

Universidad Nacional Autónoma de México
Escuela Nacional de Estudios Superiores Unidad Morelia

Documentación

Minecraft Structures Database

PRESENTA:

Roxana Pérez Medina

Luis Alberto García Orozco

Antonio Suazo Hurtado

Alexis Uriel Aguilar Uribe

PROFESOR:

LIC. Gutavo Cano Salazar

ASIGNATURA: Bases de Datos

A: 7 de Junio del 2024

ÍNDICE

1	Documento de Requerimientos	3
2	Diagrama Entidad Relación	5
3	Modelo Lógico Relacional	6
4	Diccionario de Datos	7
5	Ejemplos de Consultas	11
5.1	Ejemplo 1	11
5.2	Ejemplo 2	12
5.3	Ejemplo 3	13
5.4	Ejemplo 4	14
5.5	Ejemplo 5	15
5.6	Ejemplo 6	16
5.7	Ejemplo 7	17
5.8	Ejemplo 8	18

1 Documento de Requerimientos

El famoso youtuber y streamer Vegetta777 requiere el uso e implementación de una base de datos relacional en la que pueda reconocer y explorar de forma detallada las diferentes estructuras con las que cuenta Minecraft hasta la última versión estable, la 1.20.2, con el fin de crear un *mod* (modificación al juego base) para Minecraft donde incluya nuevas ***structures*** y ***mobs*** creadas por él mismo, para esto necesita saber cuáles son los ***blocks***, ***mobs*** e ***items*** que se encuentran más seguido en las estructuras de cada ***biome*** (para evitar usar estos en sus nuevas estructuras), además de cuáles son los ***biomes*** y ***dimensions*** que cuentan con menos ***structures***, para agregar más ***structures*** a los mismos, y hacer más inmersiva la experiencia del juego.

Minecraft se conforma de tres ***dimensions*** (dimensiones), cada una cuenta con sus propios ***biomes*** (biomas), ambas tienen características únicas, y para ser identificadas se hace uso de un ***identifier*** (identificador) y de un ***numeric id*** (identificador numérico), además de contar con un ***name*** (nombre) que las describe. Los ***biomes*** además cuentan con otras características que los distinguen como lo son su ***temperature*** (temperatura), ***precipitation*** (precipitación), ***grass color*** (color del césped), ***foliage color*** (color de las hojas) y ***water color*** (color del agua).

Las diferentes ***structures*** (estructuras) con las que cuenta Minecraft se encuentran en ciertos ***biomes*** (biomas), y algunas son únicas a ciertos ***biomes***. Para identificar a cada ***structure***, de manera interna Minecraft le asigna un ***identifier*** que es único y, para que los jugadores puedan referirse a ellas, tienen un ***name*** (nombre) que representa cómo es la ***structure***.

Todo lo que conforma a Minecraft son los ***blocks*** (bloques), y las ***structures*** no son la excepción, y esto es lo más destacable debido a que la composición de diferentes ***blocks*** permiten distinguir si estamos dentro o afuera de una ***structure***. Los diferentes ***blocks*** que pueden formar a una ***structure*** tienen tanto un ***name*** como una ***identifier***, además de tener una distinción que los hace diferentes de otros ***items*** en Minecraft, y son las propiedades de ***blast resistance*** (resistencia a las explosiones), ***hardness*** (dureza al romperlo), ***luminosity*** (cantidad de luz que emiten) y ***transparency*** (opacidad para ver otros blocks a través de él).

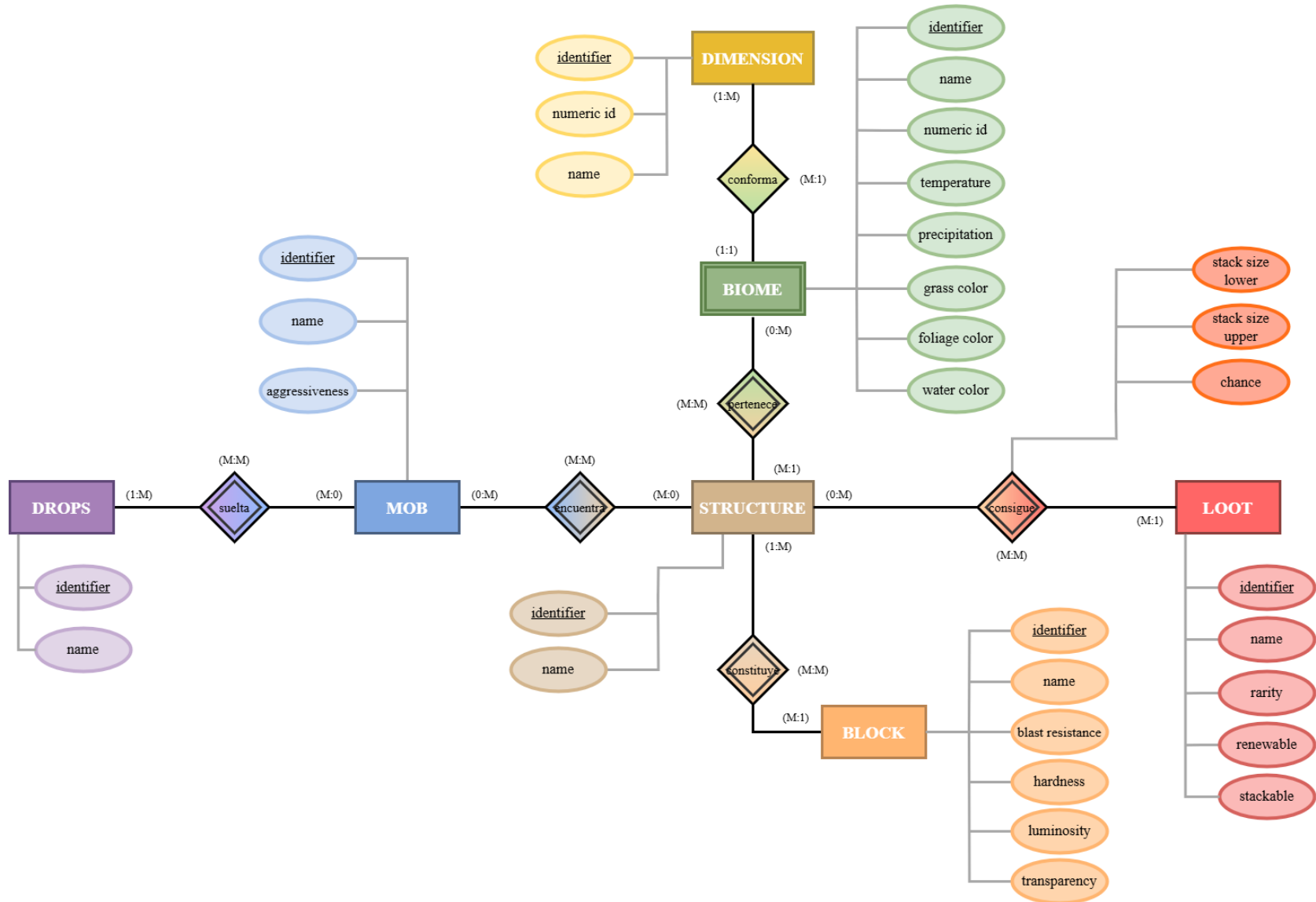
Lo atractivo de las **structures** en Minecraft, aparte de los bloques decorativos y de la emoción de exploración, son los **items** (objetos) que se pueden encontrar a los distintos cofres que están en la **structure**. Estos **items** son referidos como **loot**, algunos aparecen exclusivamente en ciertas **structures** pero existen **items** que se encuentran repetidos en distintas **structures**. La única distinción, en este caso, es que tienen diferente probabilidad de aparecer en los cofres (lo que es referido como **chance** de aparición) y el **stack size** (cantidad de **items** que se encuentran) que está dado por un rango numérico.

Para identificar a los diferentes **items** que se consiguen como **loot**, Minecraft le asigna un **identifier** y un **name** que les sirve a los jugadores para referirse a ellos. Los **items** pueden clasificarse según su **rarity** (rareza), si son **renewable** (que se pueden conseguir de manera natural en el juego), y si son **stackable** (se pueden apilar) Además se debe considerar que una **structure** puede o no tener un cofre con **items**.

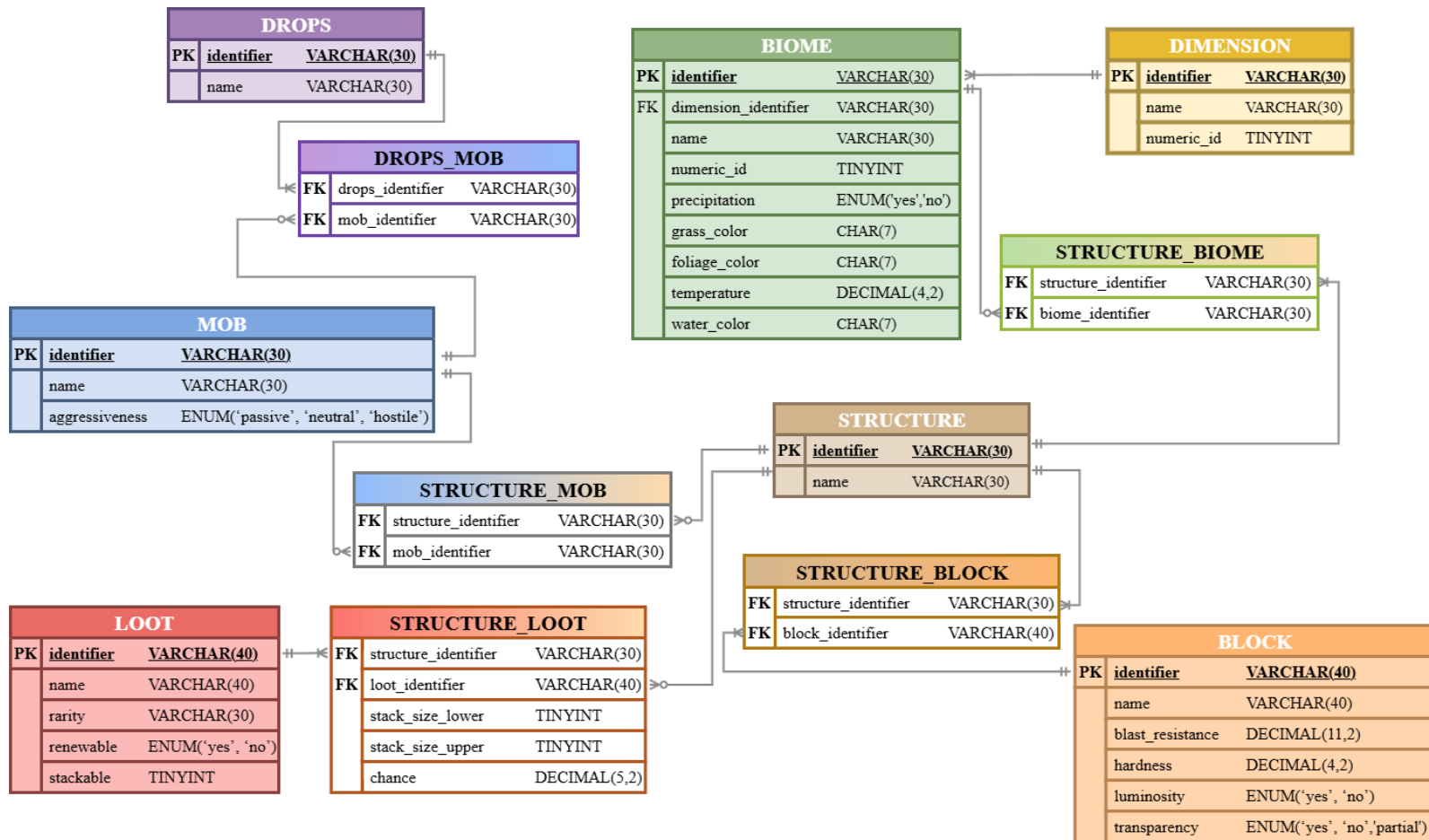
Otra forma de conseguir **items** de las estructuras es por medio de matar a los diferentes **mobs** (entidades) que se pueden encontrar al momento de explorar a las **structures**, pero a estos **items** que se obtienen de los **mobs** se les llama **drops**. No todos los **mobs** nos van a atacar, a nosotros el jugador, para medir esta agresividad se hace uso de un valor de **aggressiveness**. Los diferentes **mobs** tienen un **name** que los relaciona con su diseño y concepto, a su vez Minecraft hace uso de un **identifier** para reconocer a cada **mob** de forma interna. De todos los **mobs** con los que cuenta Minecraft, hay algunos que son exclusivos de ciertas **structures** y a su vez, existen mobs que no se generan en determinadas **structures**.

De los **drops** que se obtienen de los **mobs** ocurre algo similar, es decir, algunos **mobs** no sueltan **drops** pero otros son únicos a ciertos **mobs**, y para saber qué **items** nos da un **mob** al momento de matarlo se usa un **name**, y además de un **identifier** propio de Minecraft que permite distinguir internamente entre los diferentes **drops**.

2 Diagrama Entidad Relación



3 Modelo Lógico Relacional



4 Diccionario de Datos

<u>DIMENSION</u>			
NOMBRE CAMPO	TIPO DE DATO	RESTRICCIÓN	DESCRIPCIÓN
identifier	VARCHAR(30)	PK	Nombre interno de una DIMENSION
name	VARCHAR(30)	NOT NULL	Nombre de una DIMENSION
numeric_id	TINYINT	NOT NULL	Identificador numérico de una DIMENSION
OBSERVACIONES DE LA TABLA: Tabla fuerte que describe las características de una DIMENSION. Está relacionada con la tabla: biome.			
LLAVES FORÁNEAS:			
EJEMPLO DE REGISTRO: ('overworld', 'overworld', 0)			

<u>BIOME</u>			
NOMBRE CAMPO	TIPO DE DATO	RESTRICCIÓN	DESCRIPCIÓN
identifier	VARCHAR(30)	PK	Nombre interno de un BIOME
dimension_identifier	VARCHAR(30)	FK / NOT NULL	Nombre interno de la DIMENSION a la que pertenece un BIOME
name	VARCHAR(30)	NOT NULL	Nombre de un BIOME
numeric_id	TINYINT	UNSIGNED UNIQUE	Identificador numérico de un BIOME
precipitation	ENUM('yes', 'no')	NOT NULL	Indica la presencia de precipitación en un BIOME
grass_color	CHAR(7)	NOT NULL	Valor hexadecimal del color del pasto en un BIOME
foliage_color	CHAR(7)	NOT NULL	Valor hexadecimal del color de las hojas en un BIOME
temperature	DECIMAL(4,2)	NOT NULL	Indica la temperatura de un BIOME
water_color	CHAR(7)	NOT NULL	Valor hexadecimal del color del agua en un BIOME
OBSERVACIONES DE LA TABLA: Tabla débil que describe las características de un BIOME. Está relacionada con las tablas: dimension y structure_biome.			
LLAVES FORÁNEAS: dimension_identifier CON REFERENCIA A TABLA dimension (identifier)			
EJEMPLO DE REGISTRO: ('ocean', 'overworld', 'ocean', 44, 'yes', '#8eb971', '#71a74d', 0.5, '#3f76e4')			

<u>STRUCTURE</u>			
NOMBRE CAMPO	TIPO DE DATO	RESTRICCIÓN	DESCRIPCIÓN
identifier	VARCHAR(30)	PK	Nombre interno de una STRUCTURE
name	VARCHAR(30)	UNIQUE	Nombre de una STRUCTURE
OBSERVACIONES DE LA TABLA: Tabla fuerte que describe las características de una STRUCTURE. Está relacionada con las tablas: structure_biome, structure_block, structure_mob y structure_loot.			
LLAVES FORÁNEAS:			
EJEMPLO DE REGISTRO: ('ancient_city', 'ancient city')			

<u>STRUCTURE_BIOME</u>			
NOMBRE CAMPO	TIPO DE DATO	RESTRICCIÓN	DESCRIPCIÓN
structure_identifier	VARCHAR(30)	FK / NOT NULL	Nombre interno de una STRUCTURE
biome_identifier	VARCHAR(30)	FK / NOT NULL	Nombre interno de un BIOME
OBSERVACIONES DE LA TABLA: Tabla transitiva que une las tablas STRUCTURE y BIOME. Está relacionada con las tablas: structure y biome.			
LLAVES FORÁNEAS: structure_identifier CON REFERENCIA A LA TABLA structure (identifier) biome_identifier CON REFERENCIA A LA TABLA biome (identifier)			
EJEMPLO DE REGISTRO: ('ancient_city', 'deep_dark')			

<u>BLOCK</u>			
NOMBRE CAMPO	TIPO DE DATO	RESTRICCIÓN	DESCRIPCIÓN
identifier	VARCHAR(40)	PK	Nombre interno de un BLOCK
name	VARCHAR(40)	NOT NULL	Nombre de un BLOCK
blast_resistance	DECIMAL(11,2)	NOT NULL	Valor de la resistencia a explosiones de un BLOCK
hardness	DECIMAL(4,2)	NOT NULL	Valor de la dureza al <i>mining</i> de un BLOCK
luminosity	ENUM('yes', 'no')	NOT NULL	Indica si un BLOCK emite luz
transparency	ENUM('yes', 'no', 'partial')	NOT NULL	Indica si la luz puede atravesar un BLOCK
OBSERVACIONES DE LA TABLA: Tabla fuerte que describe las características de cada BLOCK. Está relacionada con la tabla: structure_block.			
LLAVES FORÁNEAS:			
EJEMPLO DE REGISTRO: ('oak_planks', 'oak planks', '3', '2', 'no', 'no')			

<u>STRUCTURE_BLOCK</u>			
NOMBRE CAMPO	TIPO DE DATO	RESTRICCIÓN	DESCRIPCIÓN
structure_identifier	VARCHAR(30)	FK / NOT NULL	Nombre interno de una STRUCTURE
block_identifier	VARCHAR(40)	FK / NOT NULL	Nombre interno de un BLOCK
OBSERVACIONES DE LA TABLA: Tabla transitiva que une las tablas STRUCTURE y BLOCK. Está relacionada con las tablas: structure y block.			
LLAVES FORÁNEAS: structure_identifier CON REFERENCIA a la tabla structure en ATRIBUTO (identifier) block_identifier CON REFERENCIA a la tabla block (identifier)			
EJEMPLO DE REGISTRO: ('mineshaft', 'oak_planks')			

<u>LOOT</u>			
NOMBRE CAMPO	TIPO DE DATO	RESTRICCIÓN	DESCRIPCIÓN
identifier	VARCHAR(40)	PK	Nombre interno de un LOOT
name	VARCHAR(40)	NOT NULL	Nombre de un LOOT
rarity	VARCHAR(30)	NOT NULL	Valor de rareza de un LOOT
renewable	ENUM('yes', 'no')	NOT NULL	Indica si un LOOT es renovable
stackable	TINYINT	UNSIGNED / NOT NULL	Indica el tamaño del <i>stack</i> de un LOOT
OBSERVACIONES DE LA TABLA: Tabla fuerte que describe las características de cada LOOT que se puede encontrar en una STRUCTURE. Está relacionada con la tabla: structure_loot.			
LLAVES FORÁNEAS:			
EJEMPLO DE REGISTRO: ('coal', 'coal', 'common', 'yes', 64)			

<u>STRUCTURE_LOOT</u>			
NOMBRE CAMPO	TIPO DE DATO	RESTRICCIÓN	DESCRIPCIÓN
structure_identifier	VARCHAR(30)	FK / NOT NULL	Nombre interno de una STRUCTURE
loot_identifier	VARCHAR(40)	FK / NOT NULL	Nombre interno de un LOOT
stack_size_lower	TINYINT	NOT NULL	Valor del tamaño mínimo del stack de un LOOT
stack_size_upper	TINYINT	NOT NULL	Valor del tamaño máximo del stack de un LOOT
chance	DECIMAL(5,2)	NOT NULL	Valor de la probabilidad de obtener un LOOT
OBSERVACIONES DE LA TABLA: Tabla transitiva que une las tablas STRUCTURE y LOOT. Está relacionada con las tablas: structure y loot.			
LLAVES FORÁNEAS: structure_identifier CON REFERENCIA A LA TABLA structure (identifier) loot_identifier CON REFERENCIA A LA TABLA loot (identifier)			
EJEMPLO DE REGISTRO: ('ancient_city', 'coal', 6, 15, 46.5)			

MOB			
NOMBRE CAMPO	TIPO DE DATO	RESTRICCIÓN	DESCRIPCIÓN
identifier	VARCHAR(30)	PK	Nombre interno de un MOB
name	VARCHAR(30)	UNIQUE	Nombre de un MOB
aggressiveness	ENUM('passive', 'neutral', 'hostile')	NOT NULL	Valor del nivel de agresividad de un MOB
OBSERVACIONES DE LA TABLA: Tabla fuerte que describe las características de un MOB. Está relacionada con las tablas: structure_mob y drop_mob.			
LLAVES FORÁNEAS: EJEMPLO DE REGISTRO: ('allay', 'allay', 'passive')			

STRUCTURE MOB			
NOMBRE CAMPO	TIPO DE DATO	RESTRICCIÓN	DESCRIPCIÓN
structure_identifier	VARCHAR(30)	FK / NOT NULL	Nombre interno de una STRUCTURE
mob_identifier	VARCHAR(30)	FK / NOT NULL	Nombre interno de un MOB
OBSERVACIONES DE LA TABLA: Tabla transitiva que une las tablas STRUCTURE y MOB. Está relacionada con las tablas: structure y mob.			
LLAVES FORÁNEAS: structure_identifier CON REFERENCIA A LA TABLA structure (identifier) mob_identifier CON REFERENCIA A LA TABLA mob (identifier)			
EJEMPLO DE REGISTRO: ('igloo', 'villager')			

DROPS			
NOMBRE CAMPO	TIPO DE DATO	RESTRICCIÓN	DESCRIPCIÓN
identifier	VARCHAR(30)	PK	Nombre interno de un DROP
name	VARCHAR(30)	NOT NULL	Nombre de un DROP
OBSERVACIONES DE LA TABLA: Tabla fuerte que describe las características de un DROP que se obtiene de los MOBS. Está relacionada con la tabla: drops_mob			
LLAVES FORÁNEAS: EJEMPLO DE REGISTRO: ('string', 'string')			

DROPS_MOB			
NOMBRE CAMPO	TIPO DE DATO	RESTRICCIÓN	DESCRIPCIÓN
drops_identifier	VARCHAR(30)	FK / NOT NULL	Nombre interno de un DROP
mob_identifier	VARCHAR(30)	FK / NOT NULL	Nombre interno de un MOB
OBSERVACIONES DE LA TABLA: Tabla transitiva que une las tablas DROPS y MOB. Está relacionada con las tablas: drops y mob.			
LLAVES FORÁNEAS: drop_identifier CON REFERENCIA A LA TABLA drop (identifier) mob_identifier CON REFERENCIA A LA TABLA mob (identifier)			
EJEMPLO DE REGISTRO: ('cat', 'string')			

5 Ejemplos de Consultas

5.1 Ejemplo 1

De los *loots* que se pueden conseguir de las *structures*, ¿Cuáles son los que tienen el menor valor de *chance* en cada *structure*?

Consulta

```
SELECT struc.name AS 'Nombre de la structure' , loot.name AS 'Nombre del loot con
menor probabilidad'
FROM structure AS struc
RIGHT JOIN structure_loot AS strucloot ON struc.identifier = strucloot.structure_identifier
INNER JOIN loot AS loot ON strucloot.loot_identifier = loot.identifier
WHERE (struc.identifier,strucloot.chance) IN (SELECT structure_identifier , MIN(strucloot.chance)
FROM structure_loot AS strucloot
GROUP BY strucloot.structure_identifier)
ORDER BY struc.name , loot.name;
```

Resultado

Nombre de la structure	Nombre del loot con menor probabilidad		
abandoned village	diamond horse armor	jungle temple	enchanted book
abandoned village	golden horse armor	jungle temple	golden horse armor
abandoned village	iron horse armor	jungle temple	iron horse armor
ancient city	silence armor trim smithing template	mineshaft	enchanted golden apple
bastion remnant	netherite scrap	nether fortress	rib armor trim smithing template
bonus chest	stone axe	ocean ruins	golden apple
bonus chest	stone pickaxe	ocean ruins	golden helmet
buried treasure	iron sword	ocean ruins	leather tunic
buried treasure	leather tunic	pillager outpost	enchanted book
desert pyramid	enchanted golden apple	ruined portal	bell
desert well	brick	ruined portal	block of gold
desert well	emerald	ruined portal	enchanted golden apple
desert well	stick	shipwreck	tnt
desert well	suspicious stew	stronghold	diamond horse armor
dungeon	enchanted golden apple	stronghold	enchanted book
dungeon	music disc (otherside)	stronghold	golden apple
end city	diamond horse armor	stronghold	golden horse armor
end city	golden horse armor	stronghold	iron horse armor
end city	iron horse armor	stronghold	music disc (otherside)
igloo	emerald	stronghold	saddle
jungle temple	diamond horse armor	trail ruins	beetroot seeds
jungle temple	enchanted book	trail ruins	blue stained glass pane
jungle temple	golden horse armor	trail ruins	coal
jungle temple	iron horse armor	trail ruins	dead bush
mineshaft	enchanted golden apple	trail ruins	flower pot
nether fortress	rib armor trim smithing template	trail ruins	gold nugget
ocean ruins	golden apple	trail ruins	lead
ocean ruins	golden helmet	trail ruins	light blue stained glass pane
ocean ruins	leather tunic	trail ruins	magenta stained glass pane
pillager outpost	enchanted book	trail ruins	oak hanging sign
ruined portal	bell	trail ruins	pink stained glass pane
ruined portal	block of gold	trail ruins	purple stained glass pane
ruined portal	enchanted golden apple	trail ruins	red stained glass pane
shipwreck	tnt	trail ruins	spruce hanging sign
stronghold	diamond horse armor	trail ruins	string
stronghold	enchanted book	trail ruins	wheat seeds
stronghold	golden apple	trail ruins	yellow stained glass pane
stronghold	golden horse armor	village	diamond horse armor
stronghold	iron horse armor	village	golden horse armor
stronghold	music disc (otherside)	village	iron horse armor
stronghold	saddle	woodland mansion	enchanted golden apple
trail ruins	beetroot seeds		

5.2 Ejemplo 2

De los *biomes* que contienen alguna *structure*, ¿en cuáles podemos encontrar *blocks* con *partial transparency*?

Consulta

```
SELECT DISTINCT bi.name AS Biomes
FROM block AS bl INNER JOIN structure_block AS sbl ON (bl.identifier = sbl.block_identifier)
INNER JOIN structure AS s ON (sbl.structure_identifier = s.identifier)
INNER JOIN structure_biome AS sbi ON (s.identifier = sbi.structure_identifier)
INNER JOIN biome AS bi ON (sbi.biome_identifier = bi.identifier)
WHERE transparency = 'partial';
```

Resultado

Biomes	
plains	windswept forest
meadow	windswept gravelly hills
desert	forest
savanna	flower forest
taiga	old growth pine taiga
snowy plains	old growth spruce taiga
snowy taiga	birch forest
sunflower plains	old growth birch forest
deep dark	dark forest
nether wastes	jungle
crimson forest	sparse jungle
warped forest	bamboo jungle
soul sand valley	river
ocean	frozen river
deep ocean	swamp
frozen ocean	mangrove swamp
deep frozen ocean	beach
cold ocean	stony shore
deep cold ocean	snowy beach
lukewarm ocean	ice spikes
deep lukewarm ocean	savanna plateau
warm ocean	windswept savanna
mushroom fields	badlands
cherry grove	wooded badlands
grove	eroded badlands
snowy slopes	dripstone caves
jagged peaks	lush caves
frozen peaks	the void
stony peaks	end midlands
windswept hills	end highlands
	the end
	basalt deltas

5.3 Ejemplo 3

De las *structures* que cuentan con *mobs*, ¿Cuáles son los *drops* que se pueden conseguir en cada *structure*?

Consulta

```
SELECT struc.name AS 'Nombre de la estructura' , drops.name AS 'Nombre del drop'
FROM structure AS struc
INNER JOIN structure_mob AS strucmob ON struc.identifier = strucmob.structure_identifier
INNER JOIN mob AS mob ON strucmob.mob_identifier = mob.identifier
INNER JOIN drops_mob AS dropmob ON mob.identifier = dropmob.mob_identifier
LEFT JOIN drops AS drops ON dropmob.drops_identifier = drops.identifier
ORDER BY struc.name;
```

Resultado

Nombre de la estructura	Nombre del drop		
abandoned village	rotten flesh	swamp hut	gunpowder
ancient city	sculk catalyst	swamp hut	glass bottle
bastion remnant	piglin head	swamp hut	spider eye
bastion remnant	ochre froglight	swamp hut	redstone dust
bastion remnant	verdant froglight	swamp hut	sugar
bastion remnant	pearlescent froglight	village	glowstone dust
bastion remnant	magma cream	village	string
bastion remnant	leather	village	raw mutton
bastion remnant	raw porkchop	village	wool
end city	shulker shell	village	raw porkchop
igloo	rotten flesh	village	iron ingot
mineshaft	spider eye	village	poppy
mineshaft	string	village	leather
nether fortress	gold ingot	woodland mansion	leather
nether fortress	gold nugget	woodland mansion	raw beef
nether fortress	rotten flesh	woodland mansion	string
nether fortress	wither skeleton skull	woodland mansion	illager banner
nether fortress	bone	woodland mansion	emerald
nether fortress	coal		ominous banner
nether fortress	skeleton skull		emerald
nether fortress	arrow		totem of undying
nether fortress	bone		
nether fortress	ochre froglight		
nether fortress	verdant froglight		
nether fortress	pearlescent froglight		
nether fortress	magma cream		
nether fortress	blaze rod		
ocean monument	random fish		
ocean monument	prismarine shard		
ocean monument	random fish		
ocean monument	tide armor trim		
ocean monument	wet sponge		
ocean monument	prismarine shard		
ocean ruins	rotten flesh		
ocean ruins	copper ingot		
pillager outpost	iron ingot		
pillager outpost	poppy		
swamp hut	stick		

5.4 Ejemplo 4

¿Cuáles son las 10 *structures* que se conforman de la mayor cantidad de *blocks* diferentes?

Consulta

```
SELECT struc.name AS 'Nombre de la estructura', COUNT(*) AS 'Cantidad de bloques  
diferentes'  
FROM structure AS struc  
LEFT JOIN structure_block AS strucblock ON struc.identifier = strucblock.structure_identifier  
GROUP BY struc.identifier  
ORDER BY 'Cantidad de bloques diferentes' DESC  
LIMIT 10;
```

Resultado

Nombre de la estructura	Cantidad de bloques diferentes
abandoned village	171
village	171
woodland mansion	101
ancient city	70
trail ruins	63
shipwreck	43
ocean ruins	32
igloo	28
bastion remnant	26
stronghold	26

5.5 Ejemplo 5

¿Cuáles son los 10 *loots* más comunes entre las distintas *structures*, y cuántas veces aparecen?

Consulta

```
SELECT l.name AS Loot, count(l.name) AS appearances
FROM loot as l INNER JOIN structure_loot AS sl
ON (l.identifier = sl.loot_identifier)
GROUP BY l.name
ORDER BY appearances DESC
LIMIT 10;
```

Resultado

Loot	Appearances
emerald	37
gold ingot	22
bread	22
iron ingot	21
coal	18
gold nugget	15
diamond	14
saddle	14
wheat	14
enchanted book	13

5.6 Ejemplo 6

¿Cuántos *mobs* se pueden encontrar en cada *structure*?

Consulta

```
SELECT b.identifier, COUNT(*) AS 'Número de mobs'
FROM biome AS b
LEFT JOIN structure_biome AS sb ON b.identifier = sb.biome.identifier
INNER JOIN structure AS s ON sb.structure.identifier = s.identifier
INNER JOIN structure_mob AS sm ON s.identifier = sm.structure.identifier
INNER JOIN mob AS m ON sm.mob.identifier = m.identifier GROUP BY b.identifier;
```

Resultado

identifier	Numero_de_mobs		
		jungle	2
		lukewarm_ocean	3
		lush_caves	5
		mangrove_swamp	2
badlands	2	meadow	14
bamboo_jungle	2	mushroom_fields	2
basalt_deltas	5	nether_wastes	9
beach	2	ocean	3
birch_forest	2	old_growth_birch_forest	2
cherry_grove	5	old_growth_pine_taiga	2
cold_ocean	3	old_growth_spruce_taiga	2
crimson_forest	9	plains	14
dark_forest	8	river	2
deep_cold_ocean	5	savanna	14
deep_dark	3	savanna_plateau	2
deep_frozen_ocean	5	snowy_beach	2
deep_lukewarm_ocean	5	snowy_plains	16
deep_ocean	5	snowy_slopes	7
desert	14	snowy_taiga	16
dripstone_caves	5	soul_sand_valley	9
end_highlands	1	sparse_jungle	2
end_midlands	1	stony_peaks	5
eroded_badlands	2	stony_shore	2
flower_forest	2	sunflower_plains	14
forest	2	swamp	4
frozen_ocean	3	taiga	38
frozen_peaks	5	the_end	1
frozen_river	2	the_void	2
grove	5	warm_ocean	6
ice_spikes	2	warped_forest	9
jagged_peaks	5	windswept_forest	2
		windswept_gravelly_hills	2
		windswept_hills	2
		windswept_savanna	2
		wooded_badlands	2

5.7 Ejemplo 7

¿Cuál es el número de *drops* que puede soltar los *mobs* de cada estructura?

Consulta

```
SELECT COUNT(dm.drops_idenfifier) AS 'Número de drops', mb.name AS 'Mobs',  
str.name AS 'Structure'  
FROM drops_mob AS dm  
INNER JOIN mob AS mb  
INNER JOIN structure AS str  
WHERE dm.mob_idenfifier = mb.idenfifier  
GROUP BY mb.name, str.name  
LIMIT 70, 10;
```

Resultado

Número de dops	Mobs	Structure
1	cat	stronghold
1	cat	swamp hut
1	cat	trail ruins
1	cat	village
1	cat	void start platform
1	cat	woodland mansion
2	cave spider	abandoned village
2	cave spider	ancient city
2	cave spider	bastion remnant
2	cave spider	bonus chest

5.8 Ejemplo 8

¿Cuántas *structures* de cada *dimension* comparten la misma *temperature*?

Consulta

```
SELECT bio.temperature , dimen.name , count(*)
FROM structure AS struc
INNER JOIN structure_biome AS strucbio ON struc.identifier = strucbio.structure_identifier
INNER JOIN biome AS bio ON strucbio.biome_identifier = bio.identifier
INNER JOIN dimension AS dimen ON bio.dimension_identifier = dimen.identifier
GROUP BY bio.temperature , dimen.identifier
ORDER BY bio.temperature , dimen.name;
```

Resultado

temperature	name	count(*)
-0.70	overworld	14
-0.50	overworld	12
-0.30	overworld	8
-0.20	overworld	7
0.00	overworld	42
0.05	overworld	8
0.20	overworld	25
0.25	overworld	25
0.30	overworld	8
0.50	overworld	98
0.50	the end	9
0.60	overworld	13
0.70	overworld	20
0.80	overworld	56
0.90	overworld	8
0.95	overworld	22
1.00	overworld	7
2.00	nether	15
2.00	overworld	54