# Exercises: Introduction to SQL

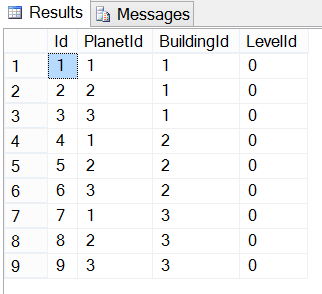
This document defines the class assignments from the ["Databases" Course @ Software University](https://softuni.bg/trainings/1168/Databases-Jun-2015).

## Create initial architecture for massive multiplayer role-play game

The interview has gone well and you are now in charge for the back-end architecture of a massive multiplayer role-play game. Your first task is to set the initial database structure using the knowledge so far. The game has players which consist of username, email, password and points. Each of them has several planets. A planet is placed in a 3D-Coordinate system (x, y, z) and have resources from predefined types (crystal and metal). There are some buildings in-game. A metal mine, mineral mine and a fusion reactor. Each building can have up to 10 levels that cost some of a predefined resource (crystal and metal). The first level of every building cost 1000 metal and 500 crystal. The metal mine multiplier per level is 1.2, the crystal mine is 1.5 and the fusion reactor’s – 2.0. Each player’s planet has initially all buildings on level 0.

### Task 1. Populate player’s buildings with initial level 0

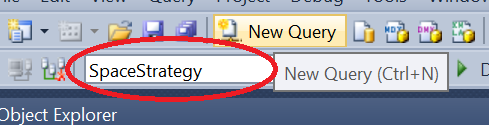
* Consider using Cartesian product in order to populate the relational table with only one INSERT query. The final result after populating should be something like the screenshot below



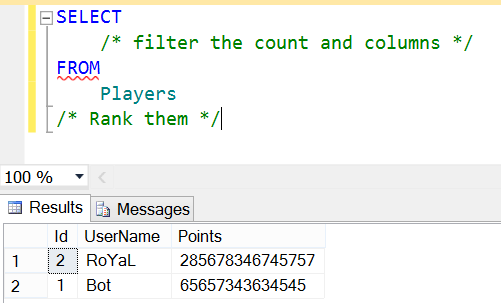
## Show game information

### Task 1. Show player ranking (first 100 players) without showing any sensitive information

* Query the players table

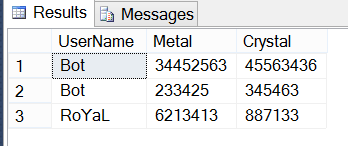


* Make a projection



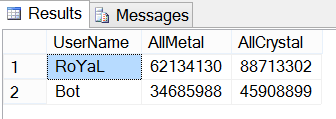
### Task 2. Show planets resource information for first three players

* Consider using **JOINS and/or NESTED SELECTS**



### Task 3. Rank players by all their resources and show them

* Consider using:
  + Aggregate functions
  + Aliases
  + Group condition



### Task 4. Extract information for each player’s building levels

* Consider using **multiple joins** and aliases
* See the example resultset below

