

Torneos de Yu-Gi-Oh

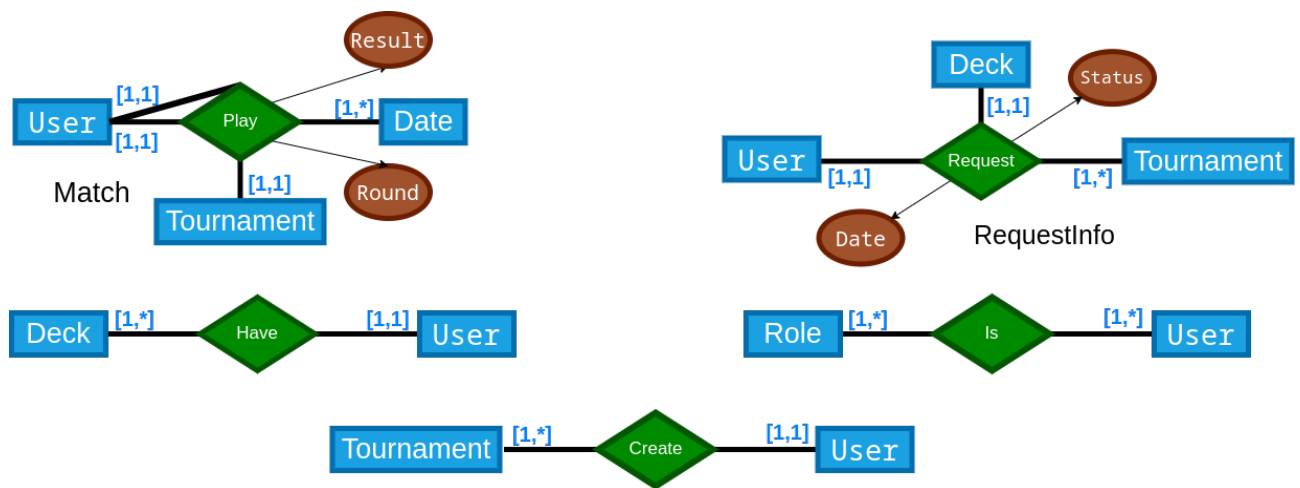
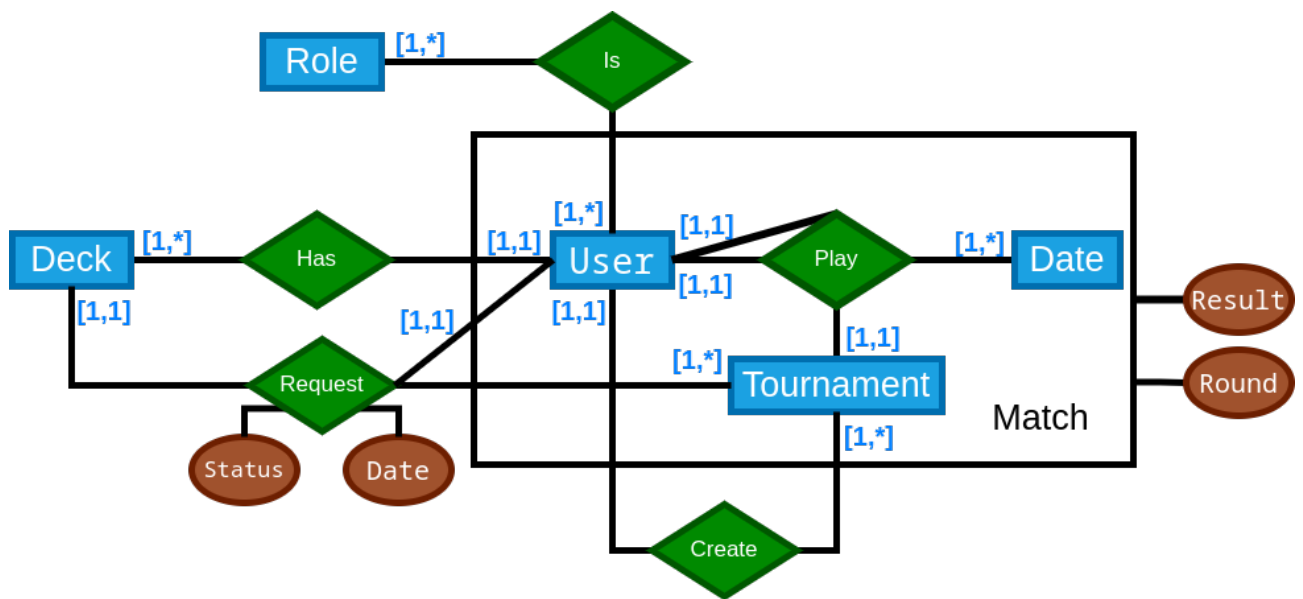
Chavely González Acosta C312
José Carlos Pendas Rodríguez C311
Lázaro David Alba Ajete C311
Max Bengochea Moré C311

December 5, 2023

1 Forma de instalar la aplicación

1. Abrir una consola
2. Colocarse en la carpeta raíz del proyecto(YuGiOh)
3. escribir "dotnet run"

2 Modelo conceptual de la base de datos:



3 Valoración del diseño de la base de datos en cuanto a si es o no correcto:

Sea el esquema relacional $R(U, F)$:

Conjunto de Atributos(U)

- UserID, user_name, township, province, phone (optional), address, DeckID, deck_name, main_deck_sz, side_deck_sz, extra_deck_sz, archetype, TournamentID, tournament_name, start_date, location, RoleID, role_name, date, result, round, status

Dependencias Funcionales(F)

UserID \rightarrow [user_name, township, province, phone (optional), address]

DeckID \rightarrow [RoleID]

DeckID \rightarrow [deck_name, main_deck_sz, side_deck_sz, extra_deck_sz, archetype]

DeckID \rightarrow [UserID]

TournamentID \rightarrow [tournament_name, start_date, location]

RoleID \rightarrow [role_name]

UserID, TournamentID \rightarrow [status, date]

UserID, TournamentID \rightarrow [DeckID]

UserID1, UserID2, TournamentID, date \rightarrow [IDUser1, IDUser2, IDTournament, Date]

UserID1, UserID2, TournamentID, date \rightarrow [Result, IDWinner, Round]

Sea $p = \{R1, R2, R3, R4, R5, R6\}$

$R1(U1, F1)$

U1: F1:

DeckID \rightarrow [deck_name, main_deck_sz, side_deck_sz, extra_deck_sz, archetype]

UserID \rightarrow [RoleID]

$R2(U2, F2)$ **U2: F2:**

DeckID \rightarrow [deck_name, main_deck_sz, side_deck_sz, extra_deck_sz, archetype]

DeckID \rightarrow [UserID]

$R3(U3, F3)$ **U3: F3:**

TournamentID \rightarrow [tournament_name, start_date, location]

$R4(U4, F4)$ **U4: F4:**

RoleID \rightarrow [role_name]

$R5(U5, F5)$ **U5: F5:**

UserID, TournamentID \rightarrow [status, date]

UserID, TournamentID \rightarrow [DeckID]

$R6(U6, F6)$ **U6: F6:**

UserID1, UserID2, TournamentID, Date \rightarrow [UserID1, UserID2, TournamentID, date]

UserID1, UserID2, TournamentID, Date \rightarrow [result, round]

Dados un esquema relacional $R(U, F)$ y una descomposición $\rho = (R1, R2, \dots, Rk)$, desde el punto de vista teórico, una base de datos relacional está correctamente diseñada o, simplemente, es un diseño correcto si se cumplen las tres propiedades siguientes:

1. Todos los esquemas relacionales $Rj(Uj, Fj)$ ($j=1, \dots, k$) de la descomposición ρ están en 3FN o una forma normal superior.

2. La descomposición ρ satisface la propiedad de join sin pérdida de información (PLJ).
3. La descomposición ρ satisface la propiedad de preservación de dependencias funcionales (PPDF).

Mediante la aplicación del algoritmo pudimos hallar una descomposición ρ que cumple con dos de las características de un diseño correcto. Ahora bien, la tercera condición (la satisfacción de PLJ) se puede comprobar fácilmente aplicando una de las consecuencias del lema que se presenta a continuación,

Consecuencia Si la llave X de $R(U, F)$ está contenida completamente en alguno de los esquemas relacionales de la descomposición, entonces, se puede afirmar directamente que ρ es un diseño correcto. En nuestro caso se cumple con las tres propiedades requeridas las llaves candidatas (IDDeck, IDTournament) o (IDUser, IDTournament) y estas están completamente contenidas en R_5 .

Play		User		Deck		Tournament	
PK,FK1	<u>TournamentID</u>	PK	<u>UserID</u>	PK	<u>DeckID</u>	PK	<u>TournamentID</u>
PK	<u>Date</u>	str	player_name	str	deck_name	str	tournament_name
PK,FK2	<u>UserRID</u>	str	township	str	archetype	str	tournament_address
PK,FK3	<u>UserLID</u>	str	province	int	main_deck_size (40-60)	date	start_date
int	round	str	player_address	int	side_deck_size (0-15)	FK	<u>UserID</u>
Enum	result	int	phone_number	int	extra_deck_size (0-15)	Request	
Is		Role		FK	<u>UserID</u>	PK,FK1	<u>UserID</u>
PK,FK1	<u>UserID</u>	PK	<u>RoleID</u>			PK,FK2	<u>TournamentID</u>
PK,FK2	<u>RoleID</u>	str	role_name			FK3	<u>DeckID</u>
						date	request_date
						Enum	status