-- DBS211 Lab 05

-- Victor Krenzel

-- 102446176

-- 10/16/2022

SET AUTOCOMMIT ON;

-- Part A (DDL) :

-- 1. Create table the following tables and their given constraints:

-- L5\_MOVIES (movieid:int, title:varchar(35), year:int, director:int,score:decimal(3,2))

-- L5\_ACTORS (actorid:int, name:varchar(20), lastname:varchar(30))

-- L5\_CASTINGS (movieid:int, actorid:int)

-- L5\_DIRECTORS(id:int, name:varchar(20), lastname:varchar(30))

CREATE TABLE L5\_movies ( mID INT PRIMARY KEY, title VARCHAR(35) NOT NULL, releaseYear INT NOT NULL, director INT NOT NULL, score DECIMAL(3, 2) );

CREATE TABLE L5\_actors ( aID INT PRIMARY KEY, firstname VARCHAR(20) NOT NULL, lastname VARCHAR(30) NOT NULL );

CREATE TABLE L5\_castings ( movieID INT PRIMARY KEY, actorID INT );

CREATE TABLE L5\_directors ( directorID INT PRIMARY KEY, firstname VARCHAR(20) NOT NULL, lastname VARCHAR(30) NOT NULL );

-- 2. Modify the movies table to create a foreign key constraint that refers to table directors.

ALTER TABLE L5\_movies

ADD CONSTRAINT movies\_directors\_fk FOREIGN KEY (director) REFERENCES L5\_directors(directorID);

-- 3. Modify the movies table to create a new constraint so the uniqueness of the movie title is guaranteed.

ALTER TABLE L5\_movies

ADD CONSTRAINT title\_unique UNIQUE (title);

-- 4. Write insert statements to add the following data to table directors and movies.

INSERT ALL

INTO L5\_directors VALUES (1010, 'Rob', 'Minkoff')

INTO L5\_directors VALUES (1020, 'Bill', 'Condon')

INTO L5\_directors VALUES (1050, 'Josh', 'Cooley')

INTO L5\_directors VALUES (2010, 'Brad', 'Bird')

INTO L5\_directors VALUES (3020, 'Lake', 'Bell')

SELECT \* FROM dual;

INSERT ALL

INTO L5\_movies VALUES (100, 'The Lion King', 2019, 3020, 3.50)

INTO L5\_movies VALUES (200, 'Beauty and the Beast', 2017, 1050, 4.20)

INTO L5\_movies VALUES (300, 'Toy Story 4', 2019, 1020, 4.50)

INTO L5\_movies VALUES (400, 'Mission Impossible', 2018, 2010, 5.00)

INTO L5\_movies VALUES (500, 'The Secret Life of Pets', 2016, 1010, 3.90)

SELECT \* FROM dual;

-- 5. Write SQL statements to remove all above tables. Is the order of tables important when removing? Why?

DROP TABLE L5\_actors;

DROP TABLE L5\_castings;

DROP TABLE L5\_directors;

DROP TABLE L5\_movies;

-- Yes, It's important to remove the child table first that's linked to a parent table via a foreign key, then remove the parent table last.

-- In this case, the L5\_directors table is a child of the L5\_movies table.

-- Part B (More DML):

-- 6. Create a new empty table employee2 the same as table employees. Use a single statement to create the table and insert the data at the same time.

CREATE TABLE employee2

AS (SELECT \* FROM employees);

-- 7. Modify table employee2 and add a new column username to this table. The value of this column is not required and does not have to be unique.

ALTER TABLE employee2

ADD username VARCHAR(20);

-- 8. Delete all the data in the employee2 table.

DELETE FROM employee2;

-- 9. Re-insert all data from the employees table into your new table employee2 using a single statement.

INSERT INTO

employee2 (employeenumber, lastname, firstname, extension, email, officecode, reportsto, jobtitle)

SELECT

\*

FROM

employees;

-- 10. In table employee2, write a SQL statement to change the first name and the last name of employee with ID 1002 to your name.

UPDATE

employee2

SET

firstname = 'Victor', lastname = 'Krenzel'

WHERE

employeenumber = 1002;

-- 11. In table employee2, generate the email address for column username for each student by concatenating the first character of employee’s first name and the employee’s last name.

-- For instance, the username of employee Peter Stone will be pstone.

-- NOTE: the username is in all lower case letters.

UPDATE employee2

SET username = LOWER(CONCAT(SUBSTR(firstname, 1, 1), lastname));

-- 12. In table employee2, remove all employees with office code 4.

DELETE FROM employee2

WHERE officecode = 4;

-- 13. Drop table employee2

DROP TABLE employee2;