Yu-Gi-Oh! Card Database - SQL & Data Modeling

Documentation written and Database designed by Fernando Cruz.

Purpose

This database is designed to be a complete and accurate catalog of Yu-Gi-Oh! trading cards. Its main goals are:

- Store the full details of cards (name, stats, archetype, description, etc.).
- Represent complex relationships between card types, subtypes, attributes, archetypes and specific mechanics like Pendulums or Links.
- Provide a flexible, maintainable model that can adapt as the game evolves. This project started out of personal passion for Yu-Gi-Oh!, but also as a technical exercise: this trading card game presents a *highly dynamic domain* where relationships and categories are continuously evolving, making it a challenging modelling case.

Functional Requirements

In-Scope (what a user should be able to do):

- Search and Browse: Look up cards by name, see their full details (attack, level, effect text, etc.).
- Filter and grouping: Filter them by type (Monster, Spell, Trap), subtype (e.g. "Effect", "Quick Spell", "Counter Trap", etc.), attribute, archetype, and other properties.
- Banlist tracking: Identify whether a card is forbidden/limited in a given format.

Out-of-Scope:

- Gameplay & Rules: The database does not simulate rules, duels or deck building.
- Pricing & Collection Tracking: This database won't track card prices, market values, or a user's personal collection (e.g., which cards they own, the condition of those cards), as of now. It might be considered as an expansion in the future.
- Images: As of now, this database doesn't store the cards' images nor URLs of any kind, but it's being considered for future versions.

Representation

Entities

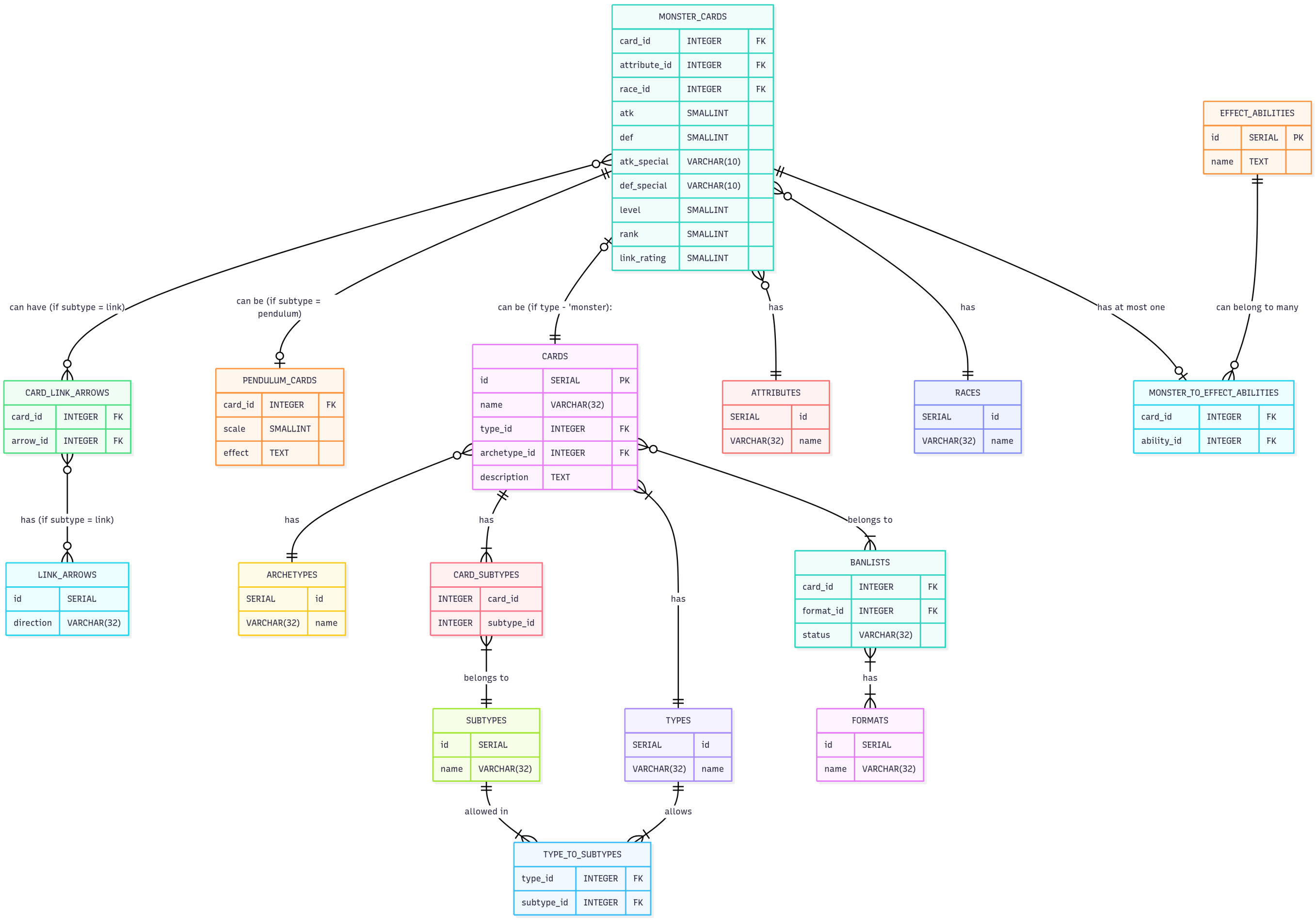
- `cards`
- `types`
- `subtypes`
- `types_to_subtypes`
- `card_subtype_relations`
- `archetypes`
- `monster_cards`
- `attributes`

- `races`
- `effect_abilities`
- `monster_to_effect_abilities`
- `pendulum_cards`
- `link_arrows`
- `card_link_arrows`
- `banlists`
- `formats`

Design Descisions

- Entities vs Enums: Instead of ENUMs, entities like `types`, `subtypes`, and `races` are stored in tables.
  - Reason: the game is dynamic and categories are constantly evolving over time (e.g. new monster types, subtypes, new attributes, etc.). Storing them as entities allows easier querying and maintenance.
- Formats table: Added to prepare for differences in TCG/OCG and other formats (as it has been stated before, future versions for storing Duel Links and Rush Duel cards are being considered).

ER Diagram



Relationships

- A `card` belongs to one `type`, and may have one `archetype`.
- A `card` can have multiple `subtypes` via `card_subtypes` IF `card` is of `type` "monster".

- `types` define which `subtypes` are valid through `type_to_subtypes` (e.g.: this ensures that a `card` of `type` "monster" doesn't have a `subtype` "Continuous" related to them, since such subtype belongs to `cards` of `type` "spell" and "trap").
- `monster_cards` extend `cards` with stats, attributes, races and special properties (sometimes, depending on the `subtype` assigned to them).
- `link_monsters` connect to `link_arrows` through `card_link_arrows`.
- `effect_abilities` are modeled separetely and belong to a specific `type` and `subtype` of cards (`type` = "monster", `subtype` = effect).
- `banlists` link cards to a `format` with a status (forbidden, limited, semi-limited).

Optimizations

As of now, no optimizations have been actively made. That being said, I'm well aware that there need to be several optimizations to ensure a better experience when querying the data through this database, such as:

- Views: Views that ensures that querying through any card in full detail doesn't take a boilerplate of joins.
- Indexes: for perfomance sake, I'm gonna investigate for future versions which are the tables and columns that need to be indexed. That can only happen once I know which data is the most commonly retrieved for determined cases.

Limitations

- Complex Monster Types: The database accurately stores a card's type and its subtype (e.g., "Monster" and its subtypes: "Fusion", "Effect", "Ritual"; “Spell” and its subtypes: “Normal”, “Field”, “Equip”, etc.) but doesn't inherently understand the rules behind these types. An application using this database would need to handle that logic.
- Card Text Parsing: The desc field is just a block of text. The database doesn't understand the meaning of the text. For example, finding all cards that "special summon" would require complex text searching and wouldn't be 100% accurate.
- Rulings: It doesn't store specific game rulings or interactions between cards. It's a catalog of facts, not a rulebook.
- Limited Archetype Logic: Some cards support an archetype without having its name in their text (or even in the archetype field if it's not officially tagged). The database might not perfectly represent every fan-defined "series" of cards, only the official ones.