasseter', 1995, 81),

(2, 'A Bugs Life', 'John Lasseter', 1998, 95),

(3, 'Toy Story 2', 'John Lasseter', 1999, 93),

(4, 'Monsters, Inc.', 'Pete Docter', 2001, 92),

(5, 'Finding Nemo', 'Andrew Stanton', 2003, 107),

(6, 'The Incredibles', 'Brad Bird', 2004, 116),

(7, 'Cars', 'John Lasseter', 2006, 117),

(8, 'Ratatouille', 'Brad Bird', 2007, 115),

(9, 'WALL-E', 'Andrew Stanton', 2008, 104),

(10, 'Up', 'Pete Docter', 2009, 101),

(11, 'Toy Story 3', 'Lee Unkrich', 2010, 103),

(12, 'Cars 2', 'John Lasseter', 2011, 120),

(13, 'Brave', 'Brenda Chapman', 2012, 102),

(14, 'Monsters University', 'Dan Scanlon', 2013, 110);

**To insert into the table BoxOffice with data given.**

INSERT INTO BoxOffice (movie\_id, rating, domestic\_sales, international\_sales)

VALUES

(5, 8.2, 380843261, 555900000),

(14, 7.4, 268492764, 475066843),

(8, 8.0, 206445654, 417277164),

(12, 6.4, 191452396, 368400000),

(3, 7.9, 245852179, 239163000),

(6, 8.0, 261441092, 370001000),

(9, 8.5, 223808164, 297503696),

(11, 8.4, 415004880, 648167031),

(1, 8.3, 191796233, 170162503),

(7, 7.2, 244082982, 217900167),

(10, 8.3, 293004164, 438338580),

(4, 8.1, 289916256, 272900000),

(2, 7.2, 162798565, 200600000),

(13, 7.2, 237283207, 301700000);

**To check the table**

select \* from BoxOffice