What are Tables?

- An object comprised of two fields:
  - Heading: Table Name ; Column Names
  - Body: The corresponding column data
- Permanent storage in db
- A static member of the DB schema

What are Views?

- An object comprised of two fields:
  - Heading: Table Name ; Column Names
  - Body: The corresponding column data
- Exists virtually in memory
- Spun up reactively to user queries

How are they different?

Despite looking visually similar at first glance, i.e. they both represent groupings of data, composed together by some sort of relational meaning, the core differences lie in how they are individually created, stored, and their relationship to the database itself.

`Tables` represent the static relationships of data on a database. Due to their unchanging static nature (which is derived from being defined by the schema), `Tables` are therefore persisted on server storage. There also exists the concept of a `System Base Table` which stores and manages the database's metadata (data about data) but they also remain consistent with what has been described as far as storage and persistence are concerned.

Dissimilarly, `Views`, also known as `Virtual Tables`, represent a temporary lense for groups of data that could otherwise be stored in separate tables. `Views` represent a dynamic, reactionary grouping of data with meaning that is core to a specific query and therefore temporarily reside in virtual memory - the specific length depends on the specific queries being ran and on which databases.

Looking specifically at the provided challenge-sqlite-database; `blogpost`, `comment`, and `user` have been provided as tables for this database. There is no in-built table support to view all of the `id`s of users who have written a comment on a given blog post together with those comments, this would necessitate generating a temporary snapshot of the data, a view if you will. Generally speaking though - and this will depend some amount on what system is in use - a `select` statement with some return values is a view, even if the data being composed already relationally exists within the database as a table, i.e. `select * from blogpost` will return a view of the table.

Another core difference comes about as a result of these static and dynamic natures. Due to a `View` representing a single snapshot that only exists in virtual memory, it is treated as "read-only", i.e. unmodifiable. There are some systems in which a `Table` will be generated underneath the view that will persist changes however this is not generally the case. Direct access to tables will be needed if the modification of stored data is required.