SQL Lab

1. Setting up Oracle Chinook

In this section you will begin the process of working with the Oracle Chinook database··
Task – Open the Chinook_Oracle.sql file and execute the scripts within.
https://github.com/lerocha/chinook-database/tree/master/ChinookDatabase/DataSources

2 SQL Queries

In this section you will be performing various queries against the Oracle Chinook database.

2.1 SELECT

Task – Select all records from the Employee table.

Task – Select all records from the Employee table where last name is King.

Task – Select all records from the Employee table where first name is Andrew and REPORTSTO is NULL.

2.2 ORDER BY

Task – Select all albums in Album table and sort result set in descending order by title.

Task – Select first name from Customer and sort result set in ascending order by city

2.3 INSERT INTO

Task – Insert two new records into Genre table

Task – Insert two new records into Employee table

Task – Insert two new records into Customer table

2.4 UPDATE

Task – Update Aaron Mitchell in Customer table to Robert Walter

Task – Update name of artist in the Artist table “Creedence Clearwater Revival” to “CCR”

2.5 LIKE

Task – Select all invoices with a billing address like “T%”

2.6 BETWEEN

Task – Select all invoices that have a total between 15 and 50

Task – Select all employees hired between 1st of June 2003 and 1st of March 2004

2.7 DELETE

Task – Delete a record in Customer table where the name is Robert Walter (There may be constraints that rely on this, find out how to resolve them).

3. SQL Functions

In this section you will be using the Oracle system functions, as well as your own functions, to perform various actions against the database

3.1 System Defined Functions

Task – Create a function that returns the current time.

Task – create a function that returns the length of name in MEDIATYPE table

3.2 System Defined Aggregate Functions

Task – Create a function that returns the average total of all invoices

Task – Create a function that returns the most expensive track

3.3 User Defined Scalar Functions

Task – Create a function that returns the average price of invoiceline items in the invoiceline table

3.4 User Defined Table Valued Functions

Task – Create a function that returns all employees who are born after 1968.

¶¶¶¶······

4.0 Stored Procedures

In this section you will be creating and executing stored procedures. You will be creating various types of stored procedures that take input and output parameters.

4.1 Basic Stored Procedure

Task – Create a stored procedure that selects the first and last names of all the employees.

4.2 Stored Procedure Input Parameters

Task – Create a stored procedure that updates the personal information of an employee.

Task – Create a stored procedure that returns the managers of an employee.

4.3 Stored Procedure Output Parameters

Task – Create a stored procedure that returns the name and company of a customer.

5.0 Transactions

In this section you will be working with transactions. Transactions are usually nested within a stored procedure.

Task – Create a transaction that given a invoiceId will delete that invoice (There may be constraints that rely on this, find out how to resolve them).

6.0 Triggers

In this section you will create various kinds of triggers that work when certain DML statements are executed on a table.

6.1 AFTER/FOR

Task - Create an after insert trigger on the employee table fired after a new record is inserted into the table.

Task – Create an after update trigger on the album table that fires after a row is inserted in the table

Task – Create an after delete trigger on the customer table that fires after a row is deleted from the table.

7.0 JOINS

In this section you will be working with combining various tables through the use of joins. You will work with outer, inner, right, left, cross, and self joins.

7.1 INNER

Task – Create an inner join that joins customers and orders and specifies the name of the customer and the invoiceId.

7.2 OUTER

Task – Create an outer join that joins the customer and invoice table, specifying the CustomerId, firstname, lastname, invoiceId, and total.

7.3 RIGHT

Task – Create a right join that joins album and artist specifying artist name and title.

7.4 CROSS

Task – Create a cross join that joins album and artist and sorts by artist name in ascending order.

7.5 SELF

Task – Perform a self-join on the employee table, joining on the reportsto column.

DUE 5pm Monday, Dec. 10
Upload a .sql file to your branch in exercise-submissions with answers to each task clearly marked with question number and prompt.