



# EXERCISES — World of Yakraft - 1

---

version #



# Copyright

This document is for internal use at EPITA ([website](#)) only.

Copyright © 2021-2022 Assistants [<assistants@tickets.assistants.epita.fr>](mailto:assistants@tickets.assistants.epita.fr)

**The use of this document must abide by the following rules:**

- ▷ You downloaded it from the assistants' intranet.\*
- ▷ This document is strictly personal and must **not** be passed onto some-one else.
- ▷ Non-compliance with these rules can lead to severe sanctions.

# Contents

1	World of Yakraft - 1	3
1.1	Goal . . . . .	3
1.2	Presentation . . . . .	4
1.3	The SQL Rationale . . . . .	4
1.4	Schema . . . . .	5
1.5	Exercises . . . . .	6
1.5.1	A few statistics . . . . .	6
1.5.2	Level-up! . . . . .	7
1.5.3	Missing creatures . . . . .	7
1.5.4	Advanced statistics . . . . .	7

---

\*<https://intra.assistants.epita.fr>

# 1 World of Yakraft - 1

## Files to submit:

- world\_of\_yakraft\_1/req01.sql
- world\_of\_yakraft\_1/req02.sql
- world\_of\_yakraft\_1/req03.sql
- world\_of\_yakraft\_1/req04.sql
- world\_of\_yakraft\_1/req05.sql
- world\_of\_yakraft\_1/req06.sql
- world\_of\_yakraft\_1/req07.sql
- world\_of\_yakraft\_1/req08\_creature.sql
- world\_of\_yakraft\_1/req08\_creature\_template.sql
- world\_of\_yakraft\_1/req09.sql
- world\_of\_yakraft\_1/req10.sql
- world\_of\_yakraft\_1/req11.sql
- world\_of\_yakraft\_1/req12.sql
- world\_of\_yakraft\_1/req13.sql
- world\_of\_yakraft\_1/req14.sql
- world\_of\_yakraft\_1/req15.sql

## Provided files:

- world\_of\_yakraft\_1/schema.sql
- world\_of\_yakraft\_1/data.sql

## 1.1 Goal

SQL is a language which allows you to do [Relational Algebra](#) with a very simple (yet expressive) syntax. It is useful in a lot of domains, and it would be a shame to overlook it because of some mistaken prejudices.

## 1.2 Presentation

**World of Yakraft** is a Massive Multi-player Online Role Playing Game you are about to create. The player can create a custom character, fight creatures, complete quests and level up.

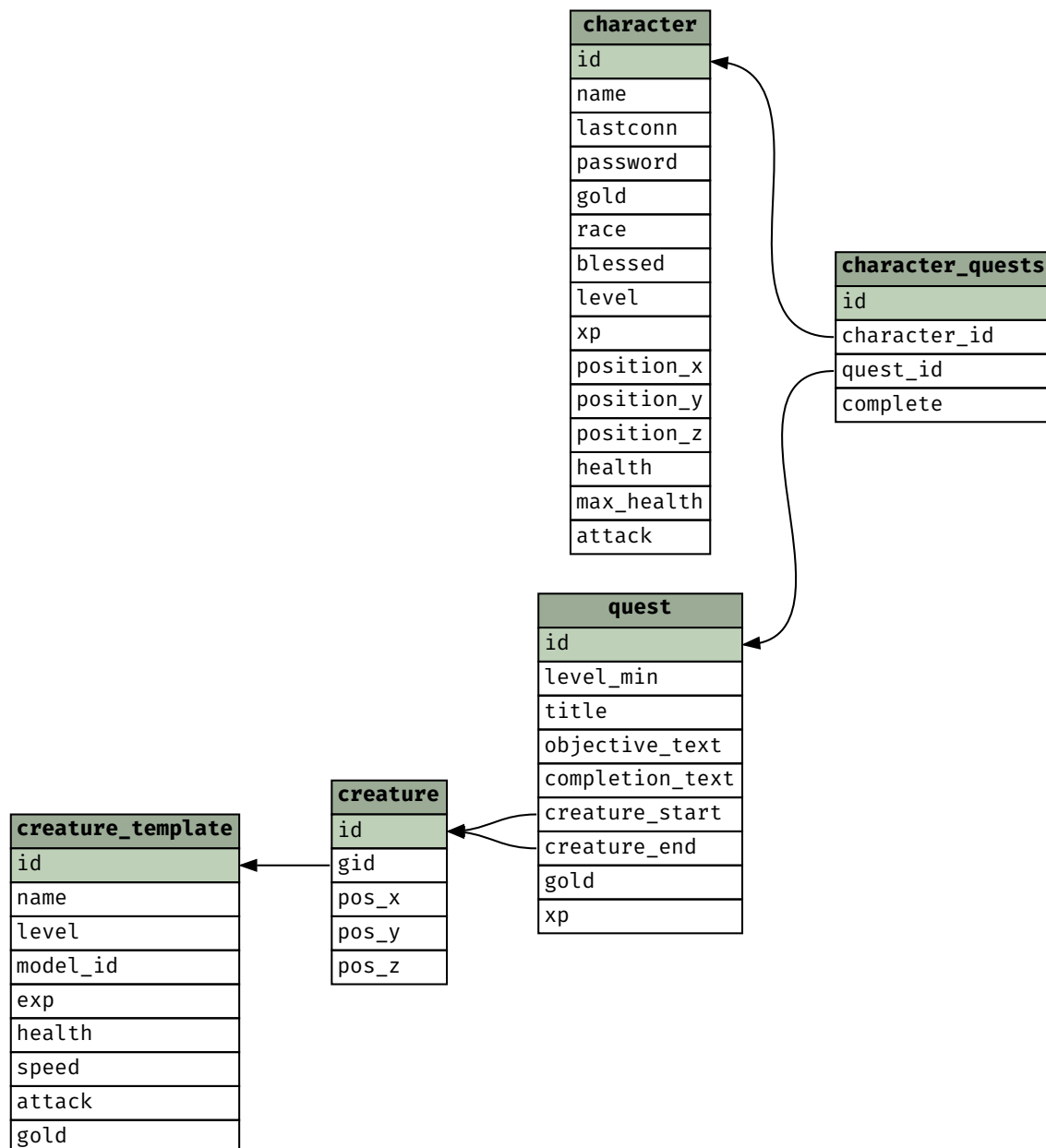
## 1.3 The SQL Rationale

It would be really difficult to try to create the server of this game without using a database. We cannot just store the data in files, because:

- The data is dynamic. Some elements like the quests you completed or the items you have change all the time, and storing this information in files would mean having a lot of Input/Output on the disk.
- You want to allow concurrent access to the database, because your server will probably be multi-threaded. It is a lot easier to do concurrent access on databases that were specifically designed for this than on files.
- It would be a pain from a developer perspective to express relational operations without using relational algebra, and you would have to write tons of useless boilerplate to circumvent this.

Now that you are hopefully convinced of the need of a good database for this problem, let us dive into details of how the game will be organized.

## 1.4 Schema



You can find on the intranet a schema of the database we will be using for this subject. It contains 5 tables for now:

- creature\_template represents an abstract creature and its characteristics. There can be multiple “instances” of a creature, at different positions in the game
- creature is the list of instances of these creatures, their position and their current state
- quest contains the list of quests

- `character` is the list of characters, with their status and characteristics
- `character_quests` is the list of quests the characters interacted with

Comments in the schema should be explicit enough to understand the purpose of every field. If not, do not hesitate to ask an assistant!

### Going further...

As an anecdote, this database is adapted from [the MaNGOS project](#) database in a highly simplified form. This project aims to be an MMO server compatible with World of Warcraft. Yes, people do actually use SQL on their gaming servers.

## 1.5 Exercises

Below is the list of exercises you must submit. If the type of ordering is not specified, then it must be ascending.

### 1.5.1 A few statistics

Your server has been running for a few days, let us do some stats!

- In `req01.sql`, write a query that returns the list of the names of all the characters of the server ordered by their names.
- In `req02.sql`, write a query that returns the names of all the dead characters of the server (which are the characters whose health is equal to zero) ordered by their names.
- In `req03.sql`, write a query that returns the names of the top 5 characters, ordered by experience in descending order.
- In `req04.sql`, write a query that returns the title of all the quests, sorted by the length of their titles.
- In `req05.sql`, write a query that returns the names of the characters ordered by how advanced in the game they are, in descending order - from the best character to the newest. To decide between characters with the same level, sort by experience. To decide between characters with the same experience, sort by the amount of gold they own.
- In `req06.sql`, write a query that returns the titles of the quests that are “worth it” for your level 10 character, ordered by their titles. You realized that it is worth it to do a quest when the creature that starts the quest is the same as the one that ends it (because you usually do not have to travel a lot) *or* the gold reward is more than 100. In any case, the level requirement must be at least 8, but not more than 10 (because you can not do the ones with a higher requirement).

### 1.5.2 Level-up!

Write the answers to this exercise in `req07.sql`.

- The player `Tilon` won a contest during a special event. Write a request that gives him the level 15.
- The player `Kuro` complained that his level was not updated despite the fact that he gained enough experience. Write a request that increments his level by one.
- You decided to change the way the `max_health` stat is computed for characters. It is now  $(level + 1) * 10$  if the character is blessed and  $level * 10$  otherwise. Write a request that updates this stat for all the characters.

### 1.5.3 Missing creatures

To please high level players, you decided to add some strong creatures in `req08_creature_template.sql`. Their name is 'Wild YAKA', they are level 99, their 3D model ID is 1387, they give 1000 experience points and 1000 gold when killed, their health is 8, their speed is 8 and their attack is 50.

Add three instances of the Young wolf to spawn at positions in `req08_creature.sql`:

- (5, 6, 7)
- (-3, -2, -1)
- (42, 43, 44)

### 1.5.4 Advanced statistics

- In `req09.sql`, write a query that returns the name of the creature whose ID is 6 (the creature ID, not the template).
- In `req10.sql`, write a query that returns the id of all the creatures (the instances, not the templates) whose level is strictly superior to 10 ordered by id.
- In `req11.sql`, write a query that returns for each quest its title and the name of the creature that starts it. Sort it by quest's title and creature's name.
- In `req12.sql`, write a query that returns the name of the creatures that are the same at the start *and* the end of a quest, ordered by their names.
- In `req13.sql`, write a query that returns the names of the creatures that never appear in the game (the template exists but is never "instantiated"), ordered by their names.
- In `req14.sql`, write a query that returns the title of the quests that the character `Kuro` completed, ordered by their titles.
- In `req15.sql`, write a query that returns for each quest its title and the name of the character that started it without completing it, ordered by quests' titles then characters' names.

*It is my job to make sure you do yours.*