Database Technologies – Week 03

In this document, I will proceed with the complete design process of a database, from the ERD, and the Data Dictionary; all-the-way to the final build. Let’s get building!

1. The “Noun-Technique”

As it pertains to a body of text, specifically the description of a business model and its attributes, there are usually patterns that emerge.

For example, a description of McDonalds may read: “A company with 15 franchises in one city, holding a set number of staff members. Each staff member will have their own role, including kitchen staff, ordering staff, and cleaning staff. The ordering staff will oversee the payments that each customer makes.”

In this example, the patterns that are of value and use in creating databases are the *nouns*. To build a database, you must start with a foundation, even if that is just refining it down to key words that are of importance. The nouns in this example are: “company”, “franchise”, “city”, “staff”, “role” (kitchen, ordering, cleaning), “payment”, “customer”.

The Oxford Dictionary definition of noun is:

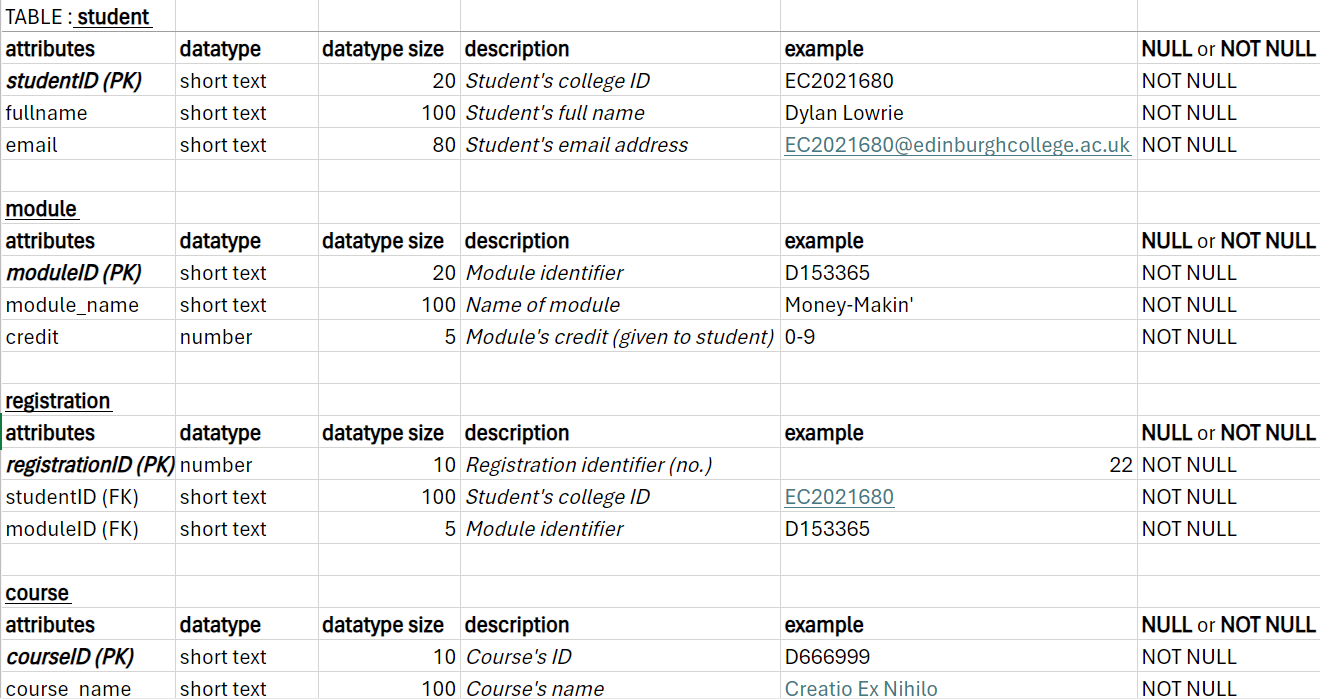
*a word (other than a pronoun) used to identify any of a class of people, places, or things (common noun), or to name a particular one of these (proper noun).*

This is significant as it is the nouns that refine each aspect of a business, or really anything at all, into categories to be made relational. This is the foundation of Relational Databases.

1. Data Dictionary

In the same way that a glossary is *an alphabetical list of words relating to a specific subject, text, or dialect, with explanations; a brief dictionary,* a “Data Dictionary” provides a textual/visual description of data attributes, their relationships, and characteristics. For example, after refining a business model into specific nouns, you may have key words such as “student”, “module”, or “registration”; these will become tables, and certain characteristics such as “studentID” and “full\_name” will be noted within said table. This is an essential building block for a database model. It keeps data consistent, accurate, and understandable, while organising it in a table of contents. This is usually done via spreadsheet apps such as Excel.

Here is an example of a Data Dictionary:



1. ERD

An ERD is the acronym for “Entity Relationship Diagram”. This is the process of turning our nouns into visuals. This is another building block of a database, and shows the relationships among people, objects, places, or events in a concise and graphical manner.

Various forms of information are shown in the data diagrams, also seen or pulled from within a Data Dictionary, such as:

* Table Names
* Attributes
* Name
* Data type (text, number, date)
* NULL or NOT NULL (Required or not required)

We also have our main keys; “Primary Keys (PK)”, “Foreign Keys (FK)”, and all other key words that make up our database.

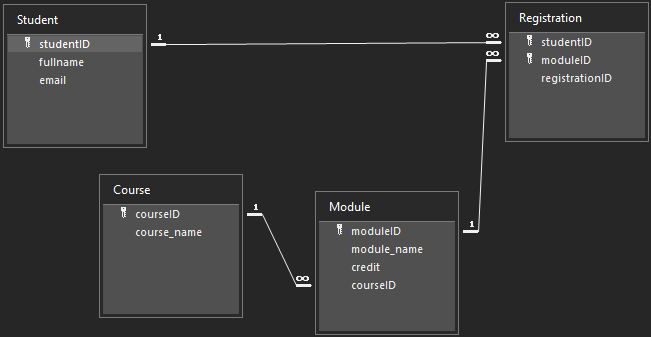
The Primary Key is like a title in that it signifies the theme of all other keys in its table. For example, in a table called “customer”, the PK may be “customer\_id” as that prefaces all other attributes like “customer\_name” and “customer\_payment\_method”.

FKs are simply keys which connect one table to another as they share the same value. For example, the table “payment” may have the row “payment\_method”, while “customer” will have this also; this is a Foreign Key.

These supposed similarities between a PK and an FK within separate tables is the essence of the *relational* diagram, their relationships. There are three types of relationships that can exist between two entities. An entity-relationship (ER) diagram can be created based on these three types, which are listed below:

* One-to-One (1:1) Relationship: Think of it like a unique pairing. Each entity A is directly related to only one entity B, and vice versa. There's a one-to-one correspondence between them.
* One-to-Many (1:N) Relationship: Picture it as a parent and children scenario. One entity A can have zero, one, or many instances of entity B associated with it, but each instance of entity B can be associated with only one instance of entity A.
* Many-to-Many (M:N) Relationship: This is like a many-to-many connection, where each entity A can be connected to zero, one, or many instances of entity B, and vice versa. It's a more complex relationship where both entities can have multiple connections to each other.

Here is an example of a final ERD:



1. PHP + SQL

Now that we have our foundational relational database, we must officially implement the information into its relevant database. We do this with PHP, using the SQL language.

There are a few steps one must take to finalise their database:

1. **Create a Database**: The first step is obvious. To have a database, it needs to be set up. Using a Database Management System like MySQL, PostgreSQL, or SQLite, you can easily set one up on the server.
2. **Insert Data (CRUD)**: Using the SQL language, you can turn the ERD into tangible tables, columns, and relationships; you can Create, Read, Update and Delete the data as you please.
3. **Connect**: PHP allows you to establish a connection between its code and your created database.

Here is what the ERD may look like on myPHP:

