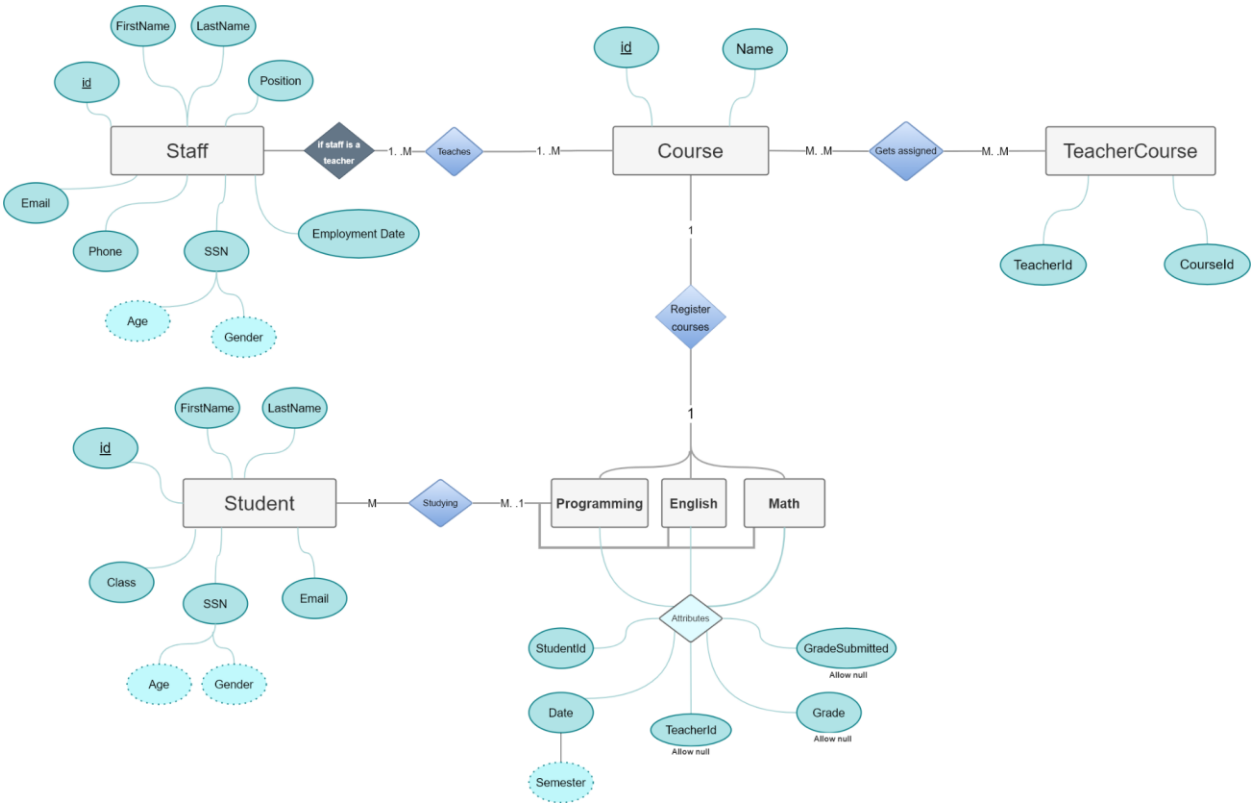
Nael Sharabi - SUT21

Database Model

When I built the entity-relationship model, I was aiming on getting the most scalable model possible for my fictional school, considering the need of having any kind of complex relationship between the entities. Thanks to the feedback I got on the old model I made, I realized that I hadn't applied normalization on the model, which is an essential thing in any ER-model. So after I made the necessary adjustments, I found out that I did have some redundancy in the old model and with that removed, it had brought some improvements on entities' relationships and data integrity. With all that done, the way was actually clear for implementing different functions on the entities.



SQL queries performance

- The queries can cover all the required functions and tasks in the project with only showing the necessary data. I used some data filtering to avoid processing undesired data records in tables. With every unnecessary process getting skipped, less system memory will be needed to inquire data, which leads to faster loads.

- All entity tables are bound by their primary and foreign keys. These unique keys help the system to instantly and accurately find the right data and join them together as one output.
- I tried to only use the suitable data types for the data attributes to avoid having too much allocated memory for the stored data size.
- All read(SELECT) commands request data records by the column name instead of using the asterisk (*) to avoid the process of converting the asterisk to the existing table names, thus the system will have fewer tasks to carry out. However, I did use the asterisk, but only to demonstrate and check a newly added record in a specific table.
- For most of the SQL commands that probably get executed frequently, I put them in stored procedures, since commands in stored procedures are always compiled and ready to run. I also implemented the necessary input parameters on the stored procedures to provide more flexibility in the queries and avoid script redundancy.