Game_universe.db

In a vast and mysterious fantasy kingdom, inhabited by various races (Races table: RaceName), creatures (Creatures table: CreatureName, Type), and magical beings, lies a legendary place known as the Kingdom of Eldoria (Locations table: LocationName).).

This kingdom is made up of majestic cities, lush forests, and vast plains (Location Table: Terrain), each inhabited by a variety of beings, from humans and elves (Character Table: RaceID) to dragons and magical creatures.

However, Eldoria is in danger due to the growing darkness that threatens to consume the world. To combat this darkness, brave adventurers embark on epic quests (Quests table:

QuestName), summoning powerful spells (MagicSpells table: SpellName, Element) and using lost magical artifacts (MagicArtifacts table: ArtifactName) to restore balance.

Adventurers include warriors of various races (Character Table: RaceID), experts in the use of swords and bows, and wizards capable of casting elemental spells (Character Spell Table: SpellID). As they explore the lands of Eldoria, they face deadly challenges, battle fearsome creatures (Creatures table: CreatureName), and collect valuable treasures and magical artifacts that grant them extraordinary powers (Items table: ItemName, Value).
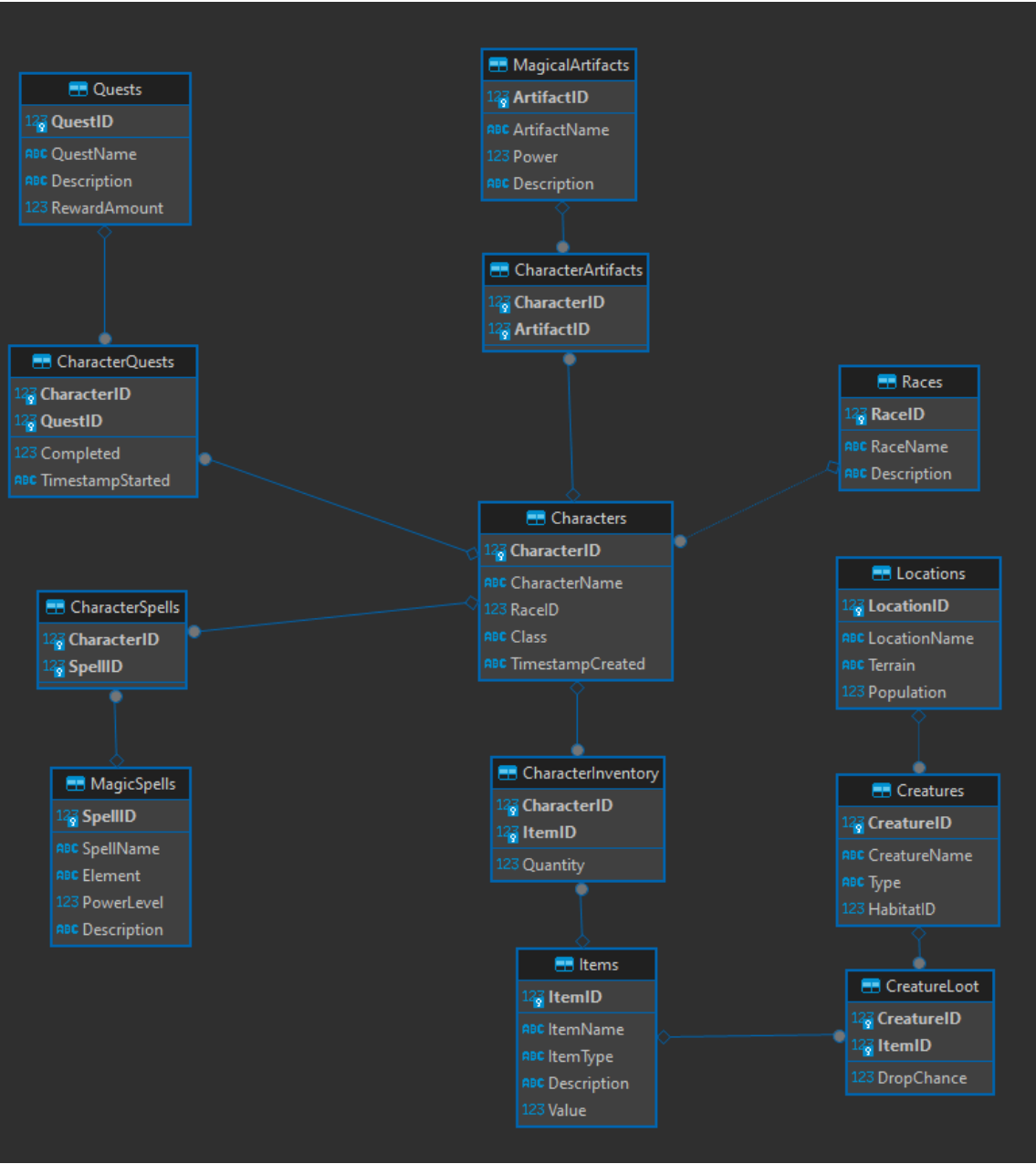
With each completed quest, adventurers earn recognition and fortune (Quest Chart: RewardAmount), forging alliances with other races (Character Chart: RaceID) and building a legacy of heroism in the fight against the darkness that threatens to destroy their world.

Contenido

Entity-Relationship Diagram

Quests
- QuestID
- QuestName
- Description
- RewardAmount

MagicalArtifacts
- ArtifactID
- ArtifactName
- Power
- Description

CharacterArtifacts
- CharacterID
- ArtifactID

CharacterQuests
- CharacterID
- QuestID
- Completed
- TimestampStarted

Races
- RaceID
- RaceName
- Description

Characters
- CharacterID
- CharacterName
- RaceID
- Class
- TimestampCreated

Locations
- LocationID
- LocationName
- Terrain
- Population

CharacterSpells
- CharacterID
- SpellID

MagicSpells
- SpellID
- SpellName
- Element
- PowerLevel
- Description

CharacterInventory
- CharacterID
- ItemID
- Quantity

Creatures
- CreatureID
- CreatureName
- Type
- HabitatID

Items
- ItemID
- ItemName
- ItemType
- Description
- Value

CreatureLoot
- CreatureID
- ItemID
- DropChance

Races:

Columns:

- RaceID: Unique identifier for the race (INTEGER).
- RaceName: Name of the race (TEXT, cannot be null).
- Description: Description of the race (TEXT).

Definition:

```sql
-- Create table for races
CREATE TABLE Races (
    RaceID INTEGER PRIMARY KEY,
    RaceName TEXT NOT NULL,
    Description TEXT
);
```

Example:

	RaceID	RaceName	Description
1	1	Human	Versatile and adaptable race of Eldoria.
2	2	Elf	Graceful and long-lived race with affinity for nature.
3	3	Dwarf	Stout and skilled craftsmen of the mountains.
4	4	Orc	Fierce and strong-willed warriors of Eldoria.
5	5	Goblin	Cunning and resourceful inhabitants of caves and forests.

Locations:

Columns:

- LocationID: Unique identifier for the location (INTEGER).
- LocationName: Name of the location (TEXT, cannot be null).
- Terrain: Terrain type of the location (TEXT).
- Population: Population of the location (INTEGER).

Definition:

```
-- Create table for locations
CREATE TABLE Locations (
    LocationID INTEGER PRIMARY KEY,
    LocationName TEXT NOT NULL,
    Terrain TEXT,
    Population INTEGER
);
```

Example:

	LocationID	LocationName	Terrain	Population
1	1	Eldoria City	Urban	10.000
2	2	Elderwood Forest	Forest	5.000
3	3	Dragonspire Mountains	Mountain	3.000
4	4	Sunset Plains	Plains	8.000
5	5	Mystic Isles	Island	2.000

Creatures:

Columns:

- CreatureID: Unique identifier for the creature (INTEGER).
- CreatureName: Name of the creature (TEXT, cannot be null).
- Type: Type of the creature (TEXT).
- HabitatID: ID of the habitat of the creature, foreign key referencing the Locations table (INTEGER).

Definition:

```sql
-- Create table for creatures
CREATE TABLE Creatures (
    CreatureID INTEGER PRIMARY KEY,
    CreatureName TEXT NOT NULL,
    Type TEXT,
    HabitatID INTEGER,
    FOREIGN KEY (HabitatID) REFERENCES Locations(LocationID)
);
```

Example:

123 CreatureID	ABC CreatureName	ABC Type	123 HabitatID
1	Dragon	Legendary	3
2	Direwolf	Beast	2
3	Giant Spider	Insectoid	2
4	Treant	Elemental	2
5	Griffin	Mythical	4

MagicSpells:

Columns:

- SpellID: Unique identifier for the spell (INTEGER).
- SpellName: Name of the spell (TEXT, cannot be null).
- Element: Element of the spell (TEXT).
- PowerLevel: Power level of the spell (INTEGER).
- Description: Description of the spell (TEXT).

Definition:

```sql
-- Create table for magic spells
CREATE TABLE MagicSpells (
    SpellID INTEGER PRIMARY KEY,
    SpellName TEXT NOT NULL,
    Element TEXT,
    PowerLevel INTEGER,
    Description TEXT
);
```

Example:

	SpellID	SpellName	Element	PowerLevel	Description
1	1	Fireball	Fire	5	Launches a powerful ball of fire.
2	2	Healing Light	Light	3	Heals wounds and ailments with radiant energy.
3	3	Frost Nova	Ice	4	Creates a freezing blast in all directions.
4	4	Thunderbolt	Electricity	5	Strikes enemies with a powerful lightning bolt.
5	5	Earthquake	Earth	5	Causes the ground to shake violently, damaging foes.

Quest:

Columns:

- QuestID: Unique identifier for the quest (INTEGER).
- QuestName: Name of the quest (TEXT, cannot be null).
- Description: Description of the quest (TEXT).
- RewardAmount: Amount of reward for completing the quest (REAL).

Definition:

```sql
-- Create table for quests
CREATE TABLE Quests (
    QuestID INTEGER PRIMARY KEY,
    QuestName TEXT NOT NULL,
    Description TEXT,
    RewardAmount REAL
);
```

Example:

QuestID	QuestName	Description	RewardAmount
1	Defeat the Dragon	Slay the fearsome dragon terrorizing the countryside.	500
2	Retrieve the Artifact	Recover a lost artifact hidden deep within the Mystic Isles.	1.000
3	Protect the Village	Defend the Sunset Plains village from orc raiders.	300
4	Explore the Elderwood	Brave the dangers of the Elderwood Forest to uncover its secrets.	200
5	Conquer the Goblin Lair	Clear out the goblin infestation in the Dragonspire Mountains.	400

Characters:

Columns:

- CharacterID: Unique identifier for the character (INTEGER).
- CharacterName: Name of the character (TEXT, cannot be null).
- RaceID: ID of the character's race, foreign key referencing the Races table (INTEGER).
- Class: Class of the character (TEXT).
- TimestampCreated: Timestamp of when the character was created (TIMESTAMP).

Definition:

```sql
-- Create table for characters
CREATE TABLE Characters (
    CharacterID INTEGER PRIMARY KEY,
    CharacterName TEXT NOT NULL,
    RaceID INTEGER,
    Class TEXT,
    TimestampCreated TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (RaceID) REFERENCES Races(RaceID)
);
```

Example:

	CharacterID	CharacterName	RaceID	Class	TimestampCreated
1	1	Auron	1	Warrior	2024-05-04 13:17:18
2	2	Lorien	2	Mage	2024-05-04 13:17:18
3	3	Thorin	3	Warrior	2024-05-04 13:17:18
4	4	Grom	4	Warrior	2024-05-04 13:17:18
5	5	Sneaky Pete	5	Rogue	2024-05-04 13:17:18

CharacterQuests:

Columns:

- CharacterID: ID of the character, foreign key referencing the Characters table (INTEGER).
- QuestID: ID of the quest, foreign key referencing the Quests table (INTEGER).
- Completed: Indicates if the quest has been completed by the character (BOOLEAN).
- TimestampStarted: Timestamp of when the quest was started by the character (TIMESTAMP).

Definition:

```sql
-- Create table for character quests (many-to-many relationship)
CREATE TABLE CharacterQuests (
    CharacterID INTEGER,
    QuestID INTEGER,
    Completed BOOLEAN,
    TimestampStarted TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (CharacterID) REFERENCES Characters(CharacterID),
    FOREIGN KEY (QuestID) REFERENCES Quests(QuestID),
    PRIMARY KEY (CharacterID, QuestID)
);
```

Example:

	CharacterID	QuestID	Completed	TimestampStarted
1	1	1	1	2024-05-04 13:17:18
2	2	2	0	2024-05-04 13:17:18
3	3	3	1	2024-05-04 13:17:18
4	4	4	0	2024-05-04 13:17:18
5	5	5	1	2024-05-04 13:17:18

Items:

Columns:

- ItemID: Unique identifier for the item (INTEGER).
- ItemName: Name of the item (TEXT, cannot be null).
- ItemType: Type of the item (TEXT).
- Description: Description of the item (TEXT).
- Value: Value of the item (REAL).

Definition:

```sql
-- Create table for items
CREATE TABLE Items (
    ItemID INTEGER PRIMARY KEY,
    ItemName TEXT NOT NULL,
    ItemType TEXT,
    Description TEXT,
    Value REAL
);
```

Example:

	ItemID	ItemName	ItemType	Description	Value
1	1	Sword of Flames	Weapon	A sword imbued with fiery magic.	200
2	2	Staff of the Archmage	Weapon	A powerful staff capable of channeling immense magical energy.	300
3	3	Dwarven Plate Armor	Armor	Heavy armor forged by skilled dwarven blacksmiths.	250
4	4	Amulet of Healing	Accessory	An amulet that grants the wearer accelerated healing abilities.	150
5	5	Potion of Invisibility	Consumable	A potion that renders the drinker invisible for a short duration.	50

CharacterInventory:

Columns:

- CharacterID: ID of the character, foreign key referencing the Characters table (INTEGER).
- ItemID: ID of the item, foreign key referencing the Items table (INTEGER).
- Quantity: Quantity of the item in the character's inventory (INTEGER).

Definition:

```sql
-- Create table for character inventory (many-to-many relationship)
CREATE TABLE CharacterInventory (
    CharacterID INTEGER,
    ItemID INTEGER,
    Quantity INTEGER,
    FOREIGN KEY (CharacterID) REFERENCES Characters(CharacterID),
    FOREIGN KEY (ItemID) REFERENCES Items(ItemID),
    PRIMARY KEY (CharacterID, ItemID)
);
```

Example:

	CharacterID	ItemID	Quantity
1	1	1	1
2	2	2	1
3	3	3	1
4	4	4	1
5	5	5	3

MagicalArtifacts:

Columns:

- ArtifactID: Unique identifier for the artifact (INTEGER).
- ArtifactName: Name of the artifact (TEXT, cannot be null).
- Power: Power of the artifact, stored as JSON.
- Description: Description of the artifact (TEXT).

Definition:

```sql
-- Create table for magical artifacts
CREATE TABLE MagicalArtifacts (
    ArtifactID INTEGER PRIMARY KEY,
    ArtifactName TEXT NOT NULL,
    Power JSON,
    Description TEXT
);
```

Example:

ArtifactID	ArtifactName	Power	Description
1	Crystal of Ages	{"Power": "Time manipulation"}	An ancient crystal with the power to control time.
2	Orb of Elements	{"Power": "Elemental control"}	A mystical orb that grants mastery over the elements.
3	Scepter of Light	{"Power": "Divine energy"}	A divine scepter said to be blessed by the gods.
4	Crown of Shadows	{"Power": "Shadow manipulation"}	A dark crown with the power to command shadows.
5	Chalice of Life	{"Power": "Healing"}	A sacred chalice said to hold the essence of life itself.

CreatureLoot:

Columns:

- CreatureID: ID of the creature, foreign key referencing the Creatures table (INTEGER).
- ItemID: ID of the item, foreign key referencing the Items table (INTEGER).
- DropChance: Probability of the creature dropping the item (REAL).

Definition:

```sql
-- Create table for creature loot (many-to-many relationship)
CREATE TABLE CreatureLoot (
    CreatureID INTEGER,
    ItemID INTEGER,
    DropChance REAL,
    FOREIGN KEY (CreatureID) REFERENCES Creatures(CreatureID),
    FOREIGN KEY (ItemID) REFERENCES Items(ItemID),
    PRIMARY KEY (CreatureID, ItemID)
);
```

Example:

CreatureLoot | Enter a SQL expression to filter results (use Ctrl+Space)

	CreatureID	ItemID	DropChance
1	1	1	0,5
2	2	2	0,3
3	3	3	0,2
4	4	4	0,4
5	5	5	0,6

CharacterSpells:

Columns:

- CharacterID: ID of the character, foreign key referencing the Characters table (INTEGER).
- SpellID: ID of the spell, foreign key referencing the MagicSpells table (INTEGER).

Definition:

```sql
-- Create table for character spells (many-to-many relationship)
CREATE TABLE CharacterSpells (
    CharacterID INTEGER,
    SpellID INTEGER,
    FOREIGN KEY (CharacterID) REFERENCES Characters(CharacterID),
    FOREIGN KEY (SpellID) REFERENCES MagicSpells(SpellID),
    PRIMARY KEY (CharacterID, SpellID)
);
```

Example:

	CharacterID	SpellID
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5

CharacterArtifacts:

Columns:

- CharacterID: ID of the character, foreign key referencing the Characters table (INTEGER).
- ArtifactID: ID of the artifact, foreign key referencing the MagicalArtifacts table (INTEGER).

Definition:

```sql
-- Create table for character artifacts (many-to-many relationship)
CREATE TABLE CharacterArtifacts (
    CharacterID INTEGER,
    ArtifactID INTEGER,
    FOREIGN KEY (CharacterID) REFERENCES Characters(CharacterID),
    FOREIGN KEY (ArtifactID) REFERENCES MagicalArtifacts(ArtifactID),
    PRIMARY KEY (CharacterID, ArtifactID)
);
```

Example:

	CharacterID	ArtifactID
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5

Query Samples

-- Get all characters of a certain race:

```sql
SELECT *
FROM Characters
WHERE RaceID = (SELECT RaceID FROM Races WHERE RaceName = 'Human');
```

-- Get the name and description of all quests completed by a specific character:

```sql
SELECT Quests.QuestName, Quests.Description
FROM Quests
JOIN CharacterQuests ON Quests.QuestID = CharacterQuests.QuestID
WHERE CharacterQuests.CharacterID = 1 AND CharacterQuests.Completed = 1;
```

-- Obtain all fire spells of level 3 or higher:

```sql
SELECT *
FROM MagicSpells
WHERE Element = 'Fire' AND PowerLevel >= 3;

-- Common Table Expression (CTE) - Get a character's inventory along with item names
and descriptions:
WITH CharacterInventoryDetails AS (
    SELECT CI.CharacterID, CI.ItemID, CI.Quantity, Items.ItemName, Items.Description
    FROM CharacterInventory CI
    JOIN Items ON CI.ItemID = Items.ItemID
    WHERE CI.CharacterID = 1
)
SELECT *
FROM CharacterInventoryDetails;
```