

JEU PUISSANCE 4

Table des matières

Demande	2
Description du projet.....	2
Fonctionnalités attendues.....	3
Questions Client	3
Les contraintes sur les données	3
Dictionnaire	4
Dépendances fonctionnelles simple/composées	3
Modèle conceptuel des données (MCD)	3
Modèle logique des données (MLD)	3
Modèle Physique des données (MPD)	3
Création des tables.....	3
Création des requêtes de test	3
Procédures stockées.....	3
Déclencheurs automatiques (événement SQL)	3

Demande

Description du projet

Puissance 4 est un jeu de stratégie combinatoire abstrait.

Pour jouer au puissance 4, il vous faut :

- Le plateau du jeu et ses 42 emplacements pour jetons répartis en 6 lignes et 7 colonnes
- 42 jetons de 2 couleurs différentes
- Être 2 joueurs (ou 1joueur contre une IA)

Commencer une partie de puissance 4 : (règle de gestion)

Pour commencer une partie de puissance 4, on désigne le premier joueur.

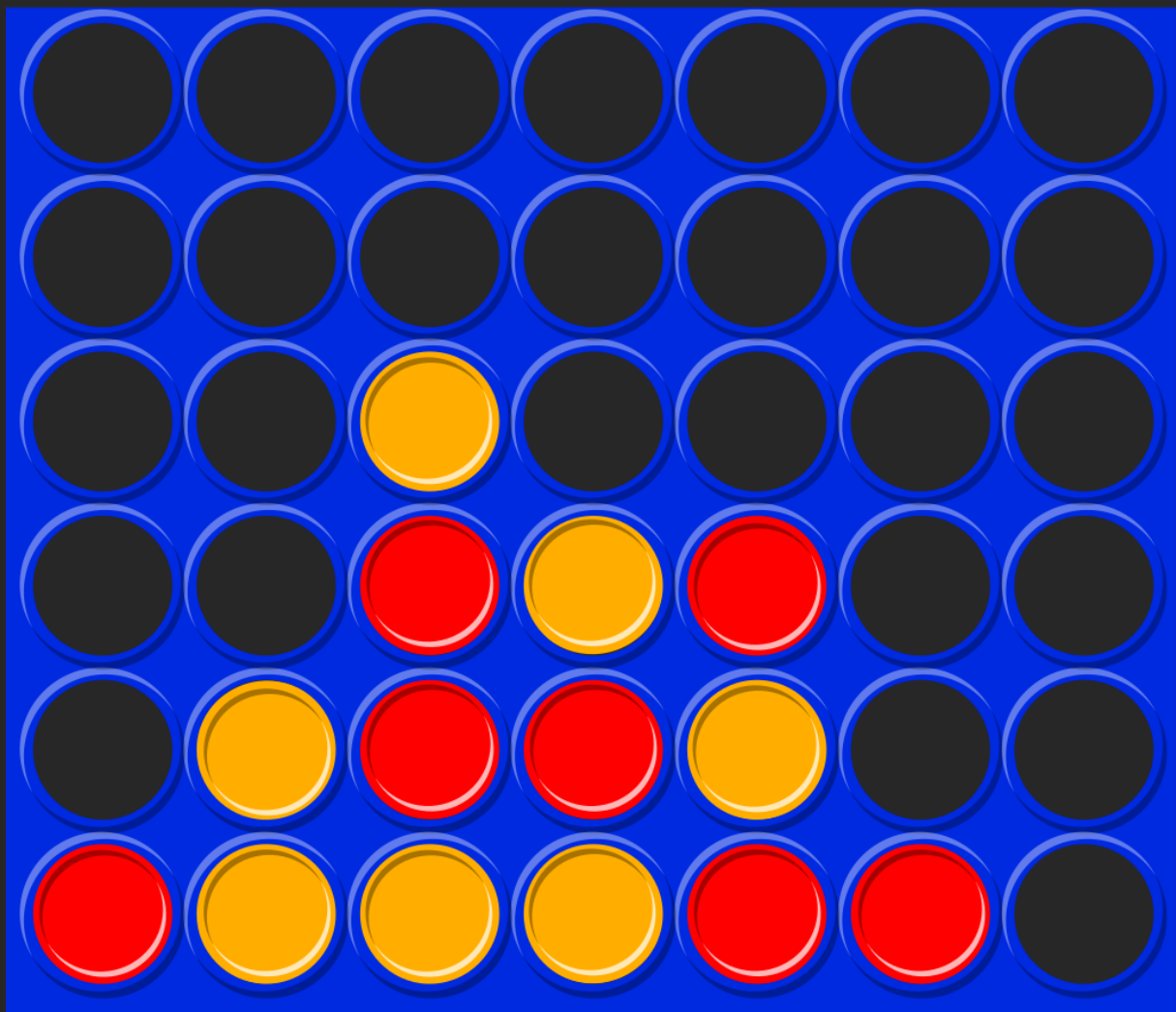
Celui-ci met un de ses jetons de couleur dans l'une des colonnes de son choix. Le jeton tombe alors en bas de la colonne.

Le deuxième joueur insère à son tour son jeton, de l'autre couleur dans la colonne de son choix. Et ainsi de suite jusqu'à obtenir une rangée de 4 jetons de même couleur.

Gagner une partie de puissance 4 :

Pour gagner une partie de puissance 4, il suffit d'être le premier à aligner 4 jetons de sa couleur horizontalement, verticalement et diagonalement.

Si lors d'une partie, tous les jetons sont joués sans qu'il y est d'alignement de jetons, la partie est déclaré nulle.



Fonctionnalités attendues

Exemple de fonctionnement du jeu en ligne :

- <https://lululataupe.com/tout-age/686-puissance-4>

Les fonctionnalités attendues sont :

Joueur

- Ajouter un joueur
- Modifier un joueur
- Supprimer un joueur
- Désactiver un joueur (sans le supprimer)
- Lister les joueurs
- Afficher les détails d'un joueur

Parties

- Un **joueur** s'identifie par son **pseudo** (pas de **mot de passe** dans un 1^{er} temps)
- Un joueur peut créer une partie (**règle de gestion**)
- Une **partie** est composée de "tours de jeu"
- Le joueur peut arrêter une partie à tout moment (et perdre)
- La partie se termine
 - o quand un joueur aligne 4 jetons et gagne
 - o quand un joueur quitte et perd
 - o quand tous les jetons sont joués sans alignement de 4 : partie nulle

Tour de Jeu

- Un tour de jeu se termine lorsque le joueur placé un jeton dans une colonne

Scores

- Les **scores** de chaque partie sont sauvegardés
- Il est possible d'afficher le % de **réussite**, % de **nul**, % de **défaite** d'un joueur

Information complémentaire

Un joueur est caractérisé en plus :

- **Email**
- Nombre de partie (**donnée calcul**)
- **Date de création** du joueur
- **Dernière connexion** du joueur
- **Si le joueur est en ligne ou pas** (type logique)
- Score totale (**donnée calcul**)

Questions Client

Le pseudo du joueur est-il égale au username ?

Non, c'est une entité à part

Veut-on aussi voire l'heure de connexion du joueur ?

Une partie gagnée par un joueur donne combien de point ?

Les contraintes sur les données

Un joueur peut avoir une et une seule information complémentaire

Une information complémentaire peut appartenir à un seul et un seul joueur

Dictionnaire

Mnémonique	Signification	Type	Longueur	Contraint
pl_pseudo	Pseudo du joueur	AN	11	identifiant
pl_username	Prénom du joueur	A	30	obligatoire
pl_password	Mot de passe du joueur	AN	15	obligatoire, doit être de maximum 15 caractère an
pl_email	Email du joueur	AN	50	obligatoire
pl_register	Date de l'enregistrement du joueur	Date		obligatoire, default date du jour
pl_lastConnect	Dernière connexion du joueur	Date & Heure		facultative, format "yy-m-d h:m:s"
score_pl_1	Score du joueur 1	N	3	facultatif
score_pl_2	Score du joueur 2	N	3	facultatif
complet_id	Identifiant pour l'association d'information complémentaire du joueur	N	11	identifiant, auto incrémentation
complet_win	Nombre de partie	N	5	obligatoire, défaut "0"
complet_null	Nombre de partie nulle	N	5	obligatoire, défaut "0"
Complet_defeat	Nombre de défaut	N	5	obligatoire, default "0"

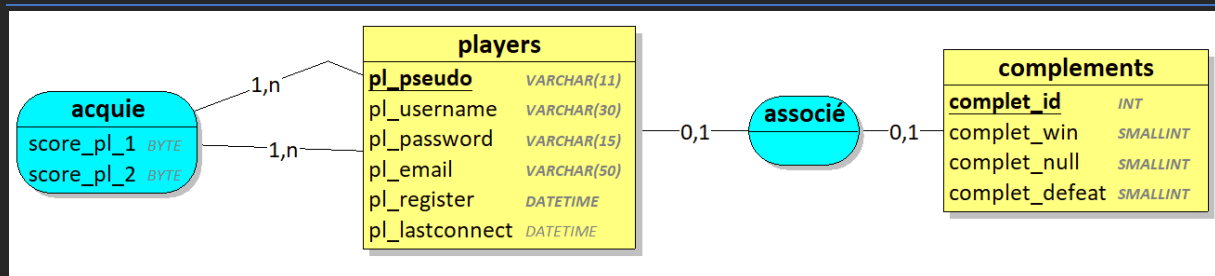
Dépendances fonctionnelles simple/composées

pl_pseudo -> pl_password, pl_email, pl_register, pl_lastConnect, complet_id

complet_id -> complet_win, complet_null, complet_defeat

acque -> pl_pseudo, pl_pseudo_1, score_pl_1, score_pl_2

Modèle conceptuel des données (MCD)



Modèle logique des données (MLD)

Players = (pl_pseudo, pl_username, pl_password, pl_email, pl_register, pl_lastconnect, #complet_id)

Complets = (complet_id, complet_win, complet_null, complet_defeat)

Acque = (#pl_pseudo, #pl_pseudo_second, score_first_pl, score_last_pl)

Modèle Physique des données (MPD)

Création des tables



puissance_4_create.s
ql

Création des requêtes de test

Procédures stockées

Déclencheurs automatiques (événement SQL)