

This is a small sample demonstrating what your project outline and database outline should look like. The Database Outline in Words, demonstrates using only the Character entity, how the attributes for an entity and the various constraints on them should be described. You are expected to describe all the attributes, and all constraints on all the entities.

Battlestar Galactica Database

Project Outline

I will be making a database representing the Battlestar Galactica universe. This is a universe from a fictional TV show where the humans are fleeing after their colonies were attacked by a cyborg race. This universe contains many interesting aspect, from people to spaceships. This complexity will make it a good candidate for a database project.

Database Outline, in Words

The entities in my database are:

- **Character** -- Character is the most important entity and participates in a relation with every other entity in this database. It has the follow attributes:
 - **id:** This number is automatically assigned to each character when they are recorded in our database. An auto-incrementing number which is the primary key.
 - **name:** Name of the character which is a string of maximum 100 characters. It cannot be blank and there is no default.
 - **race:** Race is a string of maximum 5 characters and it can only be either of the two values: Human or Cylon. It cannot be blank and the default race is Human.
 - **homeworld:** A homeworld is the planet to which a character belongs. It cannot be blank as a character will always have a homeworld. There is no default homeworld. This will contain the id of the Planet to which this character belongs. A character cannot belong to a Planet which does not exist in our database.
- **Ship** -- The different space ships in the series are stored in my database using the Ship entity.
- **Certification** -- The different ratings and skills that the characters in the show can have are recorded using the Certification entity.
- **Planet** -- The various planets from the show with their population and other geographic Information are represented in the Planet entity.

The relationships in my database are:

- **Characters are from planets** – A character can only be from one planet but a planet can be home to many characters. So, the Planet and Character entities are in a *one-to-many relationship*.
- **Characters serve on ships** – This is a *many-to-many relationship*, many Characters can serve on many Ships.
- **Characters have certifications** – This is a *many-to-many relationship* as Characters can have many Certifications and many Characters can be *certified* in the same thing.
- **Characters are subordinates of other characters** – Certain Characters serve under other Characters. This is a *one-to-many relationship* where a Character directly serves under a superior but a superior can have many subordinates.