

1 Relational Databases

The most common model, the relational model sorts data into tables, also known as relations, each of which consists of columns and rows. Each column lists an attribute of the entity in question, such as price, zip code, or birth date. Together, the attributes in a relation are called a domain. A particular attribute or combination of attributes is chosen as a primary key that can be referred to in other tables, when it's called a foreign key. Each row, also called a tuple, includes data about a specific instance of the entity in question, such as a particular employee. The model also accounts for the types of relationships between those tables, including one-to-one, one-to-many, and many-to-many relationships. The basic of a relational database is a *table*.

1.1 First Subtask

Your task is to implement this table for a relational database. The *table* for the relational database must have the following properties:

1. It should have a PRIMARY KEY.
2. We can query data using multiple different types of keys.
3. We can see what tables it has relations with.

Complete `.set` and `query` methods in the provided boilerplate code.

1.2 Second Subtask

Create an Imaginary data which contains at least one set of datapoints which can be related to the data provided. Create another table and *relate* it with the first one. Complete the `relate` and method which would populate the `relations` members.

2 Instructions and Submission Guidelines

- You are provided with a boilerplate code and data file.
- You are required to complete the code in `relation_table.py` file.
- The file `test.py` is provided so that you can check your code. If you want, you can add more code to it. This file is for your personal use only and will not be evaluated.
- Your code will be tested on completeness and functionality and will not be tested on efficiency however use of efficient methods is a plus.
- If you want to setup your code in a notebook (*jupyter* or *colab*) just copy and paste the code from `utils.py` and `relation_table.py` into the notebook.
- Create 2 page (*max*) report on the steps that you followed while solving the problem.
- Submit a zip file containing the code and the report.
- The code can be in the form of a jupyter notebook (*google colab* or *.ipynb* file) or a python script (*.py* file).
- You are **NOT** allowed to use any external python module or package. Please write your code only using the standard python library. Note that any *proprietary package* used for modelling will **result in a significant reduction of your score**.
- Detection of Plagiarism will result in a **ZERO** being awarded to the student.
- Submit the file with the following naming scheme: `ROLLNO_MiniProject2`
- Do not include any code or screenshots in the report.
- Cite all references.