Board Game Database
By Tom Dale

http://web.engr.oregonstate.edu/~dalet/CS340/FinalProject/home.php

Outline

My database is for managing board game collections. For some people like myself, I have a ton of board games, way too many board games and having a database where I can see them all in one place, and get some details about them would be handy. Also by tracking player data you track fun stats like player wins to see who has played what games. It will also track dates played so you can see the last time you played a specific game. The database will track these by using 5 main tables: player, game, type, game_type, and result. This will allow you to track different data, and link tables together to find meaningful data inside the database.

Database Outline in Words

There are 5 tables inside the database: player, game, type, game_type, and result. Player will hold the player's first and last name which are required data. The combination of first and last name will be a unique identify so no player is accidently added in twice.

Game will hold the name, minimum players needed, and maximum players needed which are all required data. The name will be a unique identifier so no single game is added multiple times.
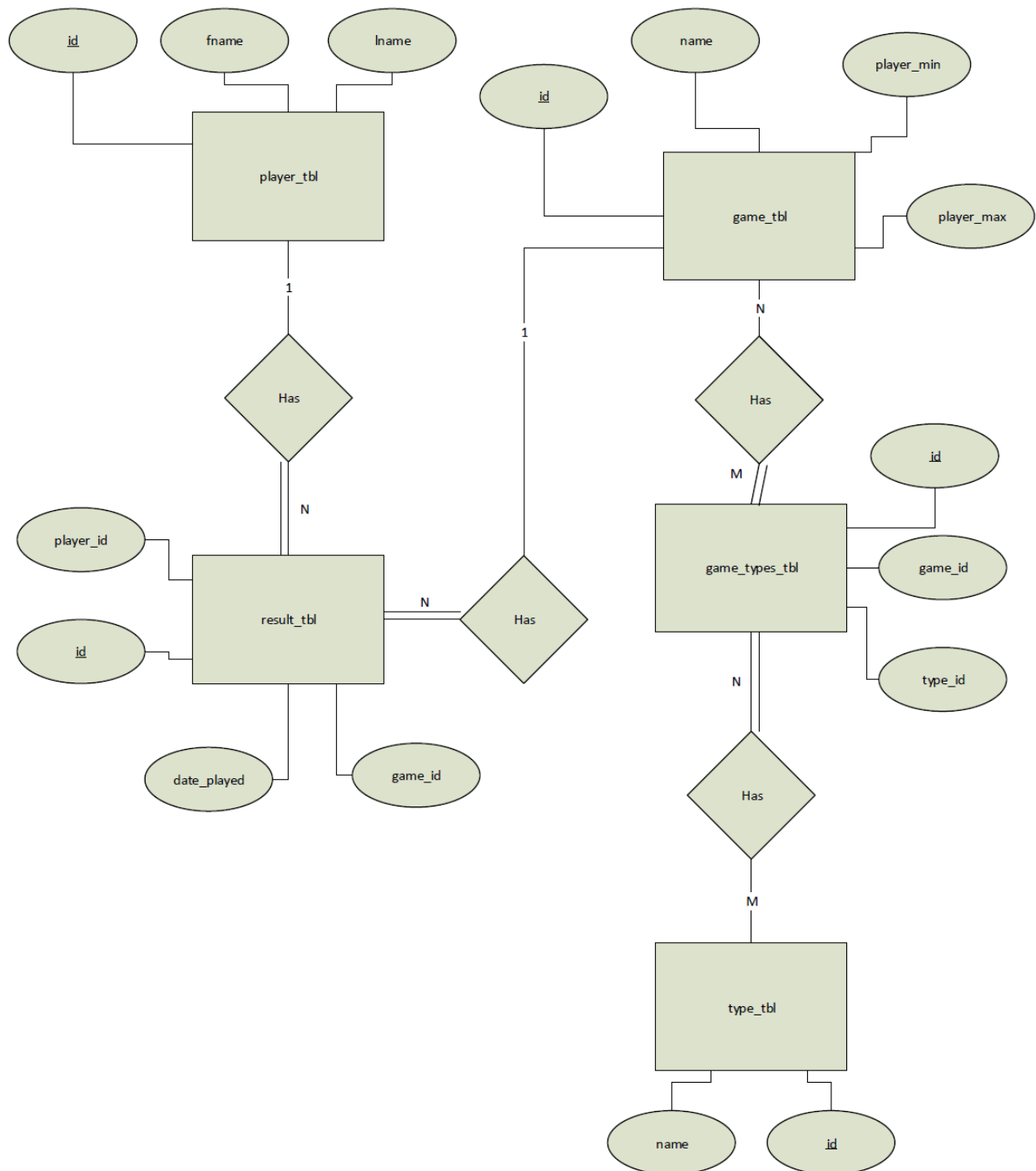
Type will hold different categories for board games to be described by such as card game, resource management, rpg, etc, again the name of the type will be unique.

Game_types will link the game and the type database by using the type_id and game_id to show which games have which types, so when looking at a game you can multiple types that are associated with that game. Because the game_types table links the two, both a type_id and game_id are required to make an entry, as well as being a unique combination, this also allows you to add multiple types to one game.

Result will hold the date the result happened, as well as a link it to the game table using a game_id so you know which game was played, as well as to a player via a player_id which will designate the winner of that particular game result. Result has no unique data as multiple games of the same board game can be played on the same day with the same winner.

All references to a different table's id is a foreign key. All data is cascades on delete for foreign keys. For example if you delete a player all his winning results will disappear with it or if you delete a game, all its game_types and results will disappear.

ER Diagram of Board Game Database

Board Game Database Schema

player_tbl
- 🔑 id INT
- ◇ fname VARCHAR(255)
- ◇ lname VARCHAR(255)

Indexes
- PRIMARY
- fname_lname_UNIQUE

game_tbl
- 🔑 id INT
- ◇ name VARCHAR(255)
- ◇ player_min INT
- ◇ player_max INT

Indexes
- PRIMARY
- name_UNIQUE

resultl_tbl
- 🔑 id INT
- ◆ game_id INT
- ◆ player_id INT
- ◇ date_played DATE

Indexes
- PRIMARY
- game_id_idx
- player_id_idx

game_types_tbl
- 🔑 id INT
- ◆ game_id INT
- ◆ type_id INT

Indexes
- PRIMARY
- game_id_idx
- type_id_idx

type_tbl
- 🔑 id INT
- ◇ name VARCHAR(255)

Indexes
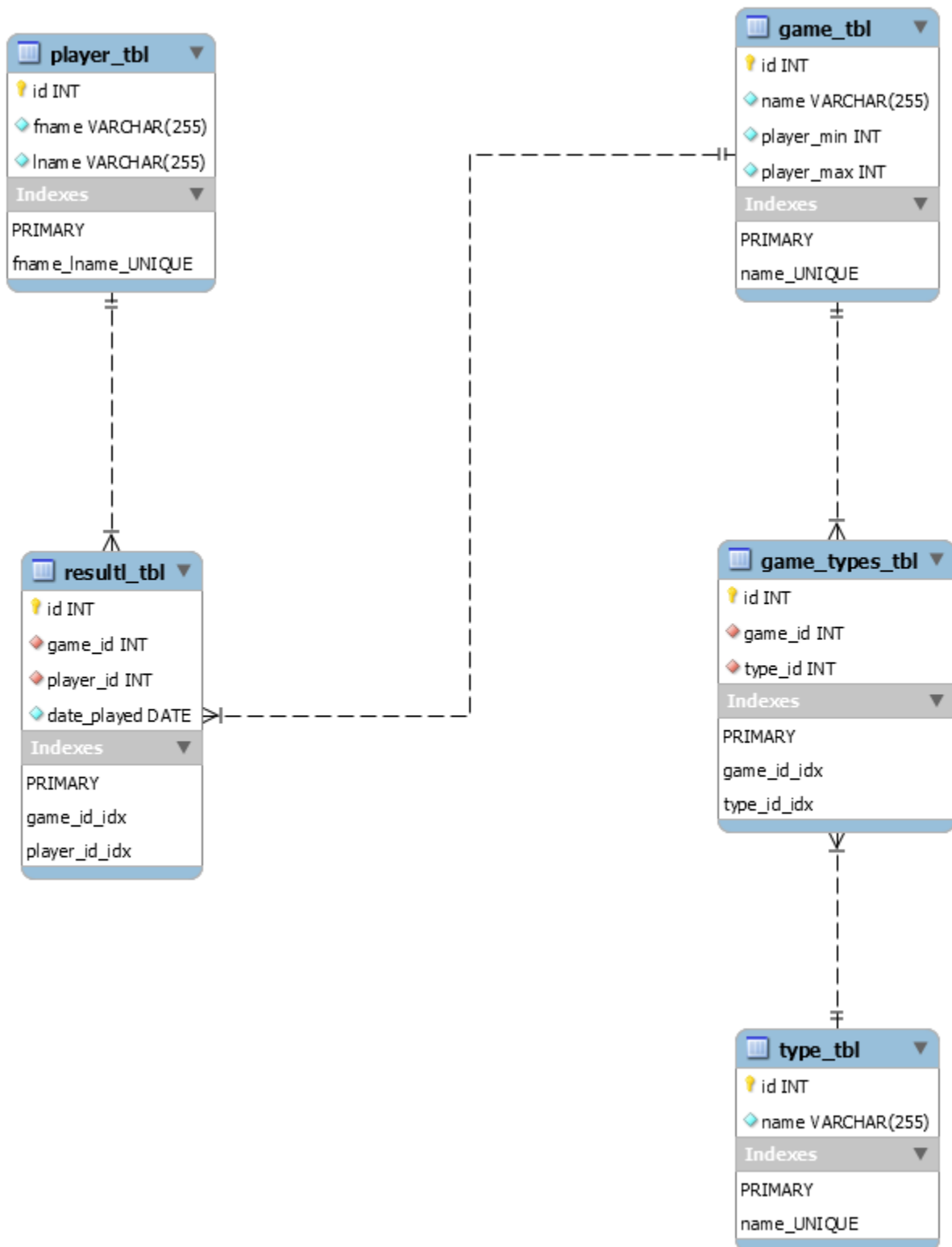- PRIMARY
- name_UNIQUE

Table Creation Queries

```
CREATE TABLE player_tbl (
        id int NOT NULL AUTO_INCREMENT,
        fname varchar(255) NOT NULL,
        lname varchar(255) NOT NULL,
        PRIMARY KEY (id),
        UNIQUE KEY (fname, lname)
) ENGINE=InnoDB;

CREATE TABLE game_tbl (
        id int NOT NULL AUTO_INCREMENT,
        name varchar(255) NOT NULL,
        player_min int NOT NULL,
        player_max int NOT NULL,
        PRIMARY KEY (id),
        UNIQUE KEY (name)
) ENGINE=InnoDB;

CREATE TABLE type_tbl (
        id int NOT NULL AUTO_INCREMENT,
        name varchar(255) NOT NULL,
        PRIMARY KEY (id),
        UNIQUE KEY (name)
) ENGINE=InnoDB;

CREATE TABLE game_types_tbl (
        id int NOT NULL AUTO_INCREMENT,
        game_id int NOT NULL,
        type_id int NOT NULL,
        PRIMARY KEY (id),
        FOREIGN KEY (game_id) REFERENCES game_tbl (id) ON DELETE CASCADE,
        FOREIGN KEY (type_id) REFERENCES type_tbl (id) ON DELETE CASCADE,
        UNIQUE KEY (game_id, type_id)
) ENGINE=InnoDB;

CREATE TABLE result_tbl (
        id int NOT NULL AUTO_INCREMENT,
        game_id int NOT NULL,
        player_id int NOT NULL,
        date_played  date NOT NULL,
        PRIMARY KEY (id),
        FOREIGN KEY (game_id) REFERENCES game_tbl (id) ON DELETE CASCADE,
        FOREIGN KEY (player_id) REFERENCES player_tbl (id) ON DELETE CASCADE
) ENGINE=InnoDB;
```

General Use Queries

For all these general use queries, from the home.php names are of games and players are displayed in drop down boxes on the forms, which have been populated with info already in the database.

 When you submit the form it will automatically send the id of the game/player you selected by name.

Add Player
INSERT INTO game_tbl(name, player_min, player_max)
        VALUES ([Game's Name],[Min # of Players Needed],[Max # number of players supported]);

Add Game
INSERT INTO player_tbl(fname, lname) VALUES ([First Name],[Last Name]);

Add Result
INSERT INTO result_tbl(game_id, player_id, date_played) VALUES ([Game ID],[Player ID],[Date Game was Played]);

Add Type
INSERT INTO type_tbl(name) VALUES ([Name of Type]);

Add Game_Type
INSERT INTO game_types_tbl(game_id, type_id) VALUES ([Game ID],[Type ID]);

Update Game
UPDATE game_tbl SET name=[new game name], player_min=[new min player], player_max=[new max player] WHERE id = [id of game updating];

Update Player
UPDATE player_tbl SET fname=[new first name], lname=[new last name] WHERE id = [id of player updating];

Delete Game
DELETE FROM game_tbl WHERE id = [id of game you are deleting];

Delete Player
DELETE FROM player_tbl WHERE id = [id of player you are deleting];

Delete Type
DELETE FROM type_tbl WHERE id = [id of type you are deleting];

Delete Game_Type
DELETE FROM game_types_tbl WHERE game_id=[id of game] AND type_id=[id of type];

View All Games
SELECT name, player_min, player_max FROM game_tbl;

View All Players
SELECT fname, lname FROM player_tbl;

View All Types
SELECT name FROM type_tbl;

View all Results
SELECT game_tbl.name, result_tbl.date_played, player_tbl.fname, player_tbl.lname FROM player_tbl
     INNER JOIN result_tbl ON result_tbl.player_id = player_tbl.id
     INNER JOIN game_tbl ON game_tbl.id = result_tbl.game_id;

View Game's Basic Info by Game ID
SELECT name, player_min, player_max FROM game_tbl
     WHERE game_tbl.id = [game id];

View Game's Types by Game ID
SELECT type_tbl.name FROM game_tbl
     INNER JOIN game_types_tbl ON game_tbl.id = game_types_tbl.game_id
     INNER JOIN type_tbl ON game_types_tbl.type_id = type_tbl.id
     WHERE game_tbl.id = [game id]

View Game's Results by Game ID
SELECT result_tbl.date_played, player_tbl.fname, player_tbl.lname FROM player_tbl
     INNER JOIN result_tbl ON result_tbl.player_id = player_tbl.id
     INNER JOIN game_tbl ON game_tbl.id = result_tbl.game_id
     WHERE result_tbl.game_id = [game_id]

View Games by Type
SELECT game_tbl.name FROM game_tbl
     INNER JOIN game_types_tbl ON game_tbl.id = game_types_tbl.game_id
     INNER JOIN type_tbl ON game_types_tbl.type_id = type_tbl.id
     WHERE type_tbl.id = [type id];

View Player by ID
SELECT fname, lname FROM player_tbl WHERE player_tbl.id = [player id];

View Results by Player
SELECT result_tbl.date_played, game_tbl.name FROM player_tbl
     INNER JOIN result_tbl ON result_tbl.player_id = player_tbl.id
     INNER JOIN game_tbl ON game_tbl.id = result_tbl.game_id
     WHERE player_tbl.id = [player id];