DDL Script Assignment

Goal: In this assignment, you are tasked with creating a fully functioning DDL script that we can run to create a demo of the ERD that you designed in the previous Database Design Assignment.  Your script must not only build the database according to the requirements below but also seed it with some data for testing purposes.  Lastly, your script must be able to run over and over, meaning it can drop database objects (e.g. tables, sequences, indexes), recreate those objects, and seed with test data in a single script.

# Assignment Requirements:

## **Coding Standards**

* **Naming**: Use the final ERD as a reference on how to name tables and columns. For constraints, make the names clear and easy to understand. Avoid abbreviations.
* **Data Types / Lengths:**
  + *NUMBER* – Follow the standard of making IDs numeric and any column that involves arithmetic.
  + *VARCHAR –* All non-numeric fields should be VARCHAR except fields mentioned in CHAR section below.
  + *CHAR* – Phone numbers will all be defined as CHAR length of 12 to allow for 12 digits to include international code. State is the abbreviation of a state so it’s CHAR length of 2. Zipcode only needs to store a length of 5. CC\_Flag will be used to track a simple ‘N’ for “No” and ‘Y’ for “Yes” so this always has a length of 1.
  + *DATE* – Any dates and time fields should clearly be DATE format
  + Note: Lengths can vary unless specified. Make sure that you choose lengths that are appropriate for the content of that column. Comments and subtitles likely need more space than a first name.
* **Commenting**
  + Include comments at least with each section of code (i.e. DROP, CREATE, INSERT, INDEXES)
  + Comments should include a description of section and your name/uteid as the author.
  + If you want to add additional comments to single statements or lines feel free but it’s not required. It’s just a best practice to comment code well.

## **Constraints**

* Assign primary and foreign keys per the design.
* Only the following can be NULL: browser\_user *middle\_name*
* The following fields should be UNIQUE: Browser\_User.Email, ContentCreator.username
* DEFAULTS:
  + CC\_flag defaults to ‘N’
  + Views, Likes, and revenue should all default to 0
* Make sure the following Check constraints are added:
  + User birthdate – birthdate must be at least 13 years prior to the today’s date. We may not have done exactly this in class, but you can figure it out any way you can by exhausting your resources including the textbook, Google, and Piazza. TIP: Be careful about whether functions do what you think and always feel free to use the DUAL table to check functions (pg. 94). It would be best to use SYSDATE, but we cannot use nondeterministic functions in a check constraint. Options ☹
    - SELECT [some function]  
      FROM dual;
  + User Email - emails should have a character length of at least 7 or more. Again, just because we didn’t cover it in class doesn’t mean you can’t google it and figure it out. Name this constraint “email\_length\_check”.

## **Other**

* Create indexes on all foreign keys that are not also a primary key. Since primary keys are indexed we won’t index a column that is both a PK and FK. Also index at least 2 other fields in the schema to show you can properly discern which columns should have indexes per design rules discussed in class
* Create sequences that auto-increment the *user\_id* starting at 1000000, *card\_id* starting at 1000000, *topic\_id, video\_id*, and *comment\_id* at reasonable values. Increment by 1. It can be best to have all numbers take up the same space, which can make starting at 1000 better than starting at 1 if the maximum is 9999, or if we want clearer consistency in id numbers.

## **Format**

* **Easy to Read Code:** All your code should be well spaced and indented. If it’s not easy to read and messy, you could lose points.
* **Sections**: You must create your script with the three following sections:
  + Drop Sequence/Tables section - Area of the script that drops all tables and sequences in proper order
  + Create Sequence/Tables section - Area of the script that creates tables/sequences and adds constraints either via CREATE or ALTER TABLE statements
  + Insert Data section - Area of the script that inserts data into the tables using “INSERT INTO”
  + Create Index section – After you seed data, add in indexes for the database to optimize performance
* **Commenting**: You should add comments before each section that includes a *description* of what is happening in that section and *your name and UTEID*, which is a best practice to know who coded what.
* The script must build the database shown in the ERD exactly which means table names, column names, and constraints must match.  That being said, we are not going to specify the exact names of constraints that you create unless stated above.  Just be sure to use logical, clear names for constraints.

## **Data Requirements**

Seed your tables with based on the following requirement: NOTE: **Include commits** after each group of inserts for a particular table and don’t forget to regular commit to avoid taxing the server and causing [NOWAIT error](https://stackoverflow.com/questions/4842765/ora-00054-resource-busy-and-acquire-with-nowait-specified-or-timeout-expired).

Make up fun names!

* Create 6 **users**, where four of them have corresponding content creation accounts and two do not. Give at least two middle names and two without middle names.
* Create the **content creation accounts** for your four users. Two of four should have two accounts each.
* Create two **credit cards** for one content creator and one credit card each for two other content creators.
* Create four **videos** with upload\_date having both date and time included. Have one video be recent and default to the default value for views, likes, and revenue.
* Create four **comments** in the comments linking table.

Optional – does not count, but is for my completionists out there.

* Create four different **topics** of videos.
* Create the **topic and video linking table** and populate with four rows.
* Create the **topic and user linking table** and populate with four rows.

## **Testing before you turn in your work**

* Make sure your script runs without errors before you submit it. Test it by dropping all the tables and the running the script. Then run the script to ensure no unexpected errors occur.

**What to turn in**

* Turn in your DDL script in a (.sql) file format. If you have issues doing this, at least save as a .txt file. **Any other file type will result in a 0.**