

## 2141 Database Systems – Final Project Part 2

### Section 0: Adjustments to the dataset

I decided that the dataset used for the last part of the project was insufficient and so I decided to change it to make the project flow smoother.

I now have 3 tables as follows:

#### **Student\_performance**

**Found on [Kaggle](#) and edited using ChatGPT to add the id, city, and extracurricular\_activity columns**

**~500 rows**

- student\_id INT (PK)
- gender VARCHAR(10)
- study\_hours\_per\_week INT
- attendance\_rate FLOAT
- past\_exam\_scores INT
- parental\_education\_level VARCHAR(50)
- internet\_access\_at\_home VARCHAR(3)
  - final\_exam\_score INT
  - pass\_fail VARCHAR(10)
  - city VARCHAR(50)
- extracurricular\_activity VARCHAR(50)

#### **locations - Generated via ChatGPT**

**~100 rows**

- city VARCHAR(50) (PK)
- country VARCHAR(50)
- continent VARCHAR(50)

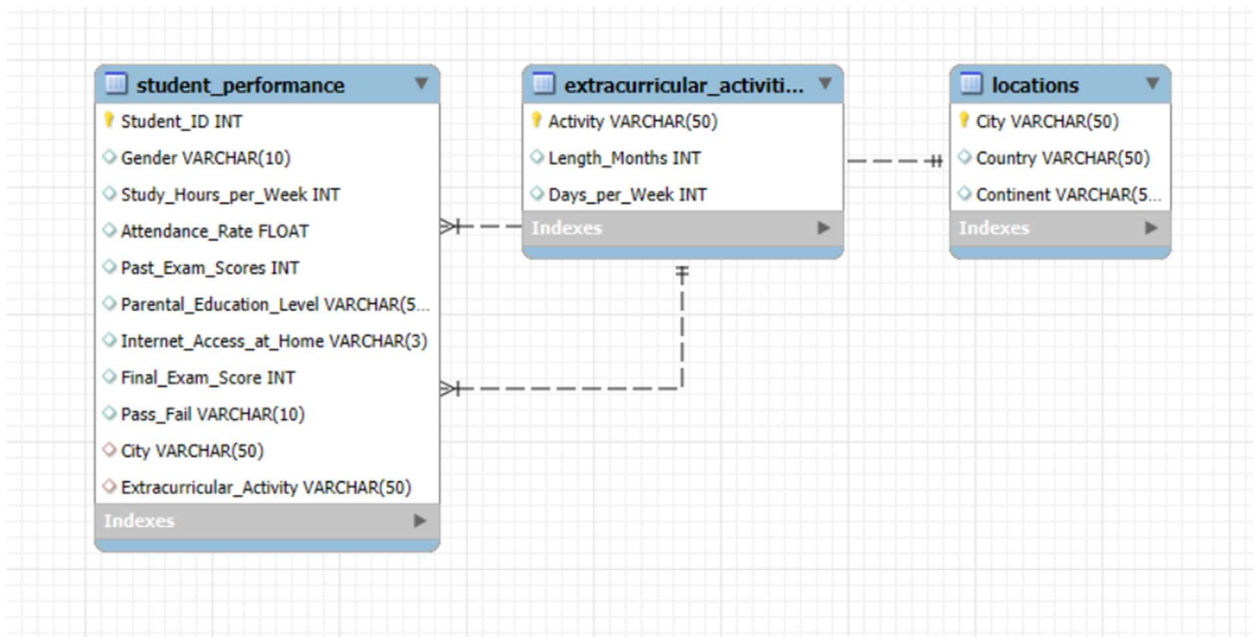
#### **extracurricular\_activities - Generated via ChatGPT**

**~10 rows**

- activity VARCHAR(50) (PK)
  - length\_months INT
  - days\_per\_week INT

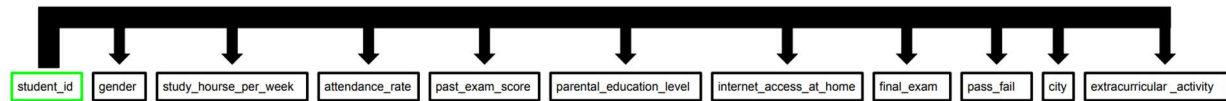
## Section 1: Designing The Database

### Internal Schema:

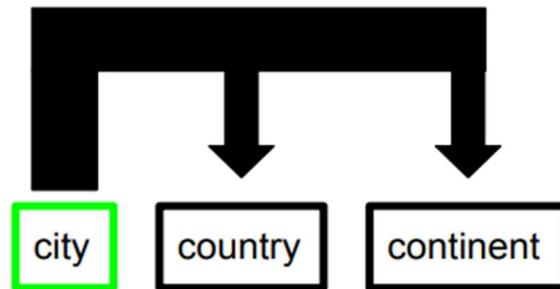


The 'main' table is Student performance it consists of 11 columns with the primary key being student\_id, a column that allows us to identify each row by a unique integer value. Student\_performance has two foreign keys, one to connect it to the extracurricular\_activities table and one to connect it to the locations table. The connection between student\_performance and extracurricular\_activities is formed by the foreign key activity, it uses the primary key in extracurricular\_activities to allow the main table to access the information about the activity the student is enrolled in. The connection between student\_performance and locations is made by the foreign key city that connects the city the student is located in, to the country and continent.

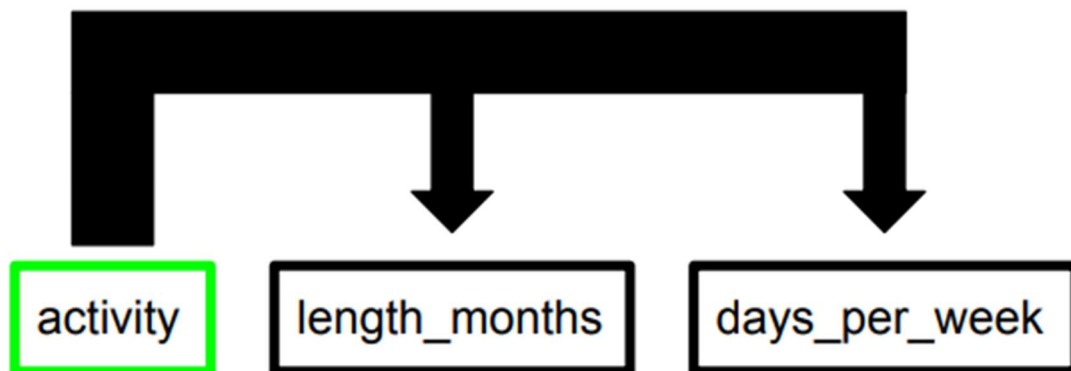
student\_performance Dependency Diagram:



location Dependency Diagram:



extracurricular\_activities Dependency Diagram:



Explanation of Dependency Diagrams:

The database is organized in such a way that no columns are dependant on each other, they are only dependant on that table's primary key. For example, the location table has three columns with the primary key being city, the other two columns are dependent on that key therefor each city has an associated country and city. To solve the issue of having a value in the main table change that could cause other values in the main table to change we find one common value and split the main table into two tables, this allows us to use the second table like an enum class in java. This avoids issues of data conjuncture and keeps the tables clean.