***************** Write the following Stored Procedures: *****************

1: **CourseList()**
The Procedure lists all the courses in the database in alphabetical order.

2: **SingleCourse()**
Lists information about a single course.
The course number is passed as a parameter

3: **NewCourse()**
Consult the **ERD** image to find out which data should be inserted(through parameters)
The procedure has an **OUT** parameter number_of_inserted_rows that after execution contains
the number of rows that were actually inserted into the database.
To find that out you can use the MySQL function row_count()
You run that function immediately after the insert statement!

4: **UpdateCourse()**
Here you use the same parameters as in 3. The course number is used here to update the right course.
Use the row_count() function here just like you did in the previous problem.

5: **DeleteCourse()**
The course number is used to delete the right course.
Note: If the course is in use(the course number exists in the CourseTrack table) you cannot delete it.
If it's the case that it is not being used you can delete it from the Courses table and also from
the Restrictor table.
As before a OUT parameter is used to return the number of courses that have been deleted

*****************   Write the following Functions: *****************--

6: **NumberOfCourses()**
The function returns the total number of courses in the database

7: **TotalTrackCredits()**
The function returns the total credits that can be taken on a specific track
You need to supply the track ID as a parameter

8: **MaxCredits()**
The function returns the number of credits of the course(s) with the most credits.
Please note that more than one course can share the most awarded credits but it should
NOT affect the results.

9: **MostNumberOfTracks()**
What is the maximum number of tracks that a single Division has. Just return the number(We’ll use this one later in the course)