1. Player

- **Propósito**: Representa al jugador del videojuego.
- **Atributos**:
    - `player_id` (int, PK): Identificador único del jugador.
    - `name` (string): Nombre del jugador.
    - `health`, `stamina`, `mana` (int): Estadísticas del personaje.
    - `gold` (int): Recursos acumulados.
    - `current_floor` (int): Piso actual en la partida.
    - `save_slot` (int): Posición activa de guardado.
- **Justificación**: Es el actor principal del sistema. Separa datos permanentes del jugador respecto a partidas (“GameRun”).
- **Relaciones**:
    - 1:N con `Inventory`, `GameRun`, `SaveSlot`, `PlayerUpgrade`.

2. Inventory

- **Propósito**: Representa los espacios de inventario que un jugador tiene.
- **Atributos**:
    - `player_id` (int, PK, FK): Jugador al que pertenece el slot.
    - `slot_number` (int, PK): Posición del slot.
    - `item_id` (int, FK): Objeto almacenado en el slot.
- **Justificación**: Permite modelar un inventario con posiciones fijas. Se usa clave primaria compuesta para evitar duplicidad.
- **Relaciones**:
    - N:1 con `Item`.
    - N:1 con `Player`.

3. Item

- **Propósito**: Almacena información sobre los objetos disponibles.
- **Atributos**:
    - `item_id` (int, PK): Identificador del objeto.
    - `name` (string): Nombre del objeto.

    - `type` (string): Tipo (arma, pocion, etc.).
    - `effect` (string): Efecto del objeto.
    - `is_permanent` (boolean): Si es permanente o temporal.
- **Justificación**: Permite manejar un catálogo reutilizable de objetos.
- **Relaciones**:
    - 1:N con `Inventory`.

4. Upgrade

- **Propósito**: Define las mejoras posibles que un jugador puede obtener.
- **Atributos**:
    - `upgrade_id` (int, PK): Identificador de la mejora.
    - `name`, `effect` (string): Información de la mejora.
    - `is_permanent` (boolean): Si permanece entre runs.
- **Justificación**: Separa las mejoras permanentes del jugador y permite su reaprovechamiento.
- **Relaciones**:
    - 1:N con `PlayerUpgrade`.

5. PlayerUpgrade

- **Propósito**: Relaciona jugadores con sus mejoras adquiridas.
- **Atributos**:
    - `player_id` (int, FK)
    - `upgrade_id` (int, FK)
    - `acquired_at_floor` (int): Nivel donde se obtuvo.
- **Justificación**: Permite registrar upgrades por jugador. Modela una relación muchos a muchos.
- **Relaciones**:
    - N:1 con `Player`.
    - N:1 con `Upgrade`.

6. GameRun

- **Propósito**: Representa una sesión de juego iniciada por un jugador.
- **Atributos**:

    - `run_id` (int, PK)
    - `player_id` (int, FK)
    - `start_time`, `end_time` (datetime): Tiempo de juego.
    - `completed` (boolean): Si se completó la sesión.
- **Justificación**: Almacena estadísticas y seguimiento de runs.
- **Relaciones**:
    - N:1 con `Player`.

7. SaveSlot

- **Propósito**: Permite tener varios puntos de guardado por jugador.
- **Atributos**:
    - `slot_id` (int, PK)
    - `player_id` (int, FK)
    - `slot_number` (int): Posición del slot.
    - `last_saved` (datetime): Fecha del último guardado.
- **Justificación**: Da soporte a sistemas con múltiples partidas por jugador.
- **Relaciones**:
    - N:1 con `Player`.

8. Room

- **Propósito**: Define las habitaciones que componen el mundo del juego.
- **Atributos**:
    - `room_id` (int, PK)
    - `room_type` (string): Tipo (combate, tienda, jefe).
    - `floor_number` (int): Piso al que pertenece.
    - `is_cleared` (boolean): Si fue completada.
- **Justificación**: Permite instanciar niveles con enemigos variables.
- **Relaciones**:
    - 1:N con `RoomEnemy`

9. Enemy

- **Propósito**: Define los enemigos del juego.
- **Atributos**:
    - `enemy_id` (int, PK)
    - `name`, `type` (string)
    - `health`, `damage` (int)
    - `floor_number` (int): Piso donde aparece normalmente
- **Justificación**: Separa los tipos de enemigos para su reuso.
- **Relaciones**:
    - 1:N con `RoomEnemy`

10. RoomEnemy

- **Propósito**: Relación entre cuartos y enemigos.
- **Atributos**:
    - `room_id`, `enemy_id` (int, PK compuesta, FK)
- **Justificación**: Modela la relación muchos a muchos entre cuartos y enemigos.
- **Relaciones**:
    - N:1 con `Room`
    - N:1 con `Enemy`