Mandatory 3: Car Registry

02807 Computational Tools for Data Science

Submission opens:	**08:00 Saturday, October 27, 2018**
Deadline:	**20:00 Sunday, November 4, 2018**
Individual:	**This assignment must be completed individually** (see collaboration policy on course page)

Exercise

In this exercise, you will create a SQLite database for a (much simplified) car registry. It will consist of two tables; models and registrations. The models table has three columns: modelId, brand, and maxSpeed of types integer, text, and real respectively. Each row in this table describes a car model. The registrations table consists of four columns: carId, modelId, registrationYear, and price of types text, integer, integer, and real. Each row in this table describes a registered car.

Below is an example of what the two tables could contain:

Models

modelId	brand	maxSpeed
5	Audi	250
9	Toyota	220
10	Audi	240

Registrations

carId	modelId	registrationYear	price
DK12345	10	2014	900000
AA73829	9	2016	400000
BA32912	9	2017	380000
CD49132	10	2016	780000

To complete this exercise, you must download the CarRegistry.py file from the course page, and implement the missing functionality. There is a description of what each function should do in the file. All functions must use the SQLite backend to perform their functionality, and should work independently from the other functions (ie. all functions should work no matter what order the functions are called in). Furthermore, the functions should do most of there work using proper SQL. **If we observe this is not the case, we may subtract points from your solution afterwards.**

Submission You must submit your solution on CodeJudge **before** the deadline (any submissions made after the deadline will **not** be counted). Your latest submission before the deadline will be used for grading thus you **must** ensure your best solution is uploaded as the latest. Submission on CodeJudge will not be available before the submission opens time stated in the top.

Grading In order to obtain full points for this assignment, you must implement all the described functions in a correct and efficient way. From submission opens until the deadline you can submit your code on CodeJudge and see how well it scores. An entirely correct solution gets 100 points. Partially correct/efficient solutions will get a score between 0 and 100. On CodeJudge you can see the results of all the tests, but all except the already provided sample test data will be hidden.

The score you obtain in this mandatory assignment will count towards your final grade.

Competition Unfortunately, there is no competition for this mandatory assignment.