

CSC 312 : Databases

Practical Assignment 1 [50]

Due: Friday, 3rd September 2021 @ 10:00AM

This practical assignment aims to help you understand the nature of file systems, as used in decades of old. You've been provided with the file "SouthAfricaCrimeStats_v2.csv" on iKamva, along with this practical sheet.

The file details the history of crimes statistics from 2004 to 2015 per province and station per type of crime in South Africa.

Resource on Kaggle: <https://www.kaggle.com/slswessels/crime-statistics-for-south-africa>

Back in the day, companies stored their records in a file very similar to this one. In fact, even up to today, some companies still store their records in a file like this. In order to store information into, or retrieve information from, a pre-historic file like this, you need to know two things:

1. The structure of the file i.e. you need to know the ordering of the columns. So if you want to filter out records by period, you would need to know that the period is columns 4 to 11 in the file, and then use that info to do your filtering of records. This means that there is structural dependence in the file.
2. The data format of each field i.e. is it a number, string etc. So if you want to filter out records by period using columns 4 - 11, you also need to know exactly how the date period has been formatted. This means that there is data dependence in the file.

Ideally, however, you want some means of being able to store and retrieve information without caring how/where the data is stored in the file, or in what format the data is stored. What you want is a structurally independent and data independent data management system. This course will move towards this.

For now, this practical assignment will demonstrate to you, practically, exactly what structural and data dependence are, and why they are undesirable. Your job is to take the role of a "data processing specialist/manager" back in the day, whose job was to write software to get answers to questions in a file.

Your job is to write a well documented Python 3 program that uses the provided file and answers the following questions.

1. Display the contents of the CSV file in tabular format.
2. Report on the amount of crimes that occurred in a user specified province and period.
3. Report on the total amount of crimes for a user specified station for period full period of 2004 to 2015.
4. How many incidents involving truck hijacking were reported for the period 2010 - 2011?
5. How many “Arson” incidents occurred in Boitekong and Ngodwana in 2009 – 2010?
6. Which type of crime had the highest number of incidents in the 2014 – 2015 period?
7. For which period did Nongoma in KwaZulu Natal have the lowest amount of murder cases?
8. Which stations in the North West province has 0 cases for Attempted murder for the period 2008 – 2009?
9. Your program should present the user with a menu, which will allow s/he to choose the different “queries” they would be able to run. In other words, the user will be able to choose the “queries” in questions 1 – 8 above.
10. Create your own “query” other than the ones I have requested in questions 1 – 8, but do not make it too easy.

Format and Submission

Your Python file must be named: **your_student_number.py**

Please do not submit any type of project, only the Python file.

Also, again, please make sure to use only Python 3.

Please don't re-submit the SouthAfricaCrimeStats_v2.csv data file along with your submission, but please do make sure you provide all other files and resources that you think will be necessary for your source code to run successfully on our machines e.g. if you've got multiple Python files etc.

Also, make absolutely sure that you leave more than enough time for your file(s) to be uploaded, transmitted across the network etc. Aim to submit your work at least 12 hours before the deadline. Absolutely no late submissions will be accepted and absolutely no excuses involving networks in computer labs/residences etc. not working will be accepted.

Total Mark Allocation	[50]
• Each correct answer (1-8)	[16]
• Self-structured query	[4]
• Menu driven program	[5]
• Input and Output	[5]
• Documented Code	[20]