**Increment #1**

**Custom Guitar Builder**

**Database Building**

**By Dylan Oszust**

**Spring, 2017**

**Table of Contents:**

Overview ----------------------------------------------pg. 3

Installing and Setting Up Software --------------pg. 4

The .SQL Files -----------------------------------------pg. 9

Running The .SQL Files ------------------------------pg. 10

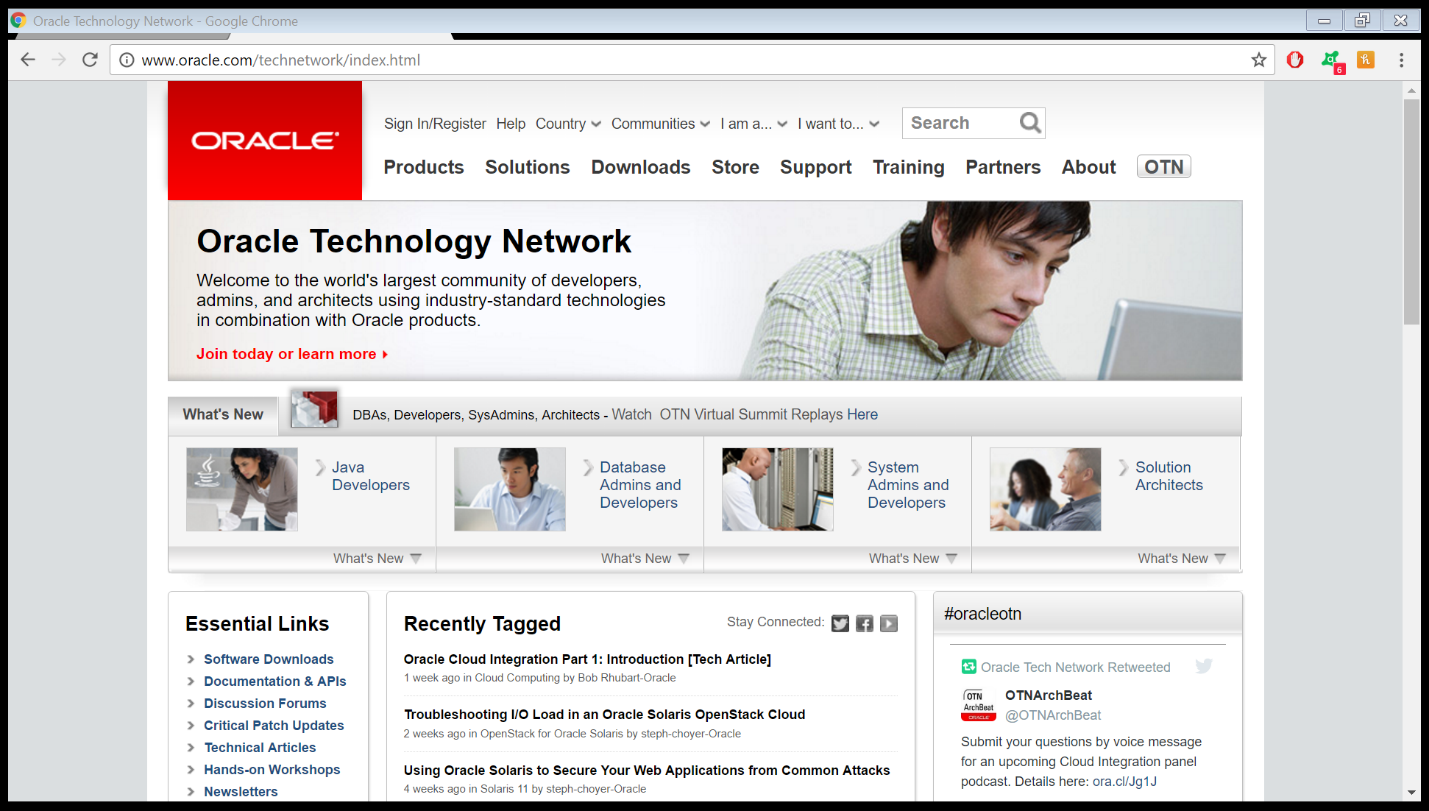
Database Overview

This database has been created for use with the Guitar Builder application. There are seven tables in the database, the tables are Customer, Neck, Body, Hardware, Cosmetics, and Product. The Customer table will hold information about the customers, including Customer ID, First and Last name, Phone#, Address, and Email. This table will also have a field to indicate rather or not the customer is a return customer. The Neck table will hold information on how the neck of the guitar is to be built. It will hold the Neck ID, Neck wood, Fret wood, Scale Length, Number of Frets, Fret Size, and the type of joint used to attach the neck to the body. The body table will describe the body ID, as well as the body wood, top wood, body shape, rather or not it is a carve top and rather or not it is a hollow body. The Hardware is going to describe what hardware is going into the guitar, including the pickups, knobs, and tuners. There will also be a Hardware ID used to identify the setup described. The cosmetic table will assign a cosmetic id to the configuration chose, as well as describe various paint options, fret marker options, and hardware/binding color options. Finally, the Product Table will have a Product ID for each guitar being made, and will assign the ID from each table (Neck, Body, Hardware, and Cosmetics.) This will keep track of which parts go to which product being ordered, and will contain a Customer ID to identify which customer the custom build is for.

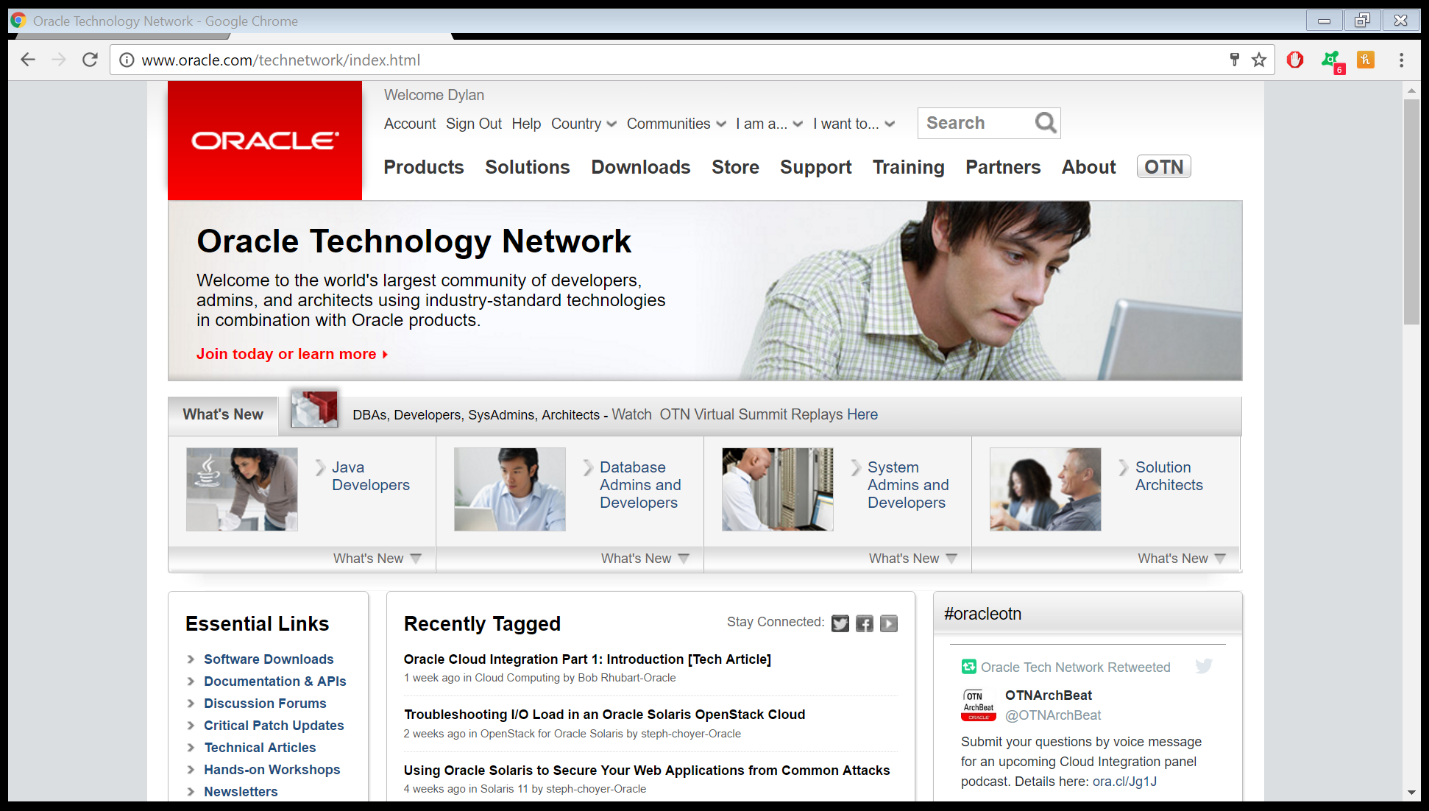
**Installing and Setting Up Necessary Programs.**

This application uses a database that runs through OracleDatabase 11g Express Edition. Follow these instructions to download and set up this software.

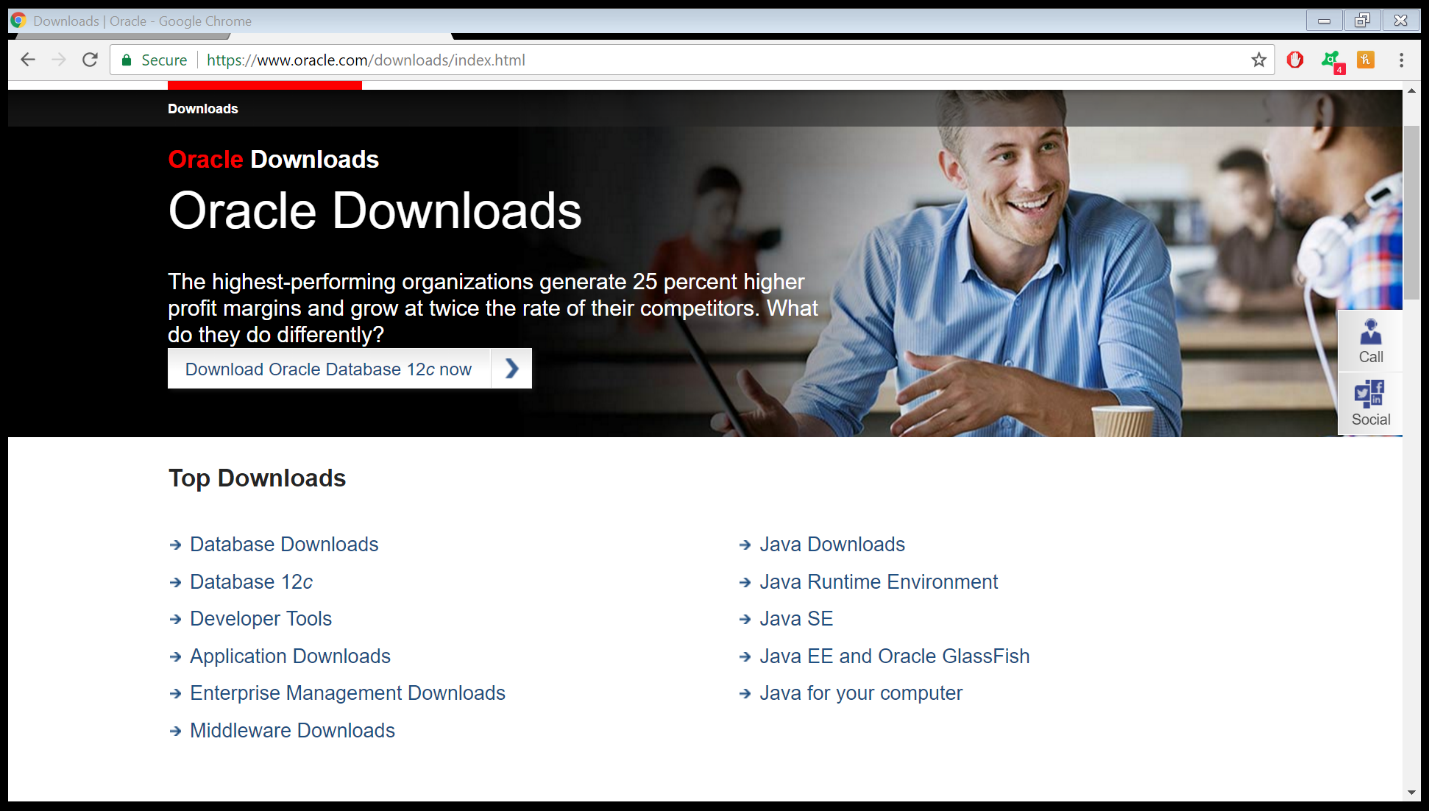
1. Go to the Oracle website: [http://www.oracle.com/technetwork/index.html](http://www.oracle.com/technetwork/index.html )
2. Set up an account, or log in with an existing account:



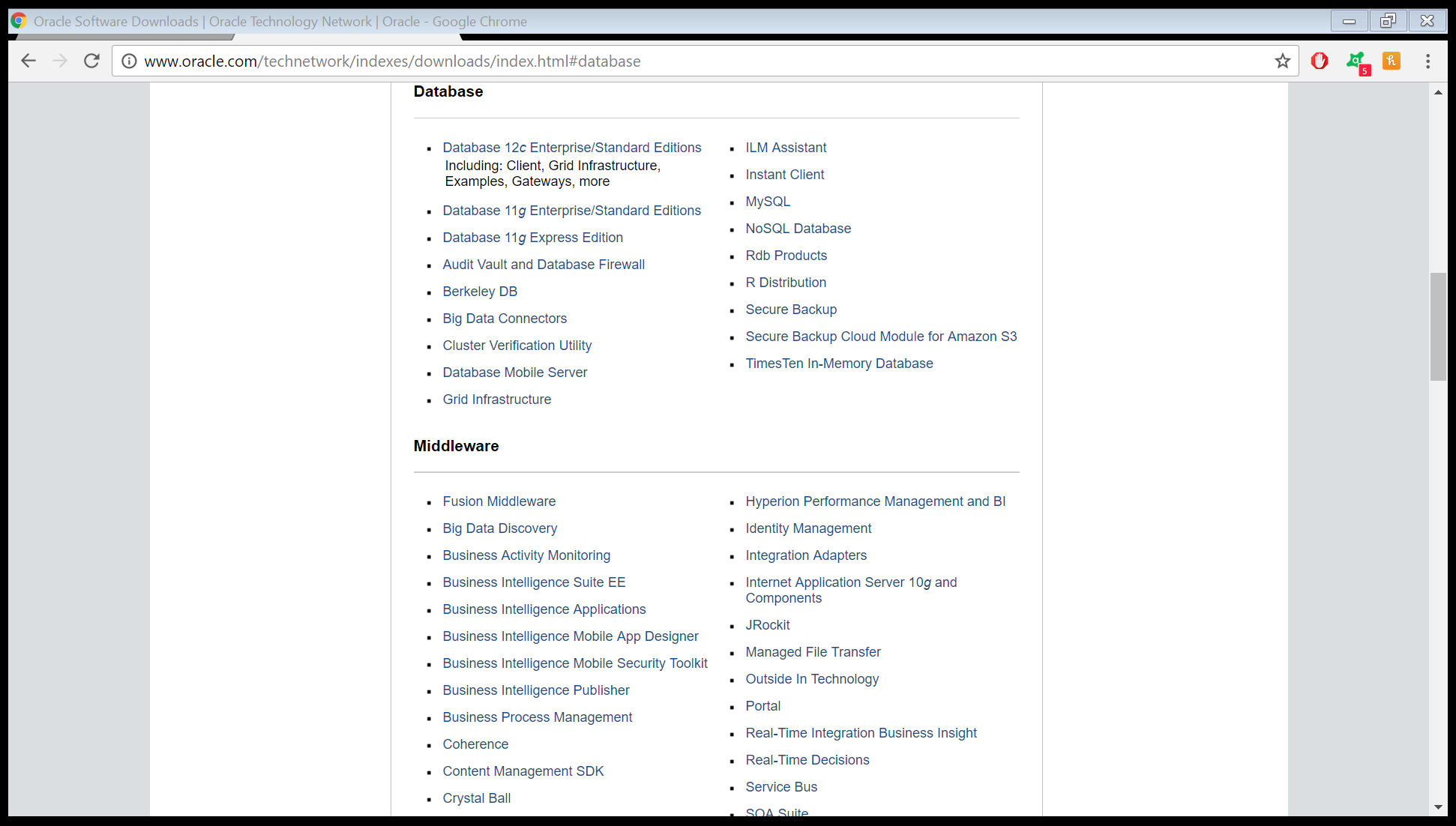
1. Once signed in, go to Downloads:



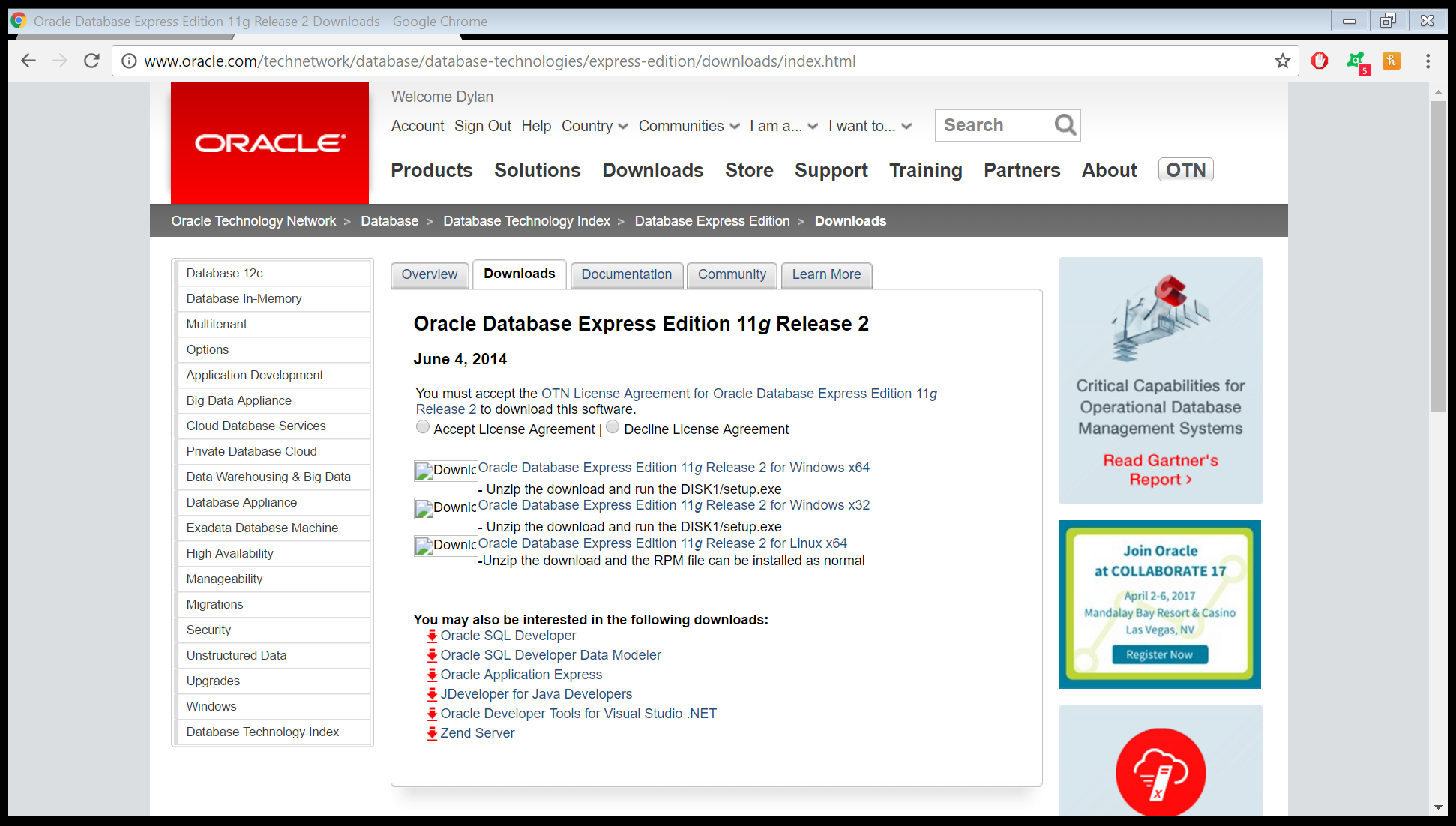
1. Click “Database Downloads”:

****

1. Click Oracle Database 11g Express Edition:

****

1. Click the “Accept License Agreement” and download the version for your pc:

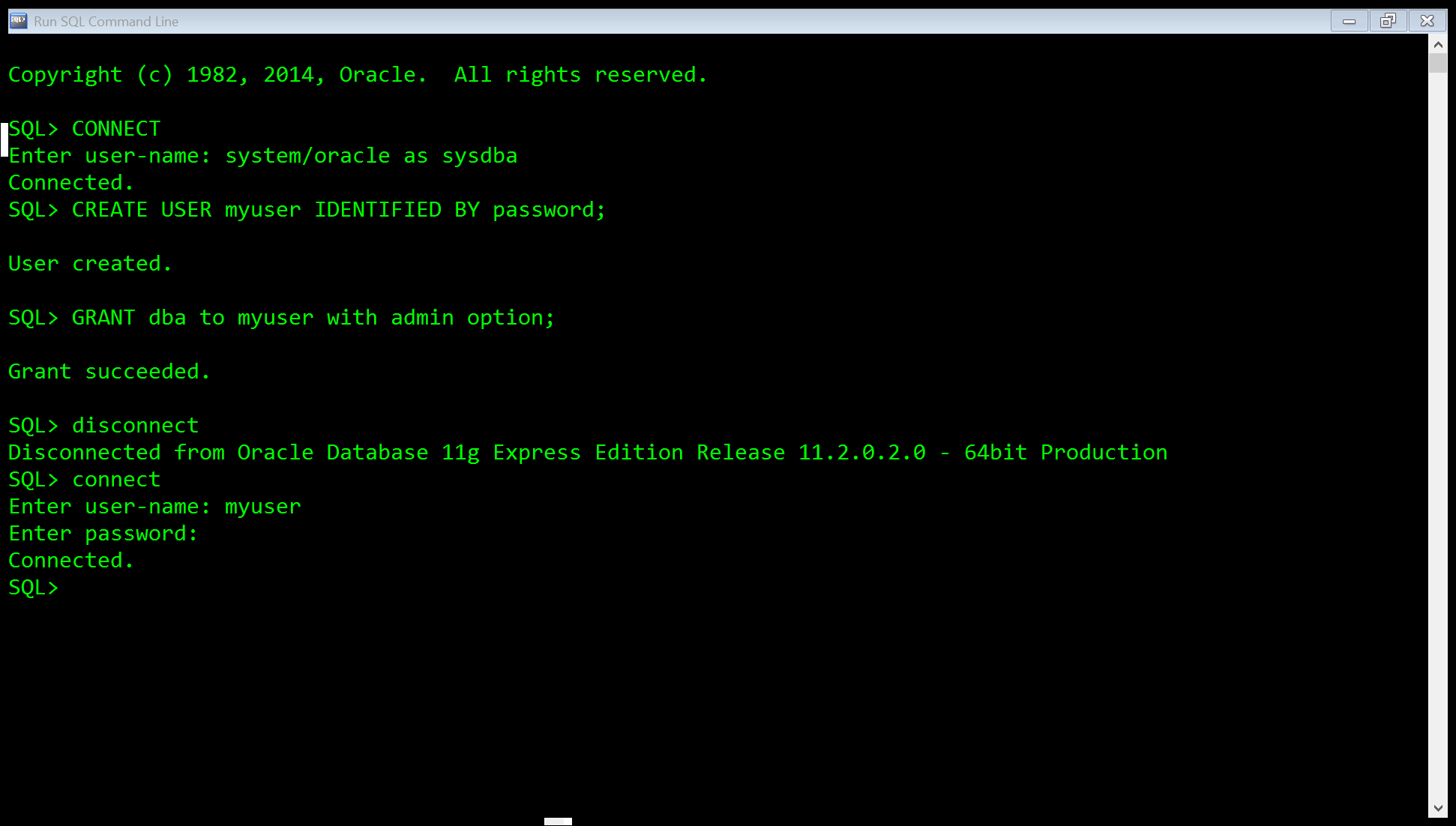


1. Follow the prompts after downloading the software for installing. Then click “Run SQL Command Line” that should be on your desktop.
2. In the SQL Command Line, you will setup your own Username & Password. To do this type “Connect” then “system/oracle as sysdba”.
3. Type in

“CREATE USER [your first name--user name- lower case - do not include brackets]” IDENTIFIED BY [your password];

Example: CREATE USER myuser IDENTIFIED BY password;

1. At prompt, type “GRANT dba to (username) with admin option;
2. After grant successful, type disconnect, then connect again, this time using your own username and password. This is displayed below: (note as a safety feature, your password will not display as you type it in, it is still entering information though.)



Step 11

Step 10

Step 9

Step 8

1. Now that you are connected you may run the .sql files in the zip package.

**The .SQL files.**

In the zip folder, there is a folder called “CustomGuitarBuilderDatabaseSQL”. This folder contains a “ReadMe” and five .SQL files. The Read Me file gives a short overview of the .SQL files included.

The files are listed below with the short description that is included in the readme.

**FILE Custom\_Guitar\_Database\_Build.sql:**

This SQL file will build the Custom Guitar Database, which will then be ready to create, read, update, and delete data to.

**FILE Custom\_Guitar\_Database\_Test\_Data.sql:**

This SQL file will enter test data into the database which will then be readable to show an example of the records that may be inserted into the database.

**FILE Custom\_Guitar\_Database\_Clean.sql:**

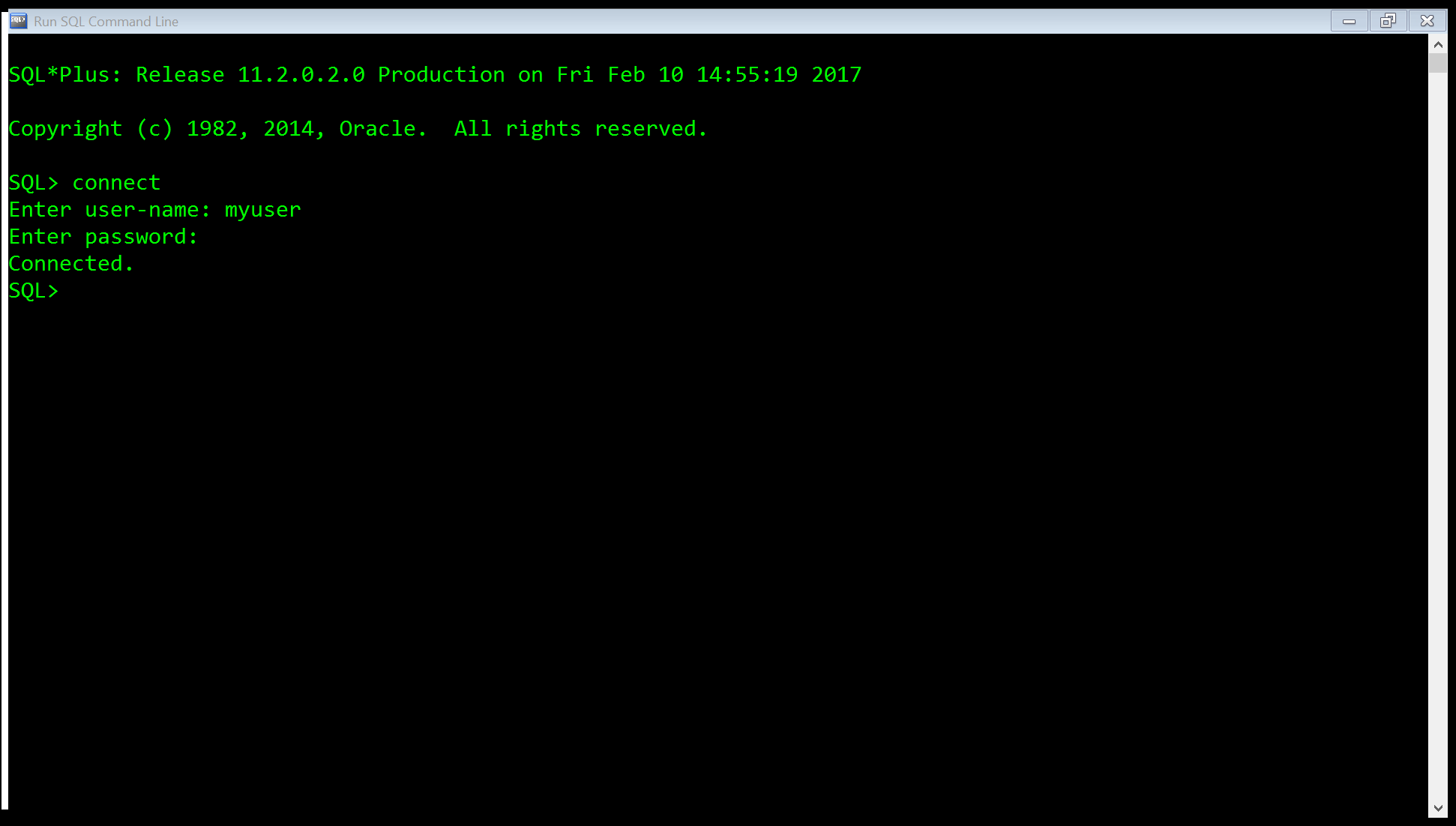
This SQL file will clean all of the data from the database and empty all of the tables, effectively giving a new instance of the system.

**FILE Custom\_Guitar\_Database\_DESTROY.sql:**

This SQL file will destroy the database, deleting the information, along with the tables and the database itself.

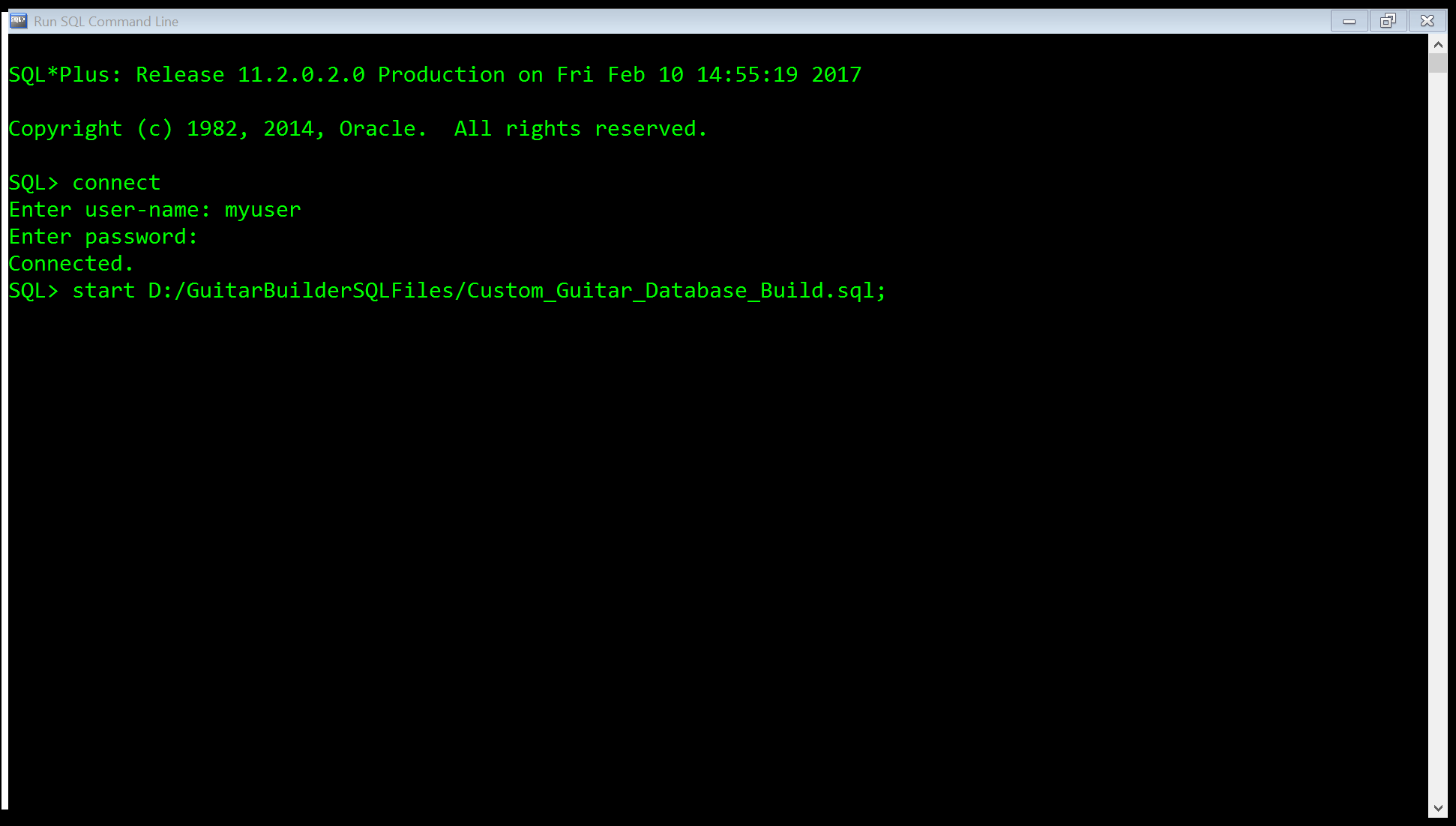
**Running the .SQL files.**

1. The first step is to unzip all of the .SQL files and place them in a folder whose you will remember. My suggestion is to save the folder somewhere as shallow in your hard drive as possible. For instance, create a folder directly in the hard drive named GuitarBuilderSQLFiles, that will store the .SQL files. This will create a short folder location address (C:/GuitarBuilderSQLFiles). This is important because to access the files through the Database Management system, you will need to remember this location.
2. The next step is to connect to the database system as described in the previous set of instructions. The screen should look like this:

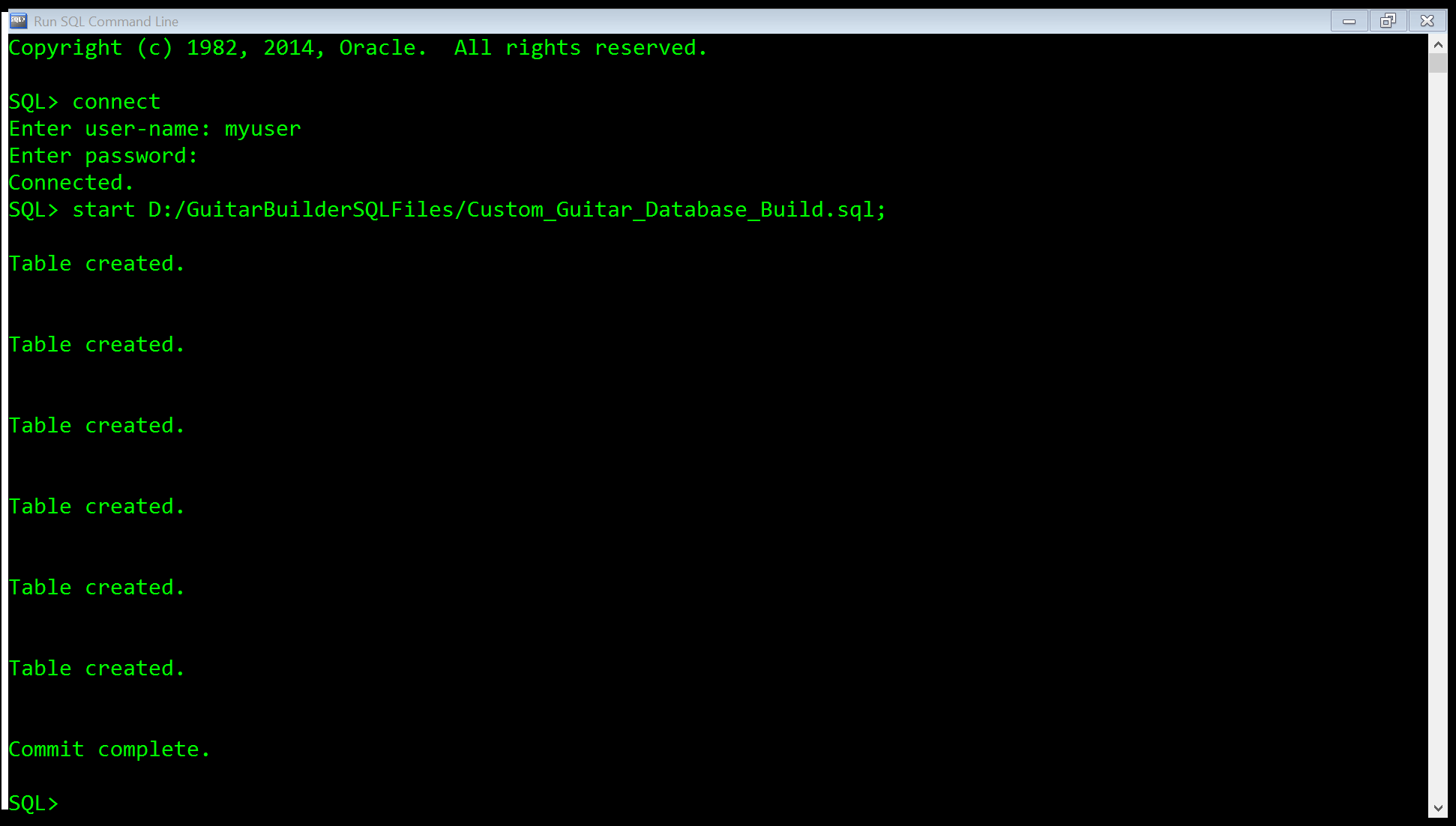


1. Finally, you are ready to run the first .SQL file. The first file you run will build the database. This is named “Custom\_Guitar\_Database\_Build.sql”. To run it type the command “start” followed by the file address. If you saved it how I described above, it should look like:

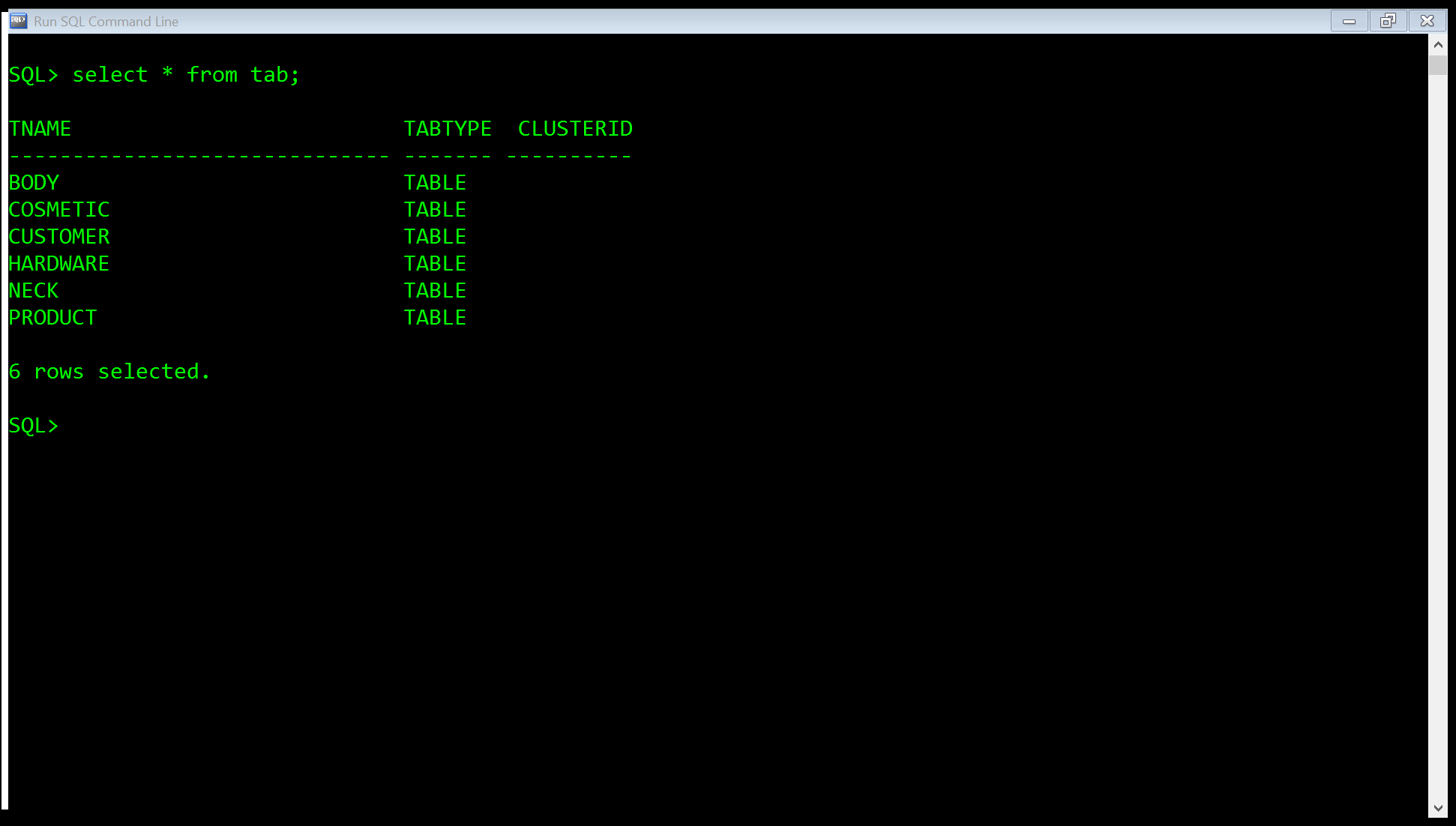
“start C:/GuitarBuilderSQLFiles/Custom\_Guitar\_Database\_Build.sql;”



1. After hitting enter, the Database Manager will build the database.

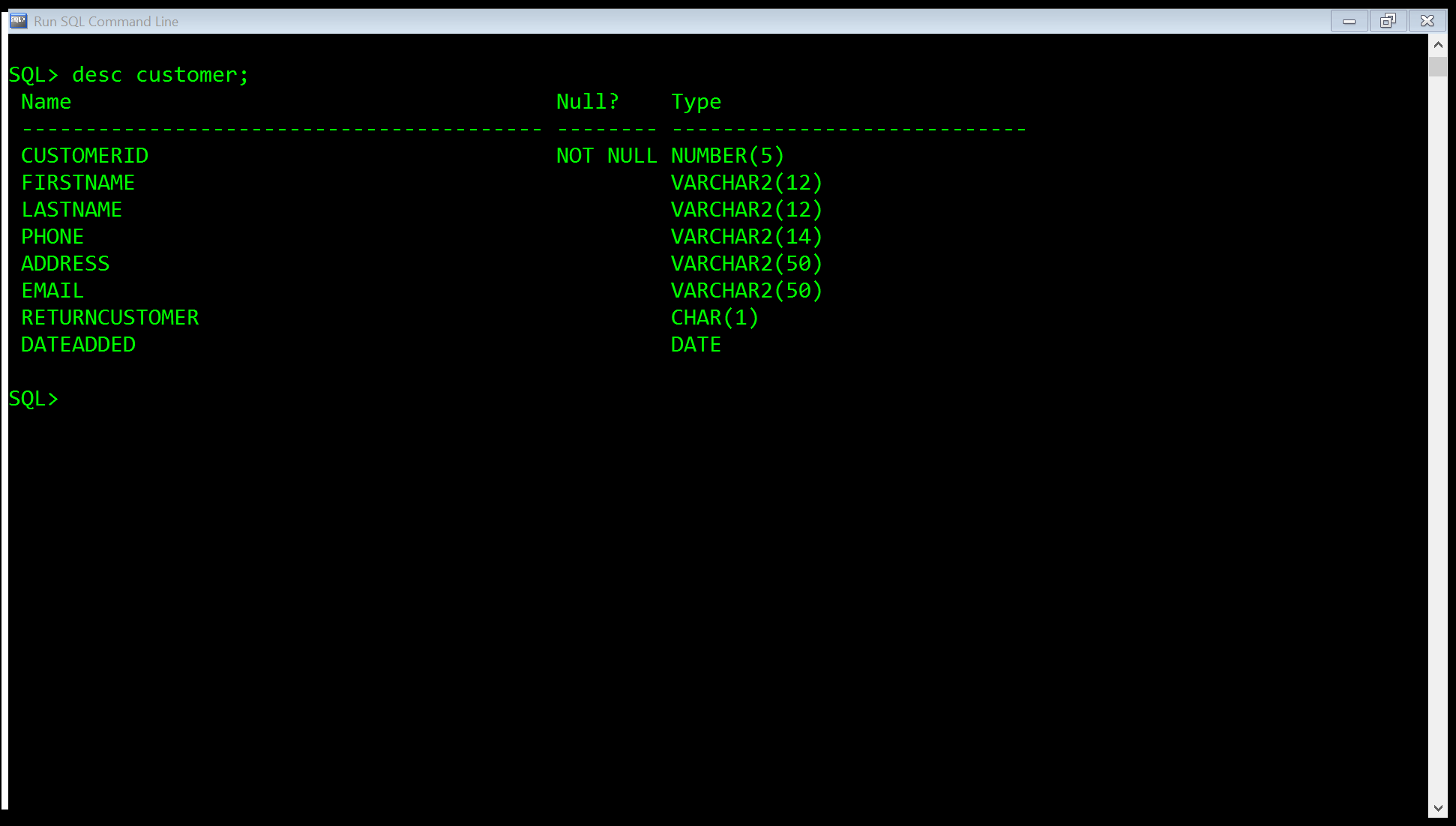


1. To View the tables, type “select \* from tab;”



1. To view the details of each table, enter the command “desc (table name);” where table name is the name of the table you want to view. This will tell you which fields are available for each table, and what type of input they accept.

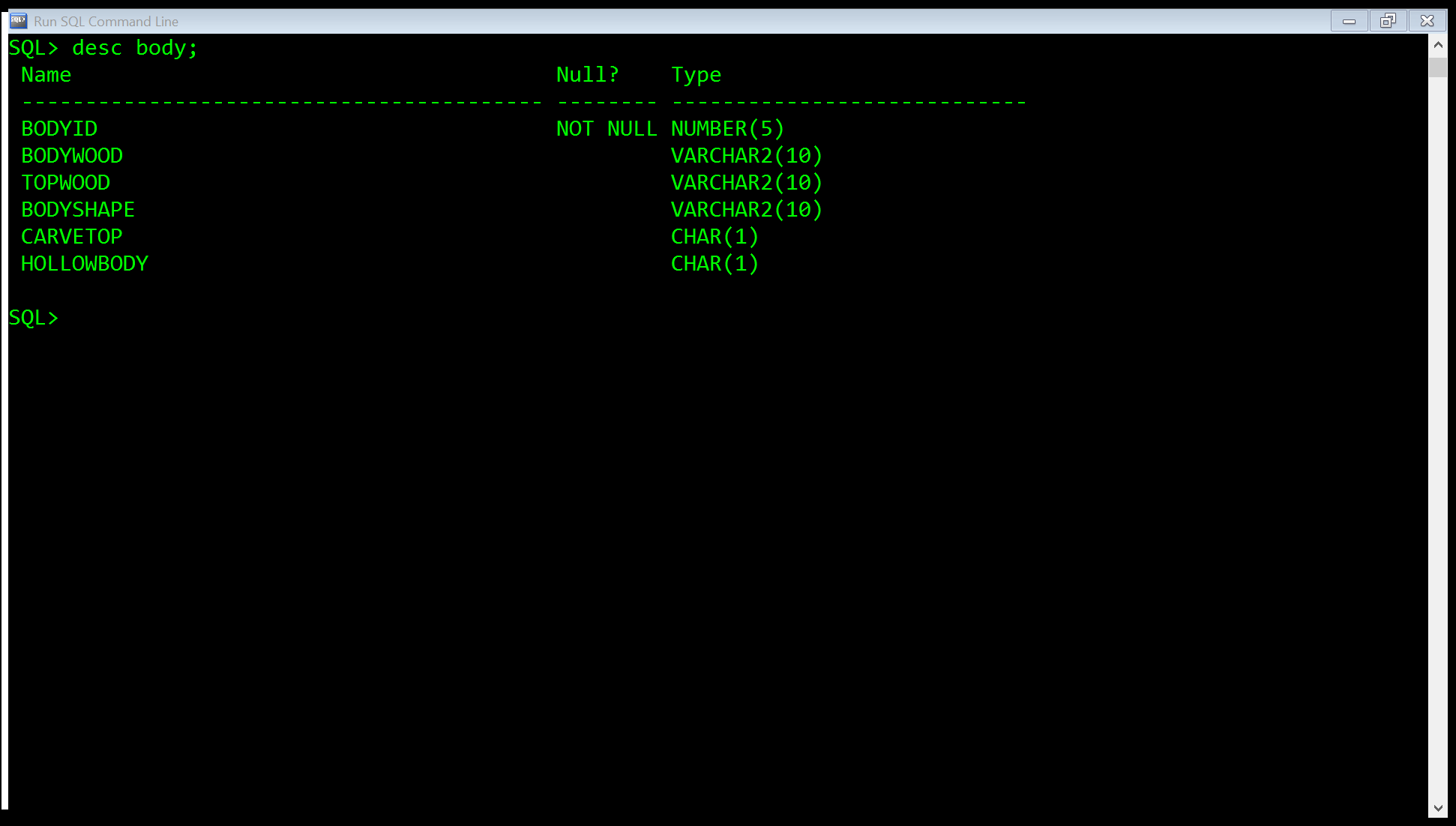
The Customer Table:



The customer table accepts the following input:

* 5-digit number for the customer ID
* 12 characters for a first name
* 12 characters for a last name
* 14 characters for a phone number formatted 1(234)567-8900
* 50 characters for an address
* 50 characters for an E-mail address
* A (Y/N) for return customer
* And a date for when the customer was added. This defaults to the current system date when you add a customer.

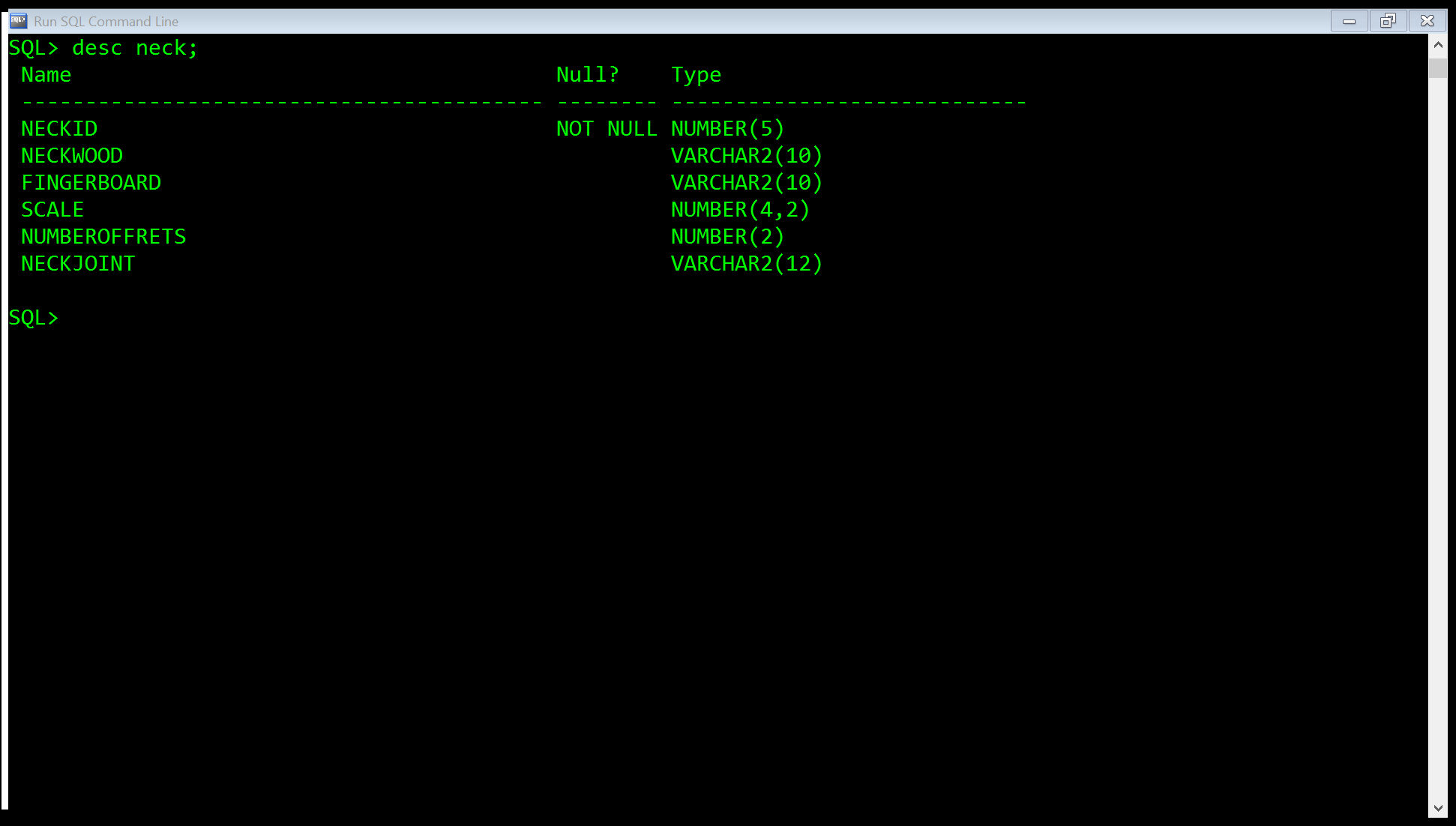
The Body Table:



This table accepts the following input:

* 5-digit number for body ID
* 10 characters for a body wood type
* 10 characters for a top wood type
* 10 characters for a body shape type
* A (Y/N) value for carve top option
* And A (Y/N) value for hollow body option

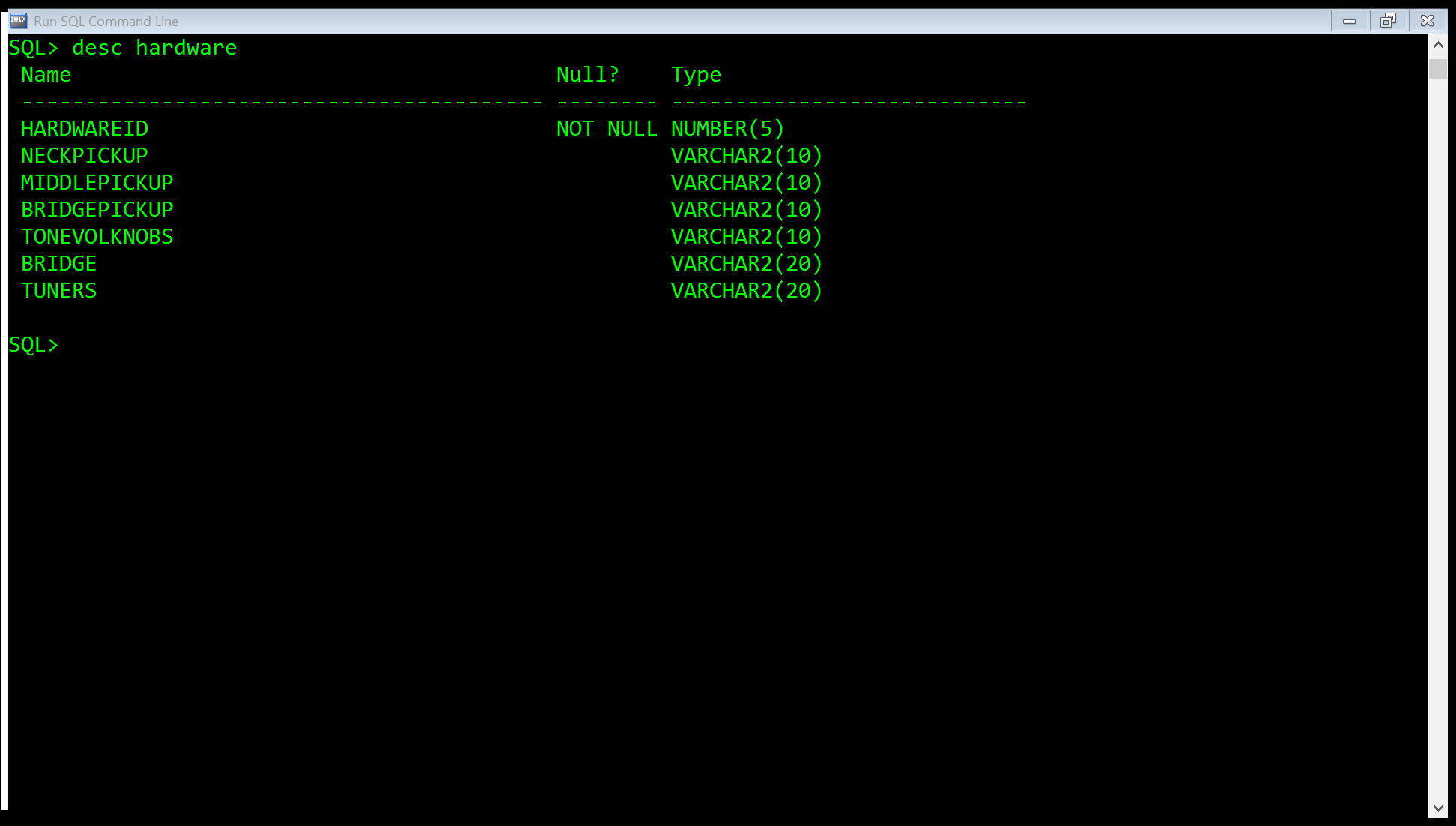
The Neck Table:



This table accepts the following input:

* 5-digit number for neck ID
* 10 characters for a neck wood type
* 10 characters for a finger board type
* A number formatted ##.## for the scale
* 2-digit number for number of frets
* 12 characters for a neck joint type

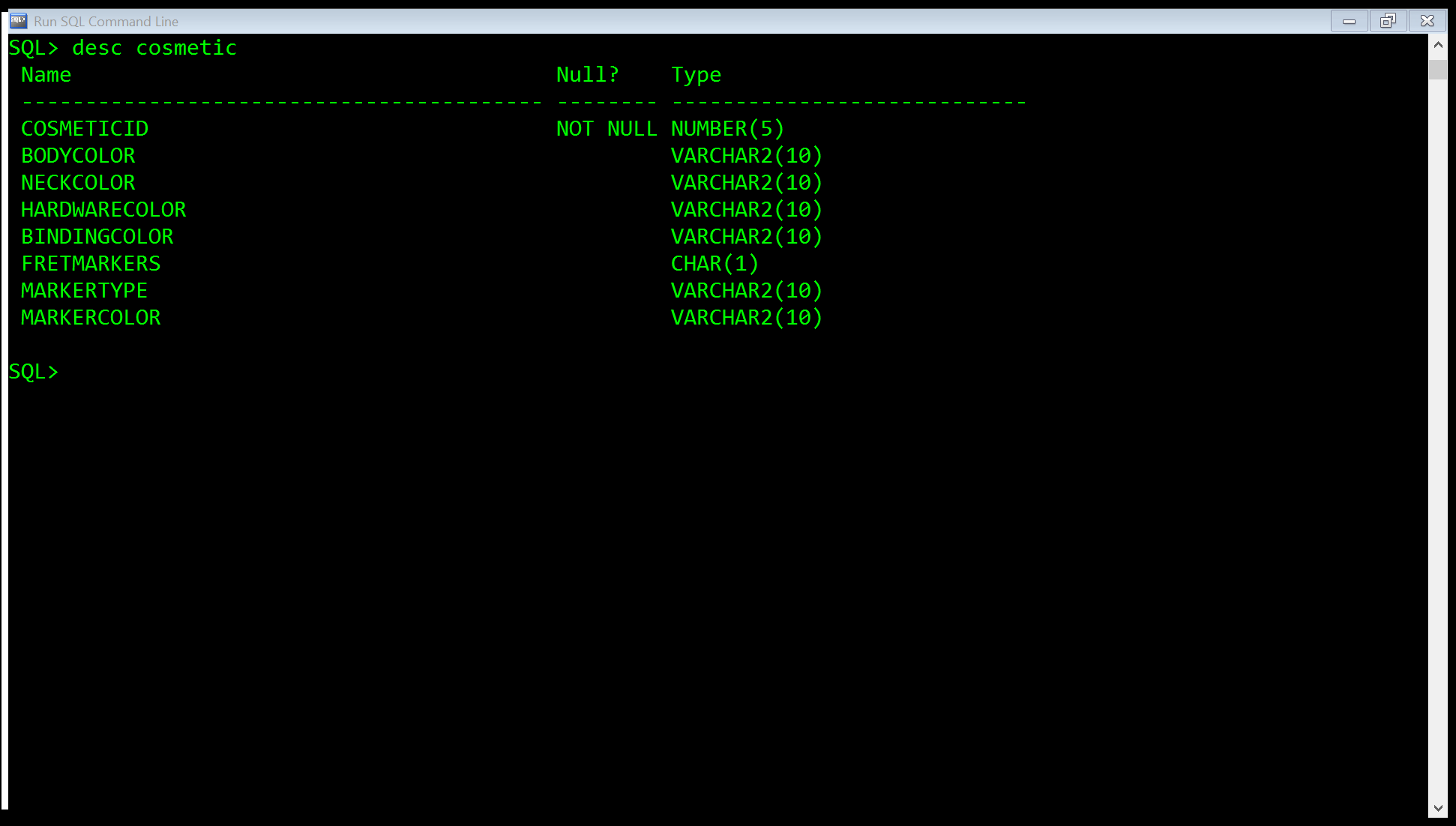
The Hardware Table



This table accepts the following input:

* 5-digit number for hardware ID
* 10 characters for a neck pickup type
* 10 characters for a middle pickup type
* 10 characters for a bridge pickup type
* 10 characters for tone/volume knobs type
* 20 characters for bridge type
* 20 characters for tuners type

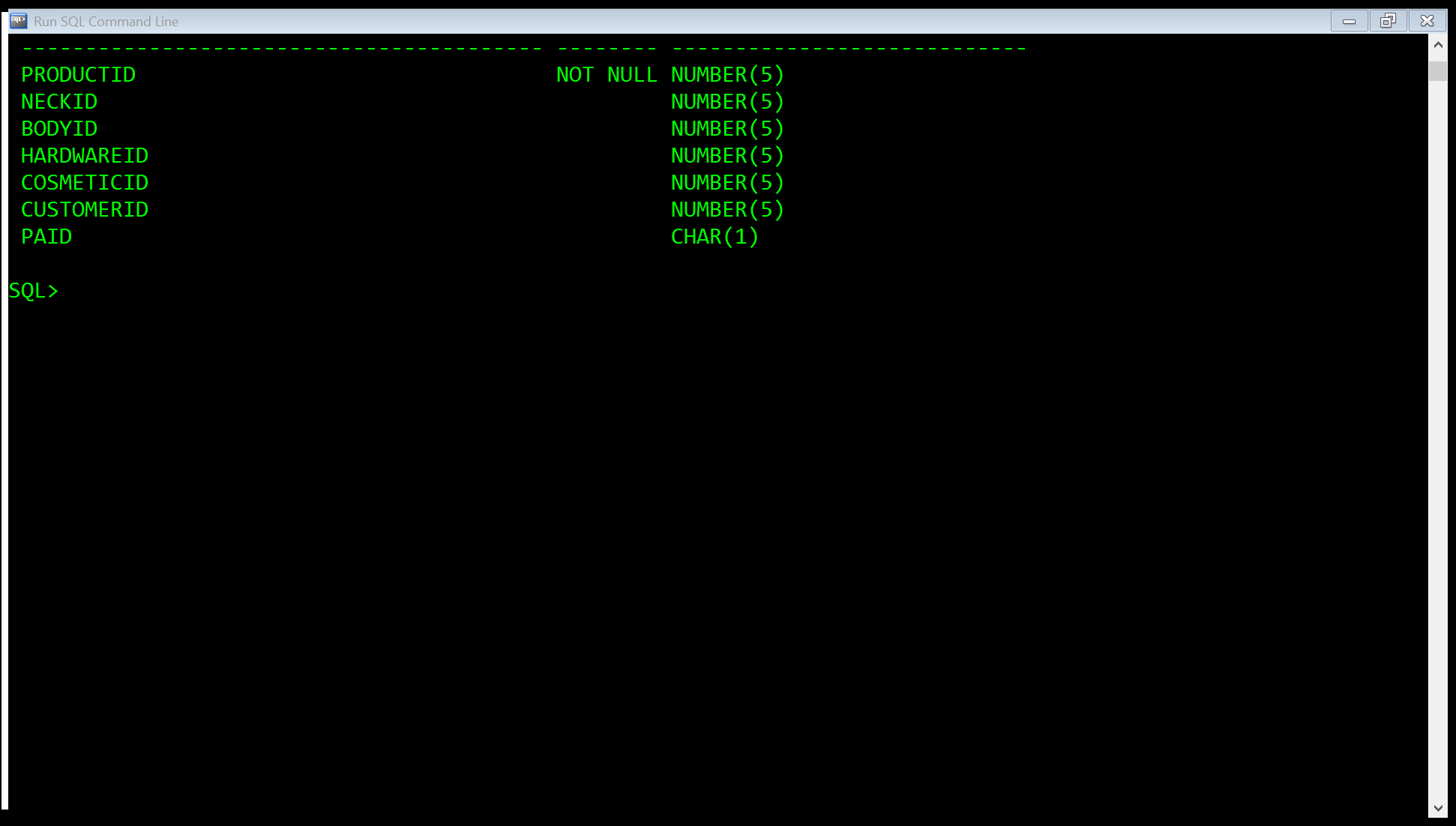
The cosmetic table



This table accepts the following input:

* 5-digit number for cosmetic ID
* 10 characters for a body color
* 10 characters for a neck color
* 10 characters for hardware color
* 10 characters for binding color
* A (Y/N) character for the fret marker option
* 10 characters for a fret marker type
* 10 characters for a fret marker color

The Product Table

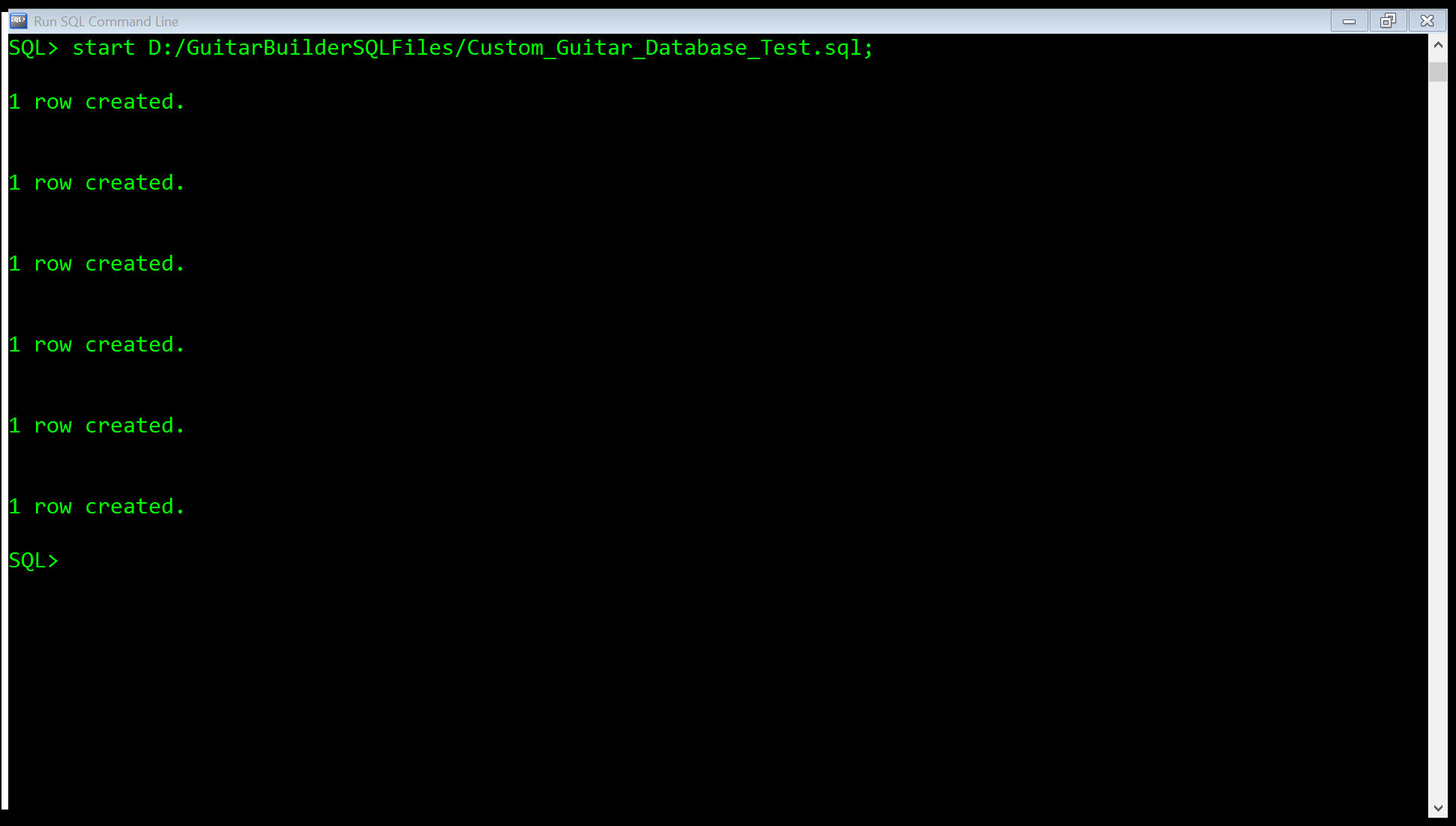


This table accepts the following input:

* 5 characters for a product ID
* 5 characters for a neck ID
* 5 characters for a body ID
* 5 characters for a hardware ID
* 5 characters for a cosmetic ID
* 5 characters for a customer ID
* A (Y/N) character for rather the product has been paid for or not.

Each ID (except the product id) is a foreign key that references an object that already exists. The ID for each item in the product table, must be an ID to an item that exists in the other tables. For instance, the Hardware ID must be an ID that already exists for Hardware ID in the Hardware table. This is to organize the parts of the guitar that go together, and identify which customer the product belongs to.

1. To test the database, start the file that is in your folder labeled Custom\_Guitar\_Database\_Test.sql the same way you started the database builder.

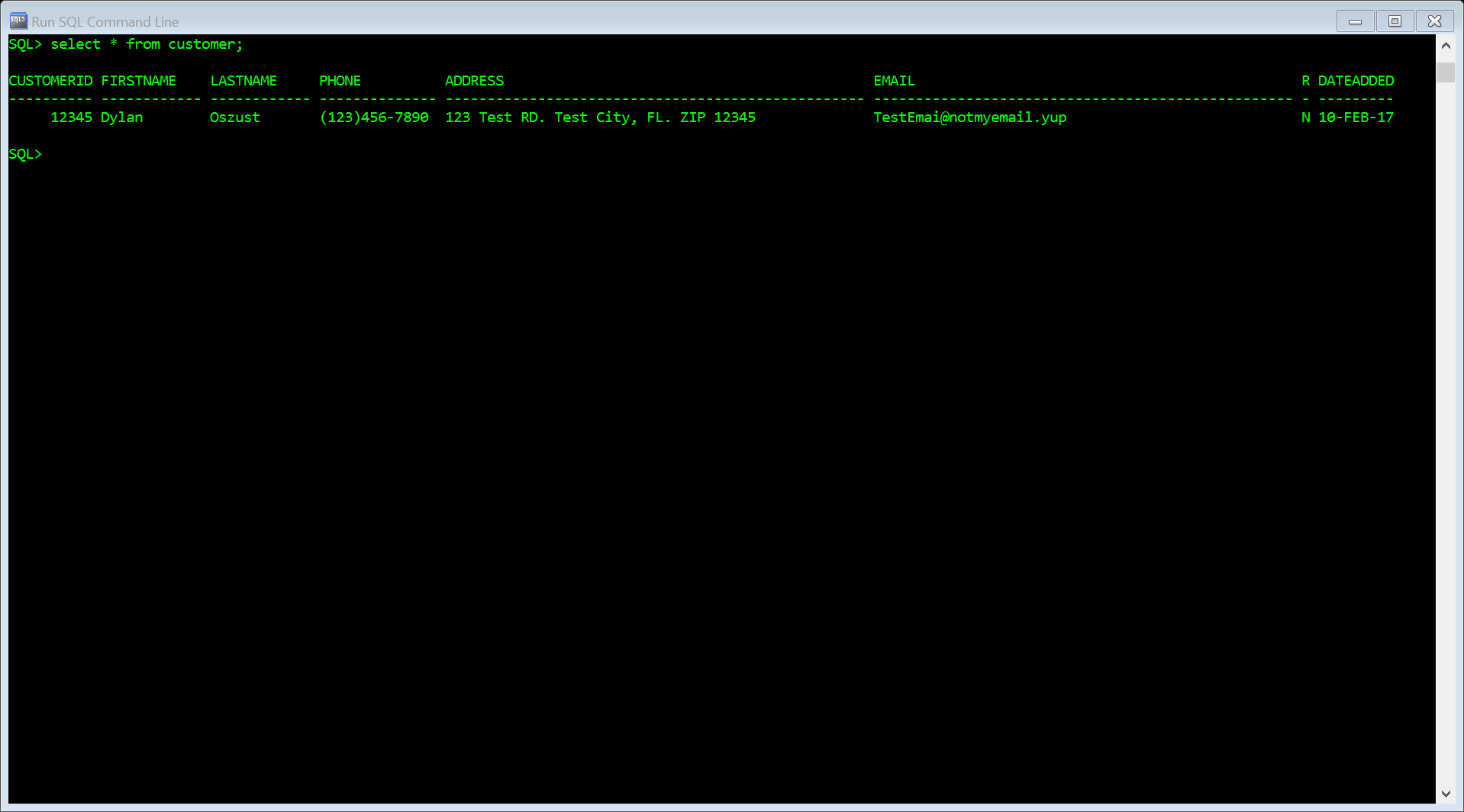


This inputs data in the table to show how the tables display data.

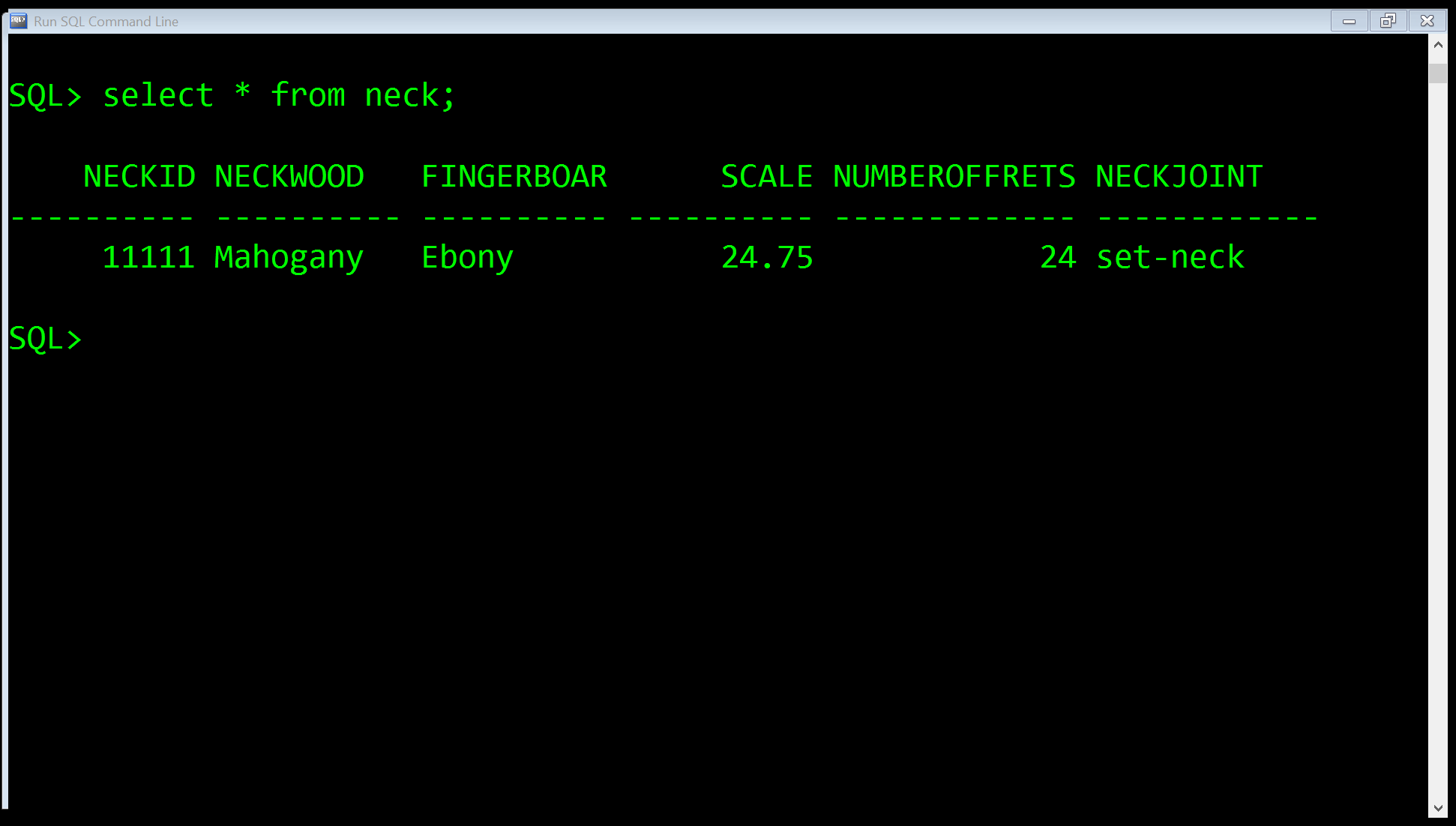
To view the data in each table, type the command:

“select \* from (table name)” where table name is the table you want to view.

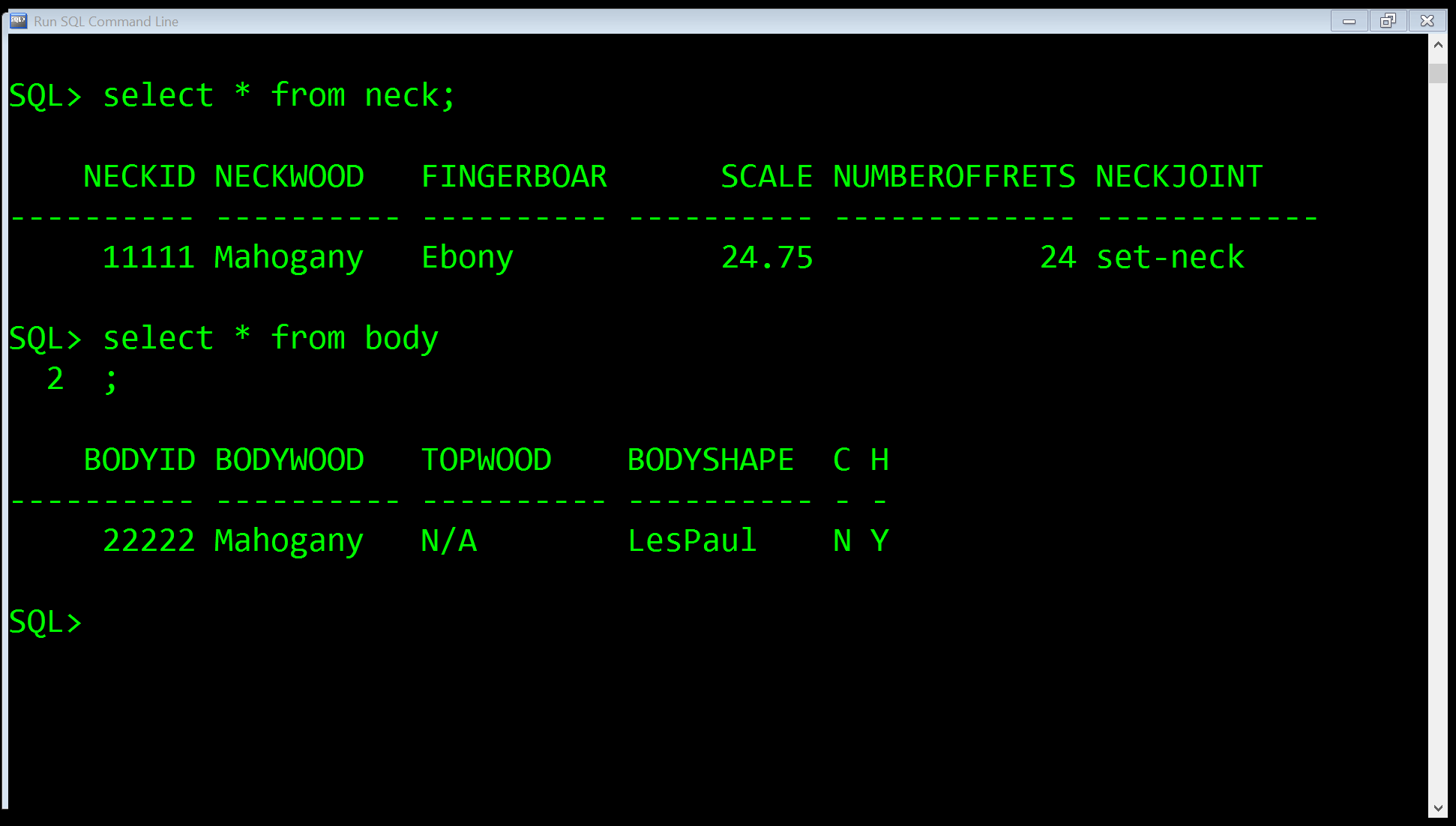
The customer table



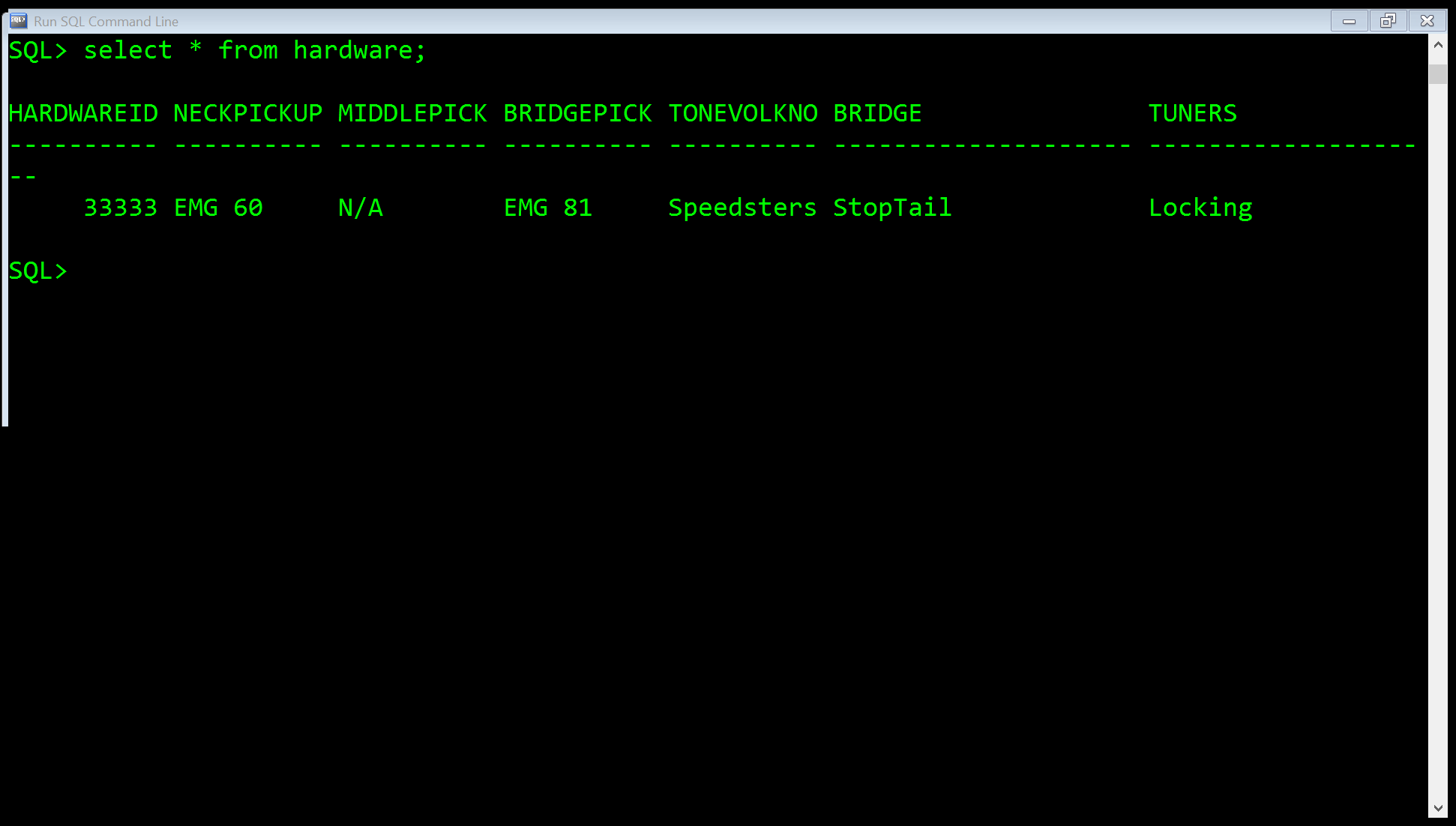
The neck table



The Body Table



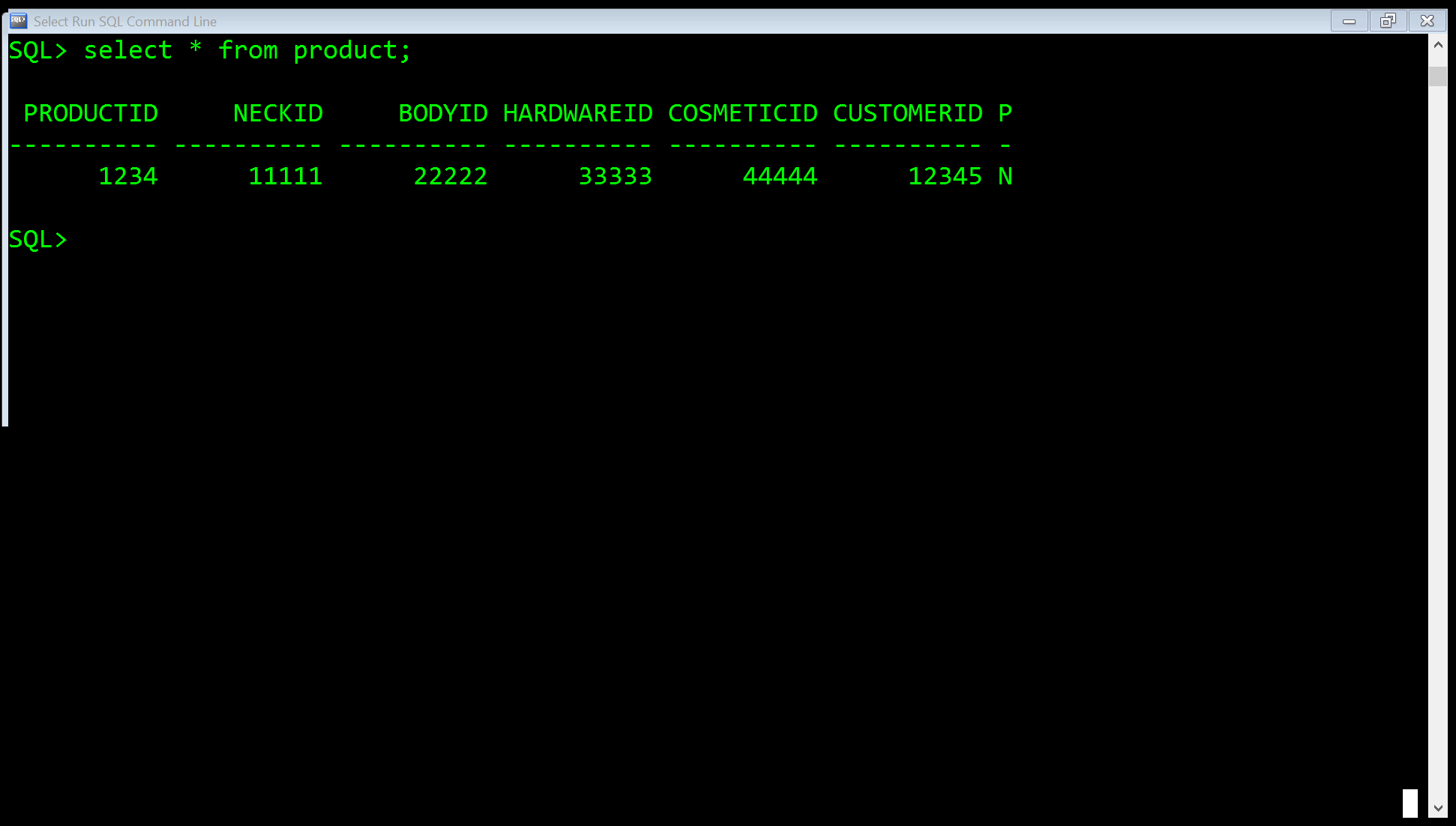
The Hardware Table



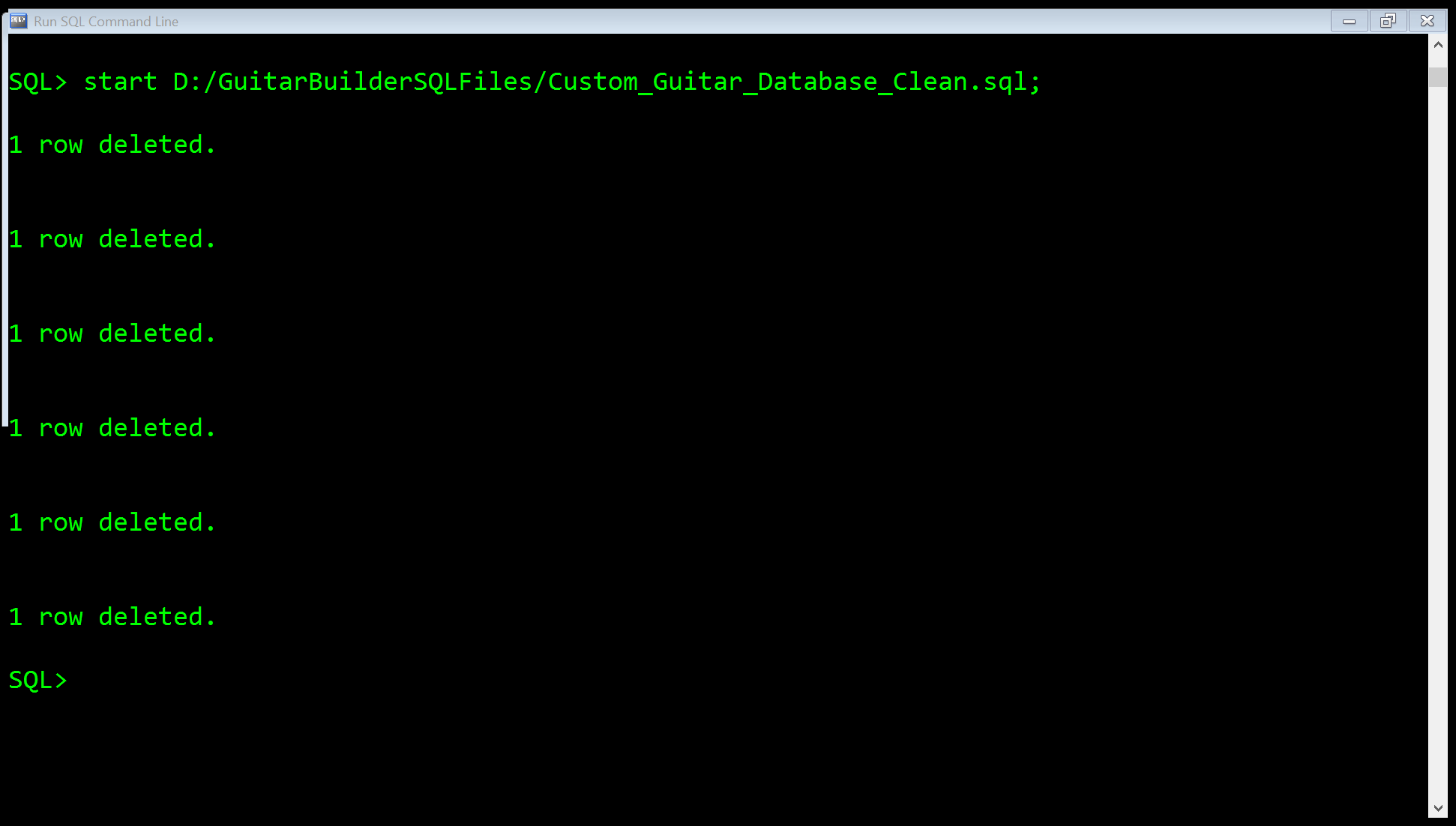
The Cosmetic Table



The Product Table



1. Finally to clean the database of the data, run the Custom\_Guitar\_Database\_Clean.sql file. This will erase all data in the database so it may be brand new to use with the application.



Your database is now ready to use. Enjoy!