

Structured Query Language

Due date: 11PM on 04/05/2018

This is an individual assignment. Do not discuss or collaborate or share your ideas/code with other students. You may not use snippets from the internet or from any other source. We will use automated and manual tools to compare submissions and test for plagiarism. Late submission is not acceptable. The submission link in Blackboard will disappear immediately at the deadline. You can submit multiple times, only the latest version will be graded. 5 points are awarded for following the handing-in instructions (see below). Each query/statement/file is worth 5 points.

1. Tools

We will use SQLite3 for this homework. If you use a *nix environment, you may install it using your favorite package manager (yum, apt, etc): `sudo apt-get install sqlite3`

Alternatively, you may install it for your windows/mac/*nix distribution by downloading the appropriate precompiled binaries from <https://sqlite.org/download.html>

We will use the Chinook data model, which represents a digital media store, including tables for artists, albums, media tracks, invoices and customers. Download this file: https://github.com/lerocha/chinook-database/blob/master/ChinookDatabase/DataSources/Chinook_Sqlite.sqlite. To view a graphical representation of this database, it's tables, columns and relationships, go to <http://schemaspy.org/sample/relationships.html>. (Click on the 'Large' tab to see the full table view.)

To open this database, at the command prompt run: `sqlite3 Chinook_Sqlite.sqlite`

Use the `.help` menu and the SQLite documentation at <https://sqlite.org/lang.html> to learn how to navigate and use the system.

There are several SQLite graphical user interfaces available for various distributions. You may download and use one of them if you would rather not use the command line. This is not required.

2. HANDING IN

For each of the following queries, create a file named `query_n.sql`, where n is the question number and store your query in the file (remember to include the semicolon). We will test your query by running the following command: `sqlite3 Chinook_Sqlite.sqlite < query_n.sql`

Your result should contain the correct set of records, with the specified columns and in the exact same order as specified in the question to be considered correct. For example, if your data is correct, but the columns are mismatched or the rows are in the wrong order, it will be considered incorrect.

Create a folder with your netid (not UIN) and store all your sql files in it. Zip the folder (please do not use .gz, .rar, .7z or other formats) and upload it to BlackBoard.

3. QUERIES

The following may return zero or more results. Save each statement in a file named `query_n.sql`.

Query	Question	Expected Result Format
1	List the albums by the artists whose name start with the letter 'U' in alphabetical order (first by Artist, then by Title).	Artist Title
2	Which album contains the longest track?	AlbumName TrackName Milliseconds
3	Which employee has supported the most customers?	EmployeeId FirstName LastName Title NoOfCustomersSupported
4	List the top 10 albums that have the most number of tracks. Who are they by?	ArtistName AlbumTitle NoOfTracks
5	List the artists that do not have any albums.	ArtistId ArtistName
6	Which employee has the greatest number of immediate subordinates?	EmployeeId FirstName LastName Title NoOfReports
7	Which customer spent the most amount of money? How many tracks did they buy?	FirstName LastName TotalMoneySpent NoOfTracks
8	Which month generated the highest revenue? Hint: you have to multiply quantity and unit prices.	Month(yyyy-mm) TotalRevenue
9	List the top 10 artists with the most number of tracks in playlists.	ArtistName NoOfTracks NoOfPlaylists
10	List the artists that have tracks in 6 or more playlists, ordered by the number of tracks (descending)	ArtistName NoOfTracks NoOfPlaylists
11	BONUS Question (10 points): List all the direct and indirect reportees of Andrew Adams, ordered by their Employee ID. Although the sample contains a small set of records, your query should be able to extract an arbitrary number of levels of hierarchy. Hint: Use recursive common table expressions.	EmployeeId FirstName LastName Title

4. DATA WAREHOUSE

We would like to build a simple Star Schema that supports analysis of sales by date, location and track. Create the following Fact and Dimension tables. All IDs are integers. For locations, we're interested in the customers' location, not the billing address.

Create the following tables and populate them with data using CREATE TABLE AS SELECT statements. Store each query in separate `olap_tablename.sql` files.

Fact_Sale	Dim_Date	Dim_Location	Dim_Track
dim_sale_id	dim_date_id	dim_location_id	dim_track_id
dim_date_id	date_value datetime	city nvarchar	track_name nvarchar
dim_location_id	day integer	state nvarchar	album_name nvarchar
dim_track_id	month integer	country nvarchar	artist_name nvarchar
unit_price numeric (10,2)	year integer	postalcode nvarchar	composer nvarchar
quantity integer			genre nvarchar
			mediatype nvarchar

Let's perform some analysis and generate reports. Save each statement in a file named `query_n.sql`.

Query	Question	Expected Result Format
12	Generate a monthly revenue report. (revenue = total sales) Hint: you have to multiply quantity and unit prices.	Year Month Revenue
13	For each artist, identify annual track sales counts in each city	Artist Year City TotalTrack sSold

(15 points) **Analysis:** Slice and dice the data in the OLAP tables that you just created and report on at least *three* trends that you notice in the data. Write approximately 100 words to describe each trend that you identify. Save your file as `trends.txt`.

FINAL DELIVERABLES

`trends.txt` (15 points), `query_1.sql` ~ `query_13.sql`; `query_11.sql` is bonus (10 points).

`olap_fact_sale.sql`, `olap_dim_date.sql`, `olap_dim_location.sql`, and `olap_dim_track.sql`.

Each file is worth 5 points unless otherwise specified.