CNIT 48700 DATABASE ADMINISTRATION LAB

- **Login credentials:**
  - Your database user name for this lab is the same as your Career account.
  - Your initial password for the database should appear in Brightspace.
  - If you run into an issue regarding your database account, please complete a Support Request with ECN: https://engineering.purdue.edu/ECN/AboutUs/ContactUs/?organization=technology.
  - To change your password in SQL Developer, use the following command, substituting the appropriate values for username and newpassword:

    ```
        alter user username identified by newpassword;
    ```

    Depending on your configuration, simply typing the password command at a prompt may allow you to change your password.
  - After the initial lab, you will create and administrate your own database with a distinct set of credentials.

- **Troubleshooting:**
  - One may view compilation errors by using the show errors command: SQL> `show errors`
  - Remember to set your SERVEROUTPUT settings where necessary. To extend the buffer size associated with the SERVEROUTPUT setting, specify "`set serveroutput on size n`" where n is the desired size.

- **Deliverables:**
  - For each lab unless otherwise noted, compose a text file containing the lab's code/commands and the output generated by executing the code/commands.
  - All artifacts (such as code and output) associated with a lab should be submitted by the lab's due date via Brightspace.
  - To receive full participation credit for a lab, you must also review your work with the lab instructor.

LAB 1: SQL REVIEW

DUE DATE: SEPTEMBER 10, 2024 (BEGINNING OF LAB)

GOAL: Review Structured Query Language (SQL)

ACTIVITIES

1. **CONNECTING TO THE DATABASE**
    1. Read Getting Started with Oracle SQL Developer and familiarize yourself with Oracle SQL Developer**.**
    2. Start Oracle SQL Developer and create a connection in SQL Developer using the following parameters:
        1. PROTOCOL = TCP
        2. HOST = oracle.tech.purdue.edu
        3. PORT = 1521
        4. SID = tech
        5. SERVICE_NAME = tech.tech.purdue.edu
    3. Save the connection using your user name and initial password.
    4. Once logged in, change your password.

2. **DESCRIPTION OF THE DATABASE**

    MusicalInfo is a database used by a music company to keep track of its albums. The database records information about musicians, albums and songs including details about song recordings. MusicalInfo consists of the following relations (primary key is underlined for each relation):

    MUSICIAN (MNO, NAME, ADDRESS, PHONE)
    ALBUM (ANO, ATITLE, COPYRIGHT_DATE, MNO, COPY_SOLD)
    SONG (SNO, STITLE, ANO, LYRICIST)
    PLAY (MNO, INSTRUMENT)
    PERFORM (MNO, SNO, INSTRUMENT)

    The MUSICIAN relation contains information about the musicians in the system.

    The ALBUM relation contains information about the albums. MNO refers to a musician already existing in the MUSICIAN relation and acts as the producer of the album. Other information is: title, copyright date, and the number of copies that have been sold. An album contains many songs. An album may have many songs performed by various musicians, but it has only one musician as its producer.

    The SONG relation contains information about songs in an album (referred to by ANO). LYRICIST is the name of the lyricist who wrote the lyrics of the song.

    The PLAY relation has information about the instruments that a musician can play.

The PERFORM relation contains information about the instrument that a musician plays in each song. Note that a musician can play many instruments, but in a particular song, they play only one instrument.

3. **LOAD AND EXPLORE DATA**
   1. Download lab1.sql from lab files, and save it to a location of your choice.
   2. Execute the lab1.sql script file in SQL Developer to create and populate tables.
   3. Determine the tables in your account by running the following command:
      ```
      SELECT table_name FROM user_tables;
      ```
   4. Use `Describe` command to determine the columns and data types for the columns in each of the tables.

4. **SQL QUERIES**
   Write SQL queries for the following questions and run them on the Oracle database. If you are making any assumptions, state them clearly and document your queries.

   1. Find the names of musicians who cannot play ‘Piano’.
   2. Find the names of musicians who only use ‘Piano’ or ‘Guitar’ to perform.
   3. Find the phone numbers of all musicians who had used ‘Guitar’ to perform.
   4. Find the ANOs of the albums which have a song performed by ‘Guitar’.
   5. Find the ANOs of the albums which only have songs performed by ‘Guitar’.
   6. Find the MNO of the musician who produces the largest number of albums.
   7. Find the names of the musicians who can play all instruments.
   8. Find the names of the musicians who can play only one instrument.
   9. Find the names of the musicians who have never performed a song produced by other musicians.
   10. Find the ANOs of the albums that have a song performed by other musicians (not the producer).
   11. Find the names of the musicians who have performed for albums that have a copyright before ‘03-Sep-1999’.
   12. Find the name of the musician who has the album with the maximum number of sold copies.
   13. Find the names of (distinct) musicians who can play 2 or more instruments.
   14. For each musician except ‘Sid Vicious’, print the musician's name, phone and average number of sold copies of albums produced by the musician. If the musician has not produced any album, print 0 for the average number of sold copies.

5. **DELETE ALL DATA**
   Run the lab1drop.sql script file downloaded from Brightspace to delete all tables.

   Check all tables in your account have been dropped:
   ```
   SELECT table_name FROM user_tables;
   ```

6. Review the lab with the instructor. Create and submit a text file containing the SQL commands via Brightspace.