|  |
| --- |
| BUILDING THE DATABASE |

What we need next is a database for Cobalt to look at and generate CRUD modules from. Looking at our RPG Character Management System’s example, we would need several pieces of information: the **Character** himself, his **Character Class,** his **Race,** and any **Skills** that he may have. We can translate these into individual tables and have them prepared for Cobalt to understand.

Let us define the tables first before assembling our database:

|  |  |
| --- | --- |
| **character** | |
| character\_id | int(11) - PK |
| character\_name | varchar(255) |
| character\_class\_id | int(11) |
| race\_id | int(11) |
| gender | varchar(255) |
| character\_portrait | varchar(255) |
| strength | varchar(255) |
| dexterity | varchar(255) |
| intelligence | varchar(255) |
| date\_created | varchar(255) |

|  |  |
| --- | --- |
| **character\_class** | |
| character\_class\_id | int(11) - PK |
| character\_class\_name | varchar(255) |
| character\_class\_type | varchar(255) |

|  |  |
| --- | --- |
| **race** | |
| race\_id | int(11) - PK |
| race\_name | varchar(255) |

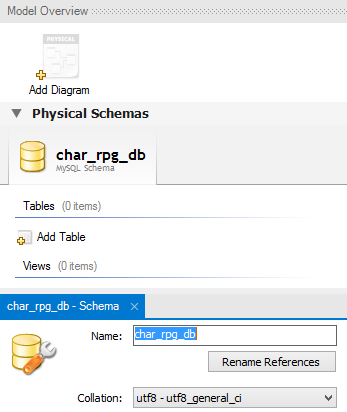
|  |  |
| --- | --- |
| **skill** | |
| skill\_id | int(11) - PK |
| skill\_name | varchar(255) |
| description | varchar(255) |

|  |  |
| --- | --- |
| **character\_skill** | |
| character\_skill\_id | int(11) |
| character\_id | int(11) |
| skill\_id | int(11) |

There are two approaches to creating the database: you may either create the database from scratch using PhpMyAdmin or you may utilize MySQL Workbench to design the database schema in a GUI-like manner.

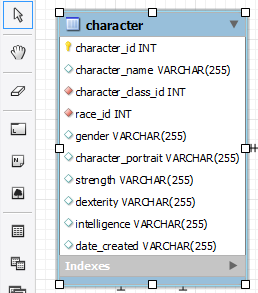
**BUILDING VIA MYSQL WORKBENCH**

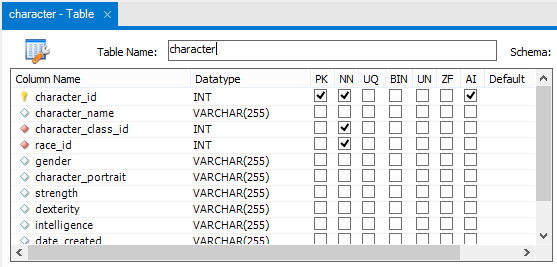
Creating the database via MySQL Workbench is slightly more convenient than working on PhpMyAdmin directly as you can visually define the table structure in the same manner as creating an ERD. Simply create a new Model and name it “char\_rpg\_db”



Hit “Add Diagram” to start building your database model. You will be taken to a blank canvas where you can define your tables and fields, then connect them via relationships.

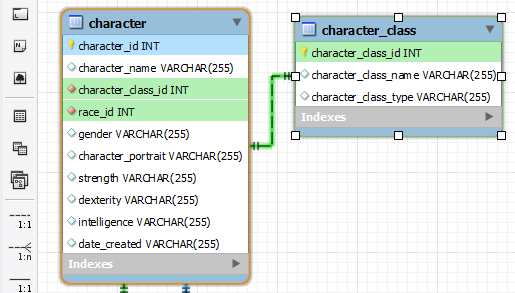
To create a new table, simply select the “Place a new table” button and click on the canvas to place your new table. You will be prompted to enter details of your new table.



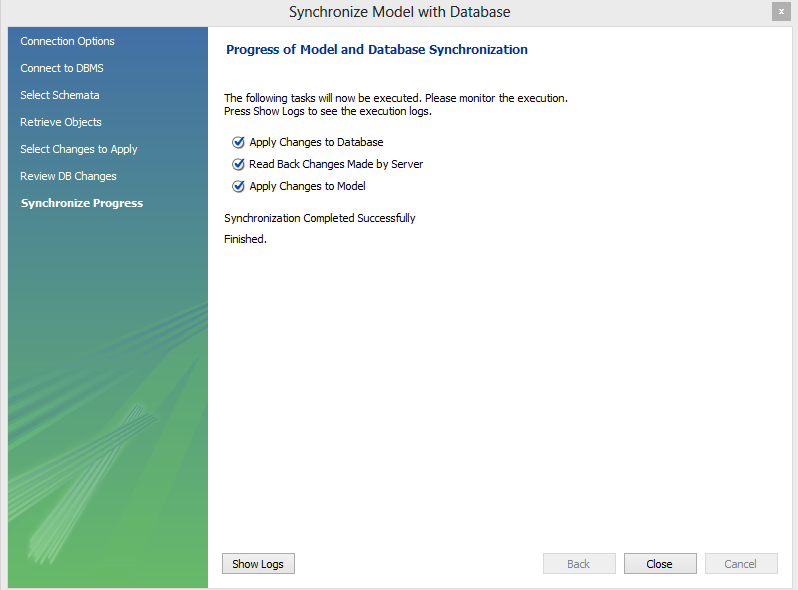


Noteworthy columns are the **PK, NN,** and **AI** columns. These stand for *Primary Key*, *Not Null*, and *Auto Increment* columns respectively. Tick the checkboxes for each database column where it applies. Make sure all your Primary Keys have all three checkboxes ticked!

To connect database tables via relationships, select the appropriate relationship type you need (either a “1:1 Non-Identifying relationship” for 1-to-1 relationships or a “1:n Non-Identifying relationship” for 1-to-Many relationships). Click the “child” table first (i.e the table that’s asking for data) followed by the “parent” table (the table giving the data) and a foreign key link will immediately be established between the two tables.



Once you are finished building all the database tables into the Model schema, it is now time to “synchronize” them into our MySQL database server. Create the database first in PhpMyAdmin then return to the Workbench. From the menu, select “Database” > “Synchronize Model”. Configure the connection parameters to your database server. Go through the wizard and execute the synchronization of your model to your database.



Once you are done the database is ready for the Cobalt code generator to process.