**Assignment 2 – Loading Data into Database**

In this assignment, we are going to write a SQL script to create and populate a database. The table definition (see below) contains necessary details about the tables and columns.

This assignment is two-part: 1) creating tables on your Oracle database and 2) inserting records into those tables.

1. The first part is to use DDL (CREATE) and create all tables, columns, and constraints as specified in the table definition. **The name, data type, and length of the columns MUST follow the definitions provided.**
2. In the second part, you need to populate the created tables using DML (INSERT). You can add as many records as you want. The more the better. But there is a requirement for the minimum number of records: your database must have at least 3 albums, 20 songs, 5 customers, and 10 orders.
   * Populate first the main tables – CUSTOMER, ORDER\_RECORD, SONG, ARTIST, GENRE, and ALBUM.
   * Then, move onto the matching (or associative) tables – WRITES, BELONGS\_TO, CONTAINS, PLAY\_HISTORY, and ORDER\_LINE -- as the records of these tables must be present in the tables they are connecting. For example, for WRITES to have song A and artist B, song A must be in SONG and artist B must be in ARTIST. Otherwise, this record (song A, artist B) in WRITES would be referring to something that does not exist in the database (which violates the foreign key constraint).

The final output would look similar to the JustLeeDB file: a script that contains all SQL commands **(see below for example codes).**

To get the full score, your script must generate the tables and records as the requirements and run without any errors. If your code fails to run on my computer, I will look into it to give you partial credits, but you won’t get full marks.

**Submission guidelines**

* Deliverable: A script file (.sql) that contains all the SQL commands needed to generate the tables, columns, constraints, and records as the requirements
  + To save a script from SQL Developer Web, click 'Download Editor Content' button on the toolbar on the top
* Before submitting, run your script on your machine and make sure it runs without errors
* File extension: .sql
* Filename: asgmt2\_lastname\_firstname.sql (e.g., asgmt2\_ lee\_kyunghee.sql)
* Upload the text file to Canvas
* As specified in the syllabus, there is a penalty for late submission, 10% per day, and any submissions over five days late will not be accepted

 If you have any questions, leave comments in [the assignment 2 discussion board!](https://canvas.wayne.edu/courses/136902/discussion_topics/681722)

**Table Definition**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table** | **Column** | **Data Description** | **Length** | **Description/Possible Values** |
| CUSTOMER | **CustomerID** | Numeric | 6 | Primary key |
|  | CustomerName | Character | 20 |  |
|  | City | Character | 20 |  |
|  | State | Fixed character | 2 |  |
|  | PostalCode | Fixed character | 5 |  |
|  | PhoneNumber | Fixed character | 10 |  |
|  | Birthday | Date |  |  |
|  | RegistrationDate | Date |  |  |
| ORDER\_RECORD | **OrderID** | Numeric | 10 | Primary key |
|  | OrderDate | Date |  |  |
|  | PaymentMethod | Fixed character | 2 | CS(Cash), CC(Credit Card), or PP(Paypal) |
|  | OrderType | Fixed character | 1 | P(Physical order), D(Digital order) |
|  | CustomerID | Numeric | 6 | Foreign key |
| DIGITAL\_ORDER | **DOrderID** | Numeric | 10 | Primary key; Foreign key |
|  | SubscriptionOption | Fixed character | 1 | M(Monthly) or Y(Yearly) |
| PHYSICAL\_ORDER | **POrderID** | Numeric | 10 | Primary key; Foreign key |
|  | DeliveryOption | Fixed character | 1 | F(Free), S(Standard), or E(Expedite) |
| PLAY\_HISTORY | **DOrderID** | Numeric | 10 | Primary key; Foreign key |
|  | **SongID** | Numeric | 10 | Primary key; Foreign key |
|  | PlayDate | Date |  |  |
|  | PlayCount | Numeric | 3 | Number of times of a song played per day (e.g., 10) |
| ORDER\_LINE | **POrderID** | Numeric | 10 | Primary key; Foreign key |
|  | **AlbumID** | Numeric | 10 | Primary key; Foreign key |
|  | QuantitiesOrdered | Numeric | 3 |  |
| SONG | **SongID** | Numeric | 10 | Primary key |
|  | SongTitle | Character | 50 |  |
|  | PlayTime | Numeric | 3 | Playtime of a song in minute (e.g., 3) |
| ALBUM | **AlbumID** | Numeric | 10 | Primary key |
|  | AlbumTitle | Character | 50 |  |
|  | AlbumPrice | Numeric | 3 |  |
|  | ReleaseDate | Date |  |  |
| ARTIST | **ArtistID** | Numeric | 10 | Primary key |
|  | ArtistName | Character | 20 |  |
|  | DebutDate | Date |  |  |
| GENRE | **GenreID** | Numeric | 4 | Primary key |
|  | GenreName | Character | 50 |  |
| CONTAINS | **SongID** | Numeric | 10 | Primary key; Foreign key |
|  | **AlbumID** | Numeric | 10 | Primary key; Foreign key |
| WRITES | **ArtistID** | Numeric | 10 | Primary key; Foreign key |
|  | **SongID** | Numeric | 10 | Primary key; Foreign key |
| BELONGSTO | **SongID** | Numeric | 10 | Primary key; Foreign key |
|  | **GenreID** | Numeric | 4 | Primary key; Foreign key |

**Example Code**

To create the CUSTOMER table:

CREATE TABLE CUSTOMER  
  
(CustomerID  NUMBER(6),  
  
CustomerName VARCHAR2(20),  
  
City VARCHAR2(20),  
  
State VARCHAR2(2),  
  
PostalCode VARCHAR2(5),  
  
PhoneNumber VARCHAR2(10),  
  
Birthday DATE,  
  
RegistrationDate DATE,  
  
     CONSTRAINT customer\_customerid\_pk PRIMARY KEY (CustomerID)  
  
);

To insert new records into the CUSTOMER table:

INSERT INTO CUSTOMER  
  
VALUES (991001, 'MORALES BONITA', 'EASTPOINT', 'FL', '32328', '3132223333', '02-APR-70', '01-SEP-10');  
  
INSERT INTO CUSTOMER  
  
VALUES (991002, 'THOMPSON RYAN', 'SANTA MONICA', 'CA', '90404', '5482222678', '28-FEB-80', '01-SEP-15');

To see the records:

SELECT \* FROM CUSTOMER;

Rubric

**Assignment 2 - Loading Data**

| Assignment 2 - Loading Data | | |
| --- | --- | --- |
| **Criteria** | **Ratings** | **Pts** |
| This criterion is linked to a Learning OutcomeCreating tables  All tables were correctly created. | |  |  |  |  | | --- | --- | --- | --- | | **7.0 pts**  **Full Marks** | **5.0 pts**  **No Description** | **3.0 pts**  **No Description** | **1.0 pts**  **Needs improvement** | | 7.0 pts |
| This criterion is linked to a Learning OutcomeColumns  Each table includes all relevant columns as stated in the table definition. | |  |  |  |  | | --- | --- | --- | --- | | **7.0 pts**  **Full Marks** | **5.0 pts**  **No Description** | **3.0 pts**  **No Description** | **1.0 pts**  **Needs improvement** | | 7.0 pts |
| This criterion is linked to a Learning OutcomePopulating tables  All tables were populated with enough data points (at least 3 albums, 20 songs, 5 customers, and 10 orders), including the associative (or matching) tables. | |  |  |  |  | | --- | --- | --- | --- | | **7.0 pts**  **Full Marks** | **5.0 pts**  **No Description** | **3.0 pts**  **No Description** | **1.0 pts**  **Needs improvement** | | 7.0 pts |
| This criterion is linked to a Learning OutcomeConstraints  The constraints required in the table definition were correctly specified. | |  |  |  |  | | --- | --- | --- | --- | | **4.0 pts**  **Full Marks** | **3.0 pts**  **No Description** | **2.0 pts**  **No Description** | **1.0 pts**  **Needs improvement** | | 4.0 pts |
| Total Points: 25.0 | | |