Minecraft Database

Ein Python und SQLite Projekt von Elias Cuin und Florian Strobl

Zeitraum: Sept. & Dez. 2022

Lehrer: Herr Raake

Inhaltsverzeichnis

[Beschreibung des Projektes 2](#_Toc123393441)

[ER-Modell 3](#_Toc123393442)

[Relationen-Modell 4](#_Toc123393443)

[Integritätsbedingungen 5](#_Toc123393444)

[Beispiel Testdaten 6](#_Toc123393445)

[SQLite CREATE Befehle für alle Tables 7](#_Toc123393446)

[SQLite INSERT INTO Befehle für alle Tables mit Testdaten 8](#_Toc123393447)

[Codestruktur mit allen Funktionssignaturen 9](#_Toc123393448)

[Erklärung zu der Anzeige und der Einpflege Funktion 10](#_Toc123393449)

[SQLite Aufgaben mit Lösungen 11](#_Toc123393450)

[Link zu einem Repl und Anhang mit Sourcecode 12](#_Toc123393451)

# Beschreibung des Projektes

Ein Minecraftserver besitzt eine Welt, in der viele verschiedene Spieler miteinander spielen können. Sie verändern die Blöcke in der Welt und interagieren mit den vielen Tieren im Spiel. Jeder Spieler kann sich genau aussuchen auf welchen Server, und damit verbunden in welcher Welt er spielen möchte.

(Entity, Relation, Attribut)

"Serverworlds" haben eine ID ("server id"), um sich mit ihnen verbinden zu können und einen Namen ("name") und Logo ("icon"), damit auch wir Menschen sie unterscheiden können.

In diesen "Serverworlds" spielen ("plays") verschiedene Spieler ("Player"). Darüber hinaus sind diese bevölkert ("populated by") durch verschiedene Entities ("m\_Entities"), also Dinge wie Tiere, und ist ("has") voller "Chunks", die selbst wiederum aus Blöcken ("Blocks") bestehen ("build of"). Diese "Chunks" speichern immer deren absolute Position ("chunk position"), um später zu wissen, wo sie sich in der Welt ("Serverworld") befinden.

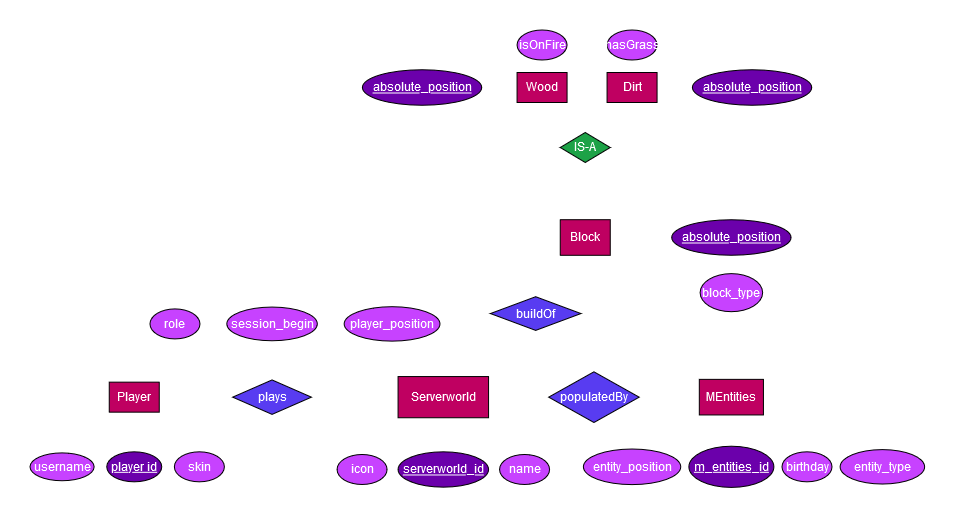
Jeder "Player" hat einen Namen ("username"), ein Avatar ("skin"), sowie eine eindeutige ID ("player id").

Das Spielen ("plays") in einer "Serverworld" muss die aktuelle absolute Position ("player position") des Spielers ("Player") speichern, wobei noch der Rang ("role") und die Uhrzeit des Spielbeginnes ("session begin") erfasst werden.

Die Enitites ("m\_Entities") der Welten haben jeweils eine eindeutige ID ("entity id"), ein Typ ("entity type"), wie beispielsweise "Kuh", eine absolute Position ("entity position") und zuletzt dessen Erstellungszeitpunkt ("birthday") gespeichert.

Die Blöcke ("Blocks") in den "Chunks" haben ihre relative Position ("relative position") gespeichert und darüber hinaus noch aus was sie bestehen ("block type"). Blöcke ("Blocks") können aus Holz ("Wood") bestehen, welche brennen können ("isOnFire") oder sie sind aus Erde ("Dirt"), welche wiederum Gras auf ihrer Oberfläche wachsen lassen können ("hasGras").

# ER-Modell



# Relationen-Modell

Serverworld(serverworld\_id, name, icon)

Player(player\_id, username, skin)

MEntities(m\_entities\_id, entity\_position, birthday, entity\_type)

Block(absolute\_position, block\_type)

Wood(^absolute\_position, isOnFire)

absolute\_position verweist auf den Primärschlüssel von Block

Dirt(^absolute\_position, hasGrass)

absolute\_position verweist auf den Primärschlüssel von Block

plays(^player\_id, ^serverworld\_id, session\_begin, player\_position, role)

player\_id verweist auf den Primärschlüssel von Player

serverworld\_id verweist auf den Primärschlüssel von Serverworld

populatedBy(^m\_entities\_id, ^serverworld\_id)

m\_entities\_id verweist auf den Primärschlüssel von MEntities

serverworld\_id verweist auf den Primärschlüssel von Serverworld

buildOf(^absolute\_position, ^serverworld\_id)

absolute\_position verweist auf den Primärschlüssel von Block

serverworld\_id verweist auf den Primärschlüssel von Serverworld

# Integritätsbedingungen

Serverworld

serverworld\_id: bigint

name: string

icon: string | null

Player

player\_id: bigint

username: string

skin: string

MEntities

m\_entities\_id: bigint

entity\_position: string

birthday: integer

entity\_type: integer

CONSTRAINT:

0 <= entity\_type < 10

Block

absolute\_position: string

block\_type: integer

Wood

^absolute\_position: integer

isOnFire: boolean = 0

Dirt

^absolute\_position: string

hasGrass: boolean = 0

plays

^player\_id

^serverworld\_id

session\_begin

player\_position

role

populatedBy

^m\_entities\_id

^serverworld\_id

buildOf

^absolute\_position

^serverworld\_id

# SQLite CREATE Befehle für alle Tables

# Beispiel Testdaten

# SQLite INSERT INTO Befehle für alle Tables mit Testdaten

# Codestruktur mit allen Funktionssignaturen

# Erklärung zu der Anzeige und der Einpflege Funktion

# SQLite Aufgaben mit Lösungen

# Link zu einem Repl und Anhang mit Sourcecode