# Introduction to SQL

* Structured Query Language
* It is not a programming language (by itself). It is a **declarative query** language
  + You tell the program what needs to be done instead of delivering the implementation details to do what needs to be done.
* It performs operations against a relational database

## Database

* It is just an organize collection of data stored in some organized format.
* They allow us to input, manage, organize, and retrieve data quickly
* Traditionally, it is organized into “tables” and each table will have a row and column
  + Rows will be the same as “records”
  + Columns will be the same as “fields”
* Data itself is the intersection between the row and column

## RDBMS

* It stands for Relational Database Management System
  + SQL is a specific style of RDBMS
* It upholds specified relationships between tables or our data
* It includes functions that maintain the security, accuracy, integrity, and consistency of the data

# SQL Sublanguages

## DDL

* Data Definition Language
* It is the sublanguage in SQL that is responsible for creating/altering the tables in your database
* Create – most used to create a table and their columns
  + Can also be used to create views, schema, etc.
* Alter – Will change the column of the table
  + Can also change certain properties or constraints
* Drop – will drop the table from the database out of existence

## DML

* Data Manipulation Language
* It is the sublanguage in SQL that is responsible for adding/changing/modifying the data within a table
* Insert – Adds row(s) to your table
* Select – gets data from your table
* Delete – remove data from your table
* Update – Modify pre-exist data in our table

# Constraints

* They are a way for you limit what data can be inputted in column
* It will specify one or more rules that the data must follow within a column

## Some commonly used constraints

* Type
  + Restrict what data type of value can be inputted in a column
* Unique
  + Data in this column cannot have repeating values
* Not null
  + Will make sure that every data you input cannot be null
* Primary Key
  + It is that every data inside of the column will be unique
  + It is implicitly unique and not null
  + Acts as the unique identifier for the records in the table
* Foreign Key
  + Data in this column will reference a data from another table/column
  + Establishes relationships between 2 columns in the same table or different tables
* Default
  + If you didn’t supply any value, default constraint will supply some default value
* Check
  + Adds an extra condition on the data
  + Ex: age column and it must have only above 18 age (age >= 18)
* Identity(1,1) – Good to know for you
  + Automatically fill in numerical values into that column
  + Essentially great way to have unique value in each row for your primary key
  + First parameter, it is the starting point the row
  + Second parameter, it is how much you want to increment by

# Multiplicity

* It is way to describe relationships between two tables
* We will be using primary and foreign key constraints to establish these relationships
* There are 3 main categories:

## One to One

* One row in Table A is directly relation to one row in Table B and it goes both ways (bi-direction)
* You have to make both columns unique constraint is needed
* Ex: One person can only have one heart

## One to Many

* One row in Table A is relation to many rows in Table B
* Make sure your foreign key is not unique
* Ex: One person has many fingers

## Many to Many

* Many rows in Table A are related to many rows in Table B
* You must use a join table to create this relationship
  + Join tables are just separate, independent table used to establish many to many
  + Consists of 2 columns are that both foreign keys
* Ex: Many students have many classes

# Normalization

* It is a design pattern that reduce/eliminate data redundancy and data duplication

## 0NF – Zero Normal Form

* No normalization is being utilized
* Data redundancy and data duplication

## 1NF – 1st Normal Form

* Each table must have a primary key
* All data must be atomic (One cell should only hold one value (NEVER A LIST))

## 2NF – 2ND Normal Form

* If you are in 2nd normal form, you are already in 1NF
* Remove partial dependencies
  + **Don’t create composite primary keys**
  + You need every column inside of that composite primary key to be dependent on the other columns of your table

## 3NF – 3rd Normal Form

* Remove transitive dependencies
* What are transitive dependencies? It is when a column in the table doesn’t depends on the primary key (they are unrelated and should be in two different tables)
* Make sure every column relates to the table you are putting in

## Referential Integrity

* It is a concept that is used to maintain our relationships between our table without missing information