In-class Lab 01

T-SQL Fundamentals Chapter 01

Introduction to SQLite

Using the CLI

How to use the command line. A refresher on common DOS commands. How to invoke the SQLite database.

Using internal database commands

How to use the internal commands the SQLite database provides.

Executing SQL statements

How to write and execute simple SQL scripts. How to use the input and output redirection operators.

SQL helps

Some useful references and resources for understanding and learning SQL.

Graded Labs — Writing and Executing SQL Scripts

This lab is an introduction to using a database. The objective is to familiarize you with the concept of writing and executing SQL scripts, and saving the output. We will use n easy, simple, and free database engine, SQLite. However, the SQL that you use is ANSI SQL that will run (perhaps with sight modifications) on any relational database management system, including Microsoft SQL Server. In a sense, ISTA-420 is merely an elaboration and explanation of this assignment.

Creating a database table

This lab requires you to write a SQL script, run the script, and direct the output to a text file. Your SQL script should b e named something like this — `mySQLscript.sql`, and your output file should be named something like the — `mySQLscript.txt`. Note that the file names are the same but the file extensions are different. Create a new database, named whatever you want. In that database, create a table, named whatever you want, subject to the following constraints. The subject matter of the table should be a domain that you have good familiarity with, such as books, music, firearms, motor vehicles, your employment history, your educational history, etc. In that table include at least fice columns, as follows:

1. A column that is both unique and ot null. This is your *primary key.* If desired, you can make this an *autoincrement* column.
2. One or more columns of type *text.* Examples include composer, performer, name, and title.
3. One or more columns of type *integer.* Examples include salary, height, age, and weight.
4. One or more columns of a text *categorical* variable. *Categorical variables* have repeating values. Examples would be the manufacturer and model of automobiles, ie.e, Ford Taurus, Honda Accord, and BMW 320i.

5. One more more columns of an integer categorical variable. Examples would include the number of cylinders and forward speeds of automobiles.

Inserting data

Write several insert queries, four or more, inserting data into your table.

Writing simple queries

You will write and execute queries from scripts, and redirect the output to text files. You will join two or more tables, filter data by given parameters, and sort data by specific variables. You are not expected to understand these queries (yet!) — just to run then and understand the output.

Deliverable

Write a SQL script describing your family (or unit), similar to the exercise we did in class. Complete this with SQLite. Use the input redirection operator and the output redirection operator to produce a text file containing your SQ: commands and the output. Here is the sample we did in class.

```
.echo on
.headers on

--Name: myfamily.sql
--Author: Charles Carter
--Date: February 21, 2018

drop table if exists family;

create table family (
  id integer,
  name text,
  sex integer,
  role text,
  age integer
);

insert into family values (1,'Charles',1,'parent',68);
insert into family values (2,'Bonnie',0,'parent',49);
insert into family values (3,'Casie',0,'child',38);
insert into family values (4,'Jackson',1,'child',30);
insert into family values (5,'Midnight',0,'pet',3);
insert into family values (6,'Max',1,'pet',8);

.schema family

select * from family;
select * from family where sex = 1;
select * from family where role like 'parent';
select * from family where role like 'pet';
select name from family;
select avg(age) from family;
select sum(age) from family;
```