

TP 2 - Monopoly en SQL

I Présentation du jeu Monopoly et objectifs de séance



FIGURE 1 – Plateau de jeu Monopoly

I.1 Le jeu Monopoly en SQL

Le jeu de plateau Monopoly (figure 1) est un jeu de société américain mondialement connu. Le but du jeu consiste à ruiner ses concurrents par des opérations immobilières. Il symbolise les aspects apparents et spectaculaires du capitalisme, les fortunes se faisant et se défaisant au fil des coups de dés. Il en existe de nombreuses versions, dont celle qui nous intéresse ici : acheter Paris.

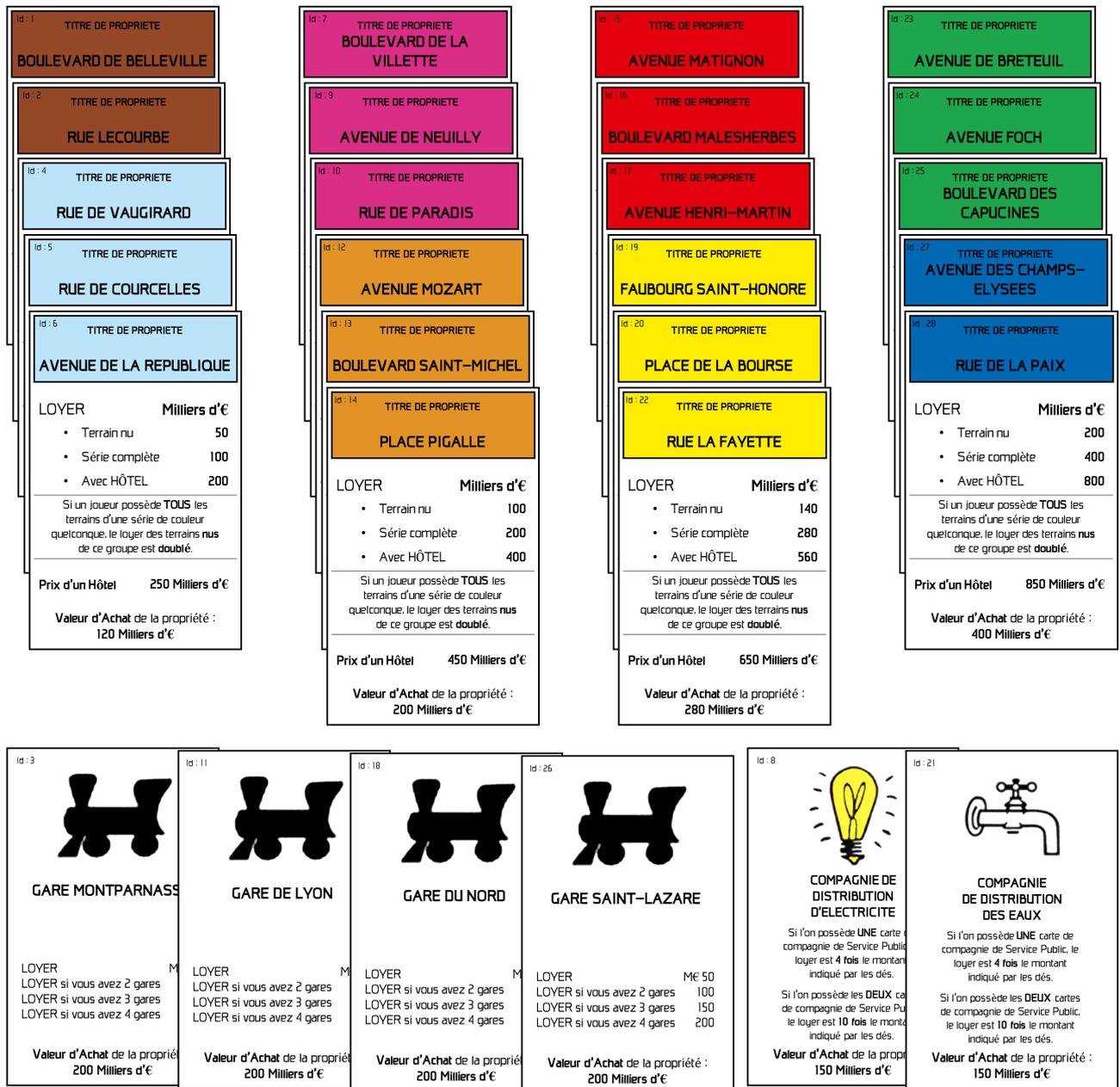


FIGURE 2 – Propriétés en jeu

Bien que les jeux de sociétés à plateau physique soient toujours largement répandus et utilisés, nombre d'entre eux possèdent aujourd'hui une version numérique qui offrent de nouvelles possibilités au joueur. Elle permet par exemple de jouer en ligne avec des amis (ou des ennemis le temps du jeu) à distance, ou encore de jouer seul contre un algorithme généré par la machine.

Dans ce TP, on se propose d'analyser une des tâches nécessaire à l'utilisation numérique du jeu de société Monopoly : **la manipulation de la base de données associée au jeu.**

Objectif

Vous allez dans ce TP :

- Analyser la structure de la base de données du Monopoly et définir les liens entre les tables.
- Réaliser des manipulations sur cette base de données associées à différentes actions que peut mener un joueur au cours de la partie.

I.2 Règles du jeu et structure des tables

On considère une partie mettant en jeu 3 joueurs (figure 3), qu'on appellera :



FIGURE 3 – Joueurs

Chaque joueur possède ou peut posséder :

- Des billets de banque qui forment son **porte-monnaie**.
- Des cartes **propriétés** (figure 2) qui peuvent être soit **des voies** (Rue Lecourbe, Boulevard Malesherbes, etc.), soit **des gares**, soit **des compagnies**. Chaque propriété possède **une couleur** et présente des améliorations lorsque le joueur possède les cartes de la même couleur.
- Des cartes **chance** (figure 4) à conserver, utilisables à tout moment de la partie.



FIGURE 4 – Cartes chance

Lorsqu'un joueur possède toutes les voies d'une même couleur, il peut (moyennant finance) y installer **des hôtels**, qui augmentent drastiquement le loyer de ces propriétés.

Règles générales du Monopoly

Lorsque c'est son tour, le joueur lance deux dés 6 faces et se déplace du nombre de cases indiqué par les dés. Suivant la case sur laquelle il arrive, il paye un loyer, peut acheter une propriété, ne rien faire ou tirer une carte chance. Les autres actions (prison, etc) ne seront pas traitées dans ce TP. Le jeu se termine quand un des joueurs est ruiné (son porte-monnaie tombe à zéro).

Les règles spécifiques seront décrites au fil de ce TP. Certaines règles ont été modifiées par rapport aux règles officielles du jeu d'origine.

I.3 Présentation des tables de la base de données Monopoly

La base de données possède 5 tables qui sont liées les unes aux autres par des clefs primaires et des clefs étrangères. On donne en annexe (chapitre III) le contenu de chaque table et figure 5 les relations entre les tables.

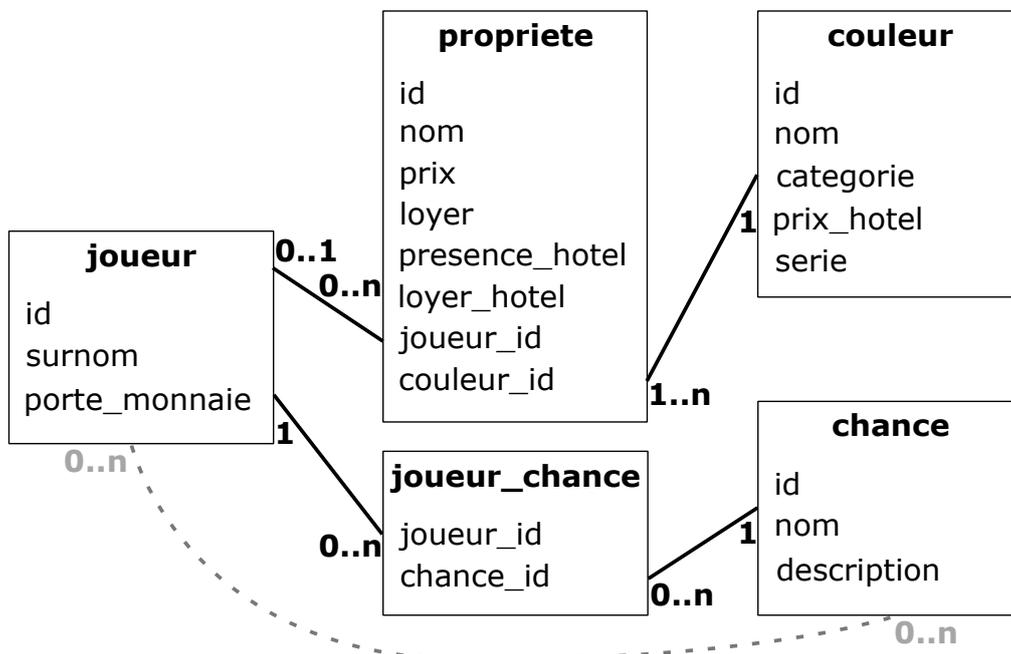


FIGURE 5 – Relation entre les tables de la base de données Monopoly

Problématique technique

Vous êtes l'algorithmiste de commande du jeu Monopoly en ligne. Quelles commandes donner à MySQL pour obtenir les informations utiles contenues dans la base de données ?

I.4 Matériel nécessaire

Vous aurez besoin, pour réaliser ce TP :

- Du SGBD (Système de Gestion de Base de Données) SQLiteStudio
- Du code SQL de la base de données 'monopoly' : "SQL_monopoly" à compiler dans votre SGBD.

II Place au jeu

II.1 Analyse du plateau au tour i

On considère la partie déjà avancée. Nous en sommes au tour i. Les trois joueurs ont des possessions diverses, certaines propriétés sont à vendre, et les porte-monnaies commencent à se vider.

Les possessions des joueurs au tour i ainsi que le plateau de jeu sont donnés figure 6



FIGURE 6 – Etat du plateau au tour i

On se propose dans cette partie d’analyser la disposition du plateau au tour i grâce à des requêtes SQL simples.

II.1.1 Analyse des propriétés

Le but de cette partie est d’avoir une vision d’ensemble des propriétés en jeu. Pour les questions suivantes, on ne réalisera que des actions sur la table ‘propriete’.

Q1. SQL - Affichez l’ensemble des éléments de la table ‘propriete’. Classez les noms dans l’ordre alphabétique.

Q2. SQL - Affichez uniquement les propriétés qui sont des compagnies, ou des gares.

Q3. SQL - A l'inverse, à l'aide de la question précédente et en utilisant l'opération ensembliste de différence (mot clef EXCEPT), affichez uniquement les propriétés qui ne sont ni des compagnies, ni des gares.

Q4. SQL - Affichez le nombre de gares.

Pour les questions suivantes, on pourra utiliser les tables 'joueur', 'propriete' et 'couleur'.

Q5. SQL - Comptez le nombre de couleur à 2, 3 ou 4 cartes par série.

Une carte propriété de catégorie 'voie' possède les informations suivantes, dans cet ordre :

id : 1	
TITRE DE PROPRIETE	
BOULEVARD DE BELLEVILLE	
LOYER	Milliers d'€
• Terrain nu	30
• Série complète	60
• Avec HÔTEL	120
Si un joueur possède TOUS les terrains d'une série de couleur quelconque, le loyer des terrains nus de ce groupe est doublé .	
Prix d'un Hôtel	150 Milliers d'€
Valeur d'Achat de la propriété : 60 Milliers d'€	

- id
- couleur
- nom
- loyer (terrain nu)
- loyer (série complète)
- loyer (avec hôtel)
- prix d'un hôtel
- prix d'achat de la propriété

FIGURE 7 – Carte propriété (catégorie 'voie')

Q6. SQL - Proposez une commande permettant d'afficher la projection de la jointure donnant l'ensemble des informations contenues sur les cartes propriétés.

A l'aide de la commande

```
CREATE VIEW carte_propriete_vw AS SELECT propriete.id, couleur.nom AS 'couleur', propriete.nom AS 'nom', loyer, loyer*2 AS 'loyer_double', loyer_hotel, couleur.prix_hotel AS 'prix_hotel', prix AS 'prix_propriete' FROM propriete JOIN couleur ON propriete.couleur_id=couleur.id;
```

vous pouvez renommer cette nouvelle table projetée, qui est en réalité **une vue**. On pourra utiliser la vue 'carte_propriete_vw' de la même manière qu'une table dans le reste du TP.

Q7. SQL - Proposez une commande permettant d'afficher l'ensemble des informations contenues sur la carte 'Boulevard de Belleville'.

SYNTHESE :

Dans cette partie, vous avez pu réaliser une jointure entre deux tables liées par des clefs primaires

et étrangères.

Q8. En vous aidant de la figure 5, quelle est la relation entre les tables 'couleur' et 'propriete' ? Et entre les tables 'joueur' et 'propriete' ? Pour les deux relations précédentes, vous préciserez les attributs qui permettent la jointure en indiquant la clef primaire et la clef étrangère.

II.1.2 Pouvoir d'achat du groupe

Imaginons que les trois joueurs s'allient et partagent leurs biens.

Q9. SQL - Affichez la somme des porte-monnaies des trois joueurs.

Q10. SQL - Affichez la somme des prix des propriétés non vendues (pour lesquelles **joueur_id IS NULL**)

Q11. En s'unissant, les joueurs peuvent-ils acheter l'ensemble des propriétés non vendues sur le plateau ?

Q12. SQL - Déterminer la somme des prix des hôtels non encore achetés

Q13. En s'unissant, les joueurs ont-ils les moyens d'acheter tous les hôtels ?

II.2 Cartes chances

Q14. SQL - Affichez dans la même projection le surnom des joueurs et les cartes chances que chacun possède.

SYNTHESE :

Dans cette partie, vous avez pu réaliser une jointure entre plusieurs tables liées par des clefs primaires et étrangères.

Q15. Quelle est la relation entre les tables joueur et chance ? A quoi sert la table de jointure joueur_chance ?

II.3 C'est parti

II.3.1 Tour i+1 - Chapeau

C'est au tour de **Chapeau** de jouer. Il est sur la case 'Gare du Nord'. Avant de lancer ses dés, il souhaite connaître les propriétés sur lesquelles il est susceptible de tomber en 1 lancé (2 dés à 6 faces).



FIGURE 8 – Possibilités de déplacement au tour i+1

Q19. SQL - Quel est le prix moyen de ces propriétés ? On pourra utiliser directement les id des propriétés de départ (Place de la Bourse : 20) et d'arrivée (avenue des Champs-Élysées : 27)

Q20. SQL - Quel loyer maximal le joueur est-il susceptible de payer (sans prendre en compte les hôtels ni les loyers doubles) et sur quelle propriété ? On donne la commande SQL pour savoir si une propriété appartient à un joueur (dans la table 'propriete') : `joueur_id IS NOT NULL`. On pourra utiliser l'id du joueur chapeau (1) directement dans la commande.

Q21. SQL - Quel loyer maximal de quelle propriété est-il susceptible de payer en prenant en compte les hôtels ?

Le joueur lance les dés. Il arrive sur la case 'avenue Foch' (id 24).

Q16. A partir de l'extrait du plateau figure 8, déterminez les noms de la première et de la dernière propriété atteignables par le joueur.

Q17. SQL - Affichez les id et les noms de la première et la dernière propriété atteignables par le joueur.

Q18. SQL - En utilisant les commandes `LIMIT` et `OFFSET` et en vous aidant du plateau figure 8, affichez les caractéristiques de l'ensemble des propriétés sur lesquelles le joueur peut tomber en un lancé des deux dés à 6 faces. Vous pourrez utiliser les id des propriétés déterminées à la question précédente.

id - 24	
TITRE DE PROPRIETE 	
AVENUE FOCH	
LOYER	Milliers d'€
• Terrain nu	150
• Série complète	300
• Avec HÔTEL	600
Si un joueur possède TOUS les terrains d'une série de couleur quelconque, le loyer des terrains nus de ce groupe est doublé .	
Prix d'un Hôtel	750 Milliers d'€
Valeur d'Achat de la propriété : 	
300 Milliers d'€	

FIGURE 9 – Carte propriété 'Avenue Foch'

Q22. SQL - Affichez la projection de la table 'propriete' donnant le nom, le prix d'achat et les trois loyers possibles de la propriété

Q23. SQL - La propriété est-elle libre ?

Q24. SQL - Si non, à quel joueur appartient la propriété ? Donnez la commande SQL permettant de le déterminer, en affichant son id, son nom et le nom de la propriété possédée (avenue Foch).

Q25. SQL - Le joueur qui possède l'avenue Foch possède-t-il d'autres propriétés de cette couleur et a-t-il placé des hôtels ? Vous pourrez directement utiliser l'id de la propriété (24) pour élaborer la commande.

Q26. En déduire le loyer que doit payer le joueur. A-t-il assez d'argent ?

Pour pouvoir payer le loyer, le joueur chapeau décide de revendre sa propriété la moins chère possible, ne possédant pas d'hôtel.

Q27. SQL - Proposez une commande permettant d'afficher la propriété qui répond à ces critères et qui permet au joueur de payer le loyer (il lui manque 50 000€). L'id du joueur chapeau est 1.

SYNTHESE :

Dans cette partie, en suivant le raisonnement du joueur Chapeau, vous avez élaboré les commandes permettant d'aller chercher des données simples et leurs agrégats dans la base de données. Ces commandes sont utiles pour le jeu : post-traitées, elles permettent à l'algorithme de jeu d'afficher à l'écran les informations demandées par le joueur au cours de sa partie.

II.3.2 Tour i+2 - Voiture

Le joueur Voiture s'intéresse à ses possessions.

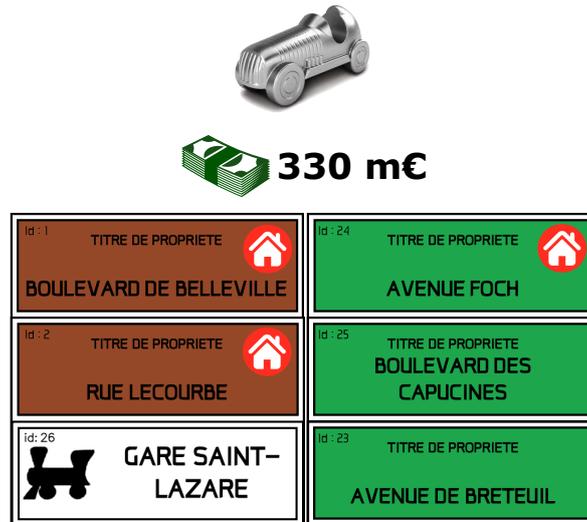


FIGURE 10 – Possessions du joueur Voiture au début de son tour

- Q28.** SQL - Afficher dans une table le surnom du joueur, son porte-monnaie, le nom et la couleur des propriétés qu'il possède ainsi que la présence ou non d'un hôtel sur ces dernières.
- Q29.** SQL - Quelles propriétés est capable de s'offrir le joueur (libres ou déjà vendues) ?
- Q30.** SQL - Quelles propriétés libres (commande : joueur_id IS NULL) est capable de s'offrir le joueur ? Classez-les par ordre de prix décroissant.
- Q31.** SQL - Donner le patrimoine financier (porte-monnaie + prix des propriétés + prix des hôtels) du joueur.

Le joueur Voiture tire les dés. 4 : pas de chance, il tombe sur la case 'Gare de Lyon'.



FIGURE 11 – Carte propriété 'Gare de Lyon'

- Q32.** SQL - Affichez le prix du loyer que doit payer le joueur et à qui il doit le payer. Le loyer d'une

gare est proportionnel au nombre de gares possédées par le propriétaire.

II.3.3 Tour i+3 - Toutou

C'est au tour du joueur Toutou de jouer (id 3). Il souhaite tomber sur une propriété libre qui lui permettrait de compléter une série.



FIGURE 12 – Possessions du joueur Toutou au début de son tour

Q33. SQL - Affichez le nombre de propriétés par couleur possédées par le joueur (id_joueur=3) et le nom de la couleur associée.

Q34. SQL - Rajoutez un filtrage d'agrégats HAVING permettant de ne garder que les séries que le joueur peut compléter en un achat.

Q35. SQL - En utilisant les résultats de la question précédente, affichez les propriétés disponibles à l'achat, permettant au joueur d'avoir une série complète en un achat.

Le joueur Toutou lance les dés : 2. Il va sur la case 'boulevard malesherbes'. Il achète la propriété et souhaite poser un maximum d'hôtels sur l'ensemble de ses séries complètes avec son budget de 810 000€



FIGURE 13 – Possessions du joueur Toutou au milieu de son tour

Q36. SQL - Affichez l'ensemble des informations des tables jointes 'propriete' et 'couleur' pour les 6 propriétés roses et rouges. Ordonnez-les par prix d'hôtel croissant.

Q37. SQL - Affichez le prix d'hôtel le plus bas de ces propriétés. Combien d'hôtels le joueur peut-il acheter au maximum et pourquoi ?

Q38. SQL - Affichez l'ensemble des combinaisons de propriétés sur lesquelles le joueur peut acheter des hôtels à hauteur de son budget de 810 000€. On effectuera une auto-jointure sur la vue 'carte_propriete_vw' permettant d'afficher l'ensemble des doublons possibles.

SYNTHESE :

Dans cette partie, vous avez pu manipuler les agrégats de données pour les utiliser comme conditions de recherche. Vous avez aussi pu voir l'utilité de l'auto-jointure pour regarder les combinaisons possibles entre les données d'une même table, respectant les conditions données.

III Annexes - détail des tables Monopoly

propriete		
Nom colonne	Type colonne	Description
<i>id</i>	INTEGER PRIMARY KEY	Clef primaire de la carte propriété
<i>nom</i>	VARCHAR(100) UNIQUE	Nom de la propriété
<i>prix</i>	INT	Prix d'achat de la propriété
<i>loyer</i>	INT	Loyer de la propriété pour un terrain nu (sans hôtel)
<i>presence_hotel</i>	BOOLEAN	Indique la présence d'un hôtel ou non sur la propriété (Nul pour les cartes de type 'gare' ou 'compagnie')
<i>loyer_hotel</i>	INT	Loyer de la propriété avec hôtel
<i>joueur_id</i>	INTEGER FOREIGN KEY	Indique l'id du joueur propriétaire
<i>couleur_id</i>	INTEGER FOREIGN KEY	Indique la couleur de la propriété

couleur		
Nom colonne	Type colonne	Description
<i>id</i>	INTEGER PRIMARY KEY	Clef primaire de la couleur des propriétés
<i>nom</i>	VARCHAR(30) UNIQUE	Nom de la couleur
<i>categorie</i>	VARCHAR(30)	Type de la propriété : voie, gare ou compagnie
<i>prix_hotel</i>	INT	Prix d'achat d'un hôtel pour la couleur, en Milliers d'€
<i>serie</i>	INT(1)	Nombre de cartes propriétés disponibles dans la couleur (2, 3 ou 4)

joueur		
Nom colonne	Type colonne	Description
<i>id</i>	INTEGER PRIMARY KEY	Clef primaire du joueur
<i>surnom</i>	VARCHAR(20) UNIQUE	Surnom du joueur
<i>porte_monnaie</i>	INT	Porte-monnaie du joueur, en Milliers d'€

chance		
Nom colonne	Type colonne	Description
<i>id</i>	INTEGER PRIMARY KEY	Clef primaire de la carte chance
<i>nom</i>	VARCHAR(100) UNIQUE	Nom de la carte chance
<i>description</i>	VARCHAR	Description de l'action permise par la carte

joueur_chance (table de liaison)		
Nom colonne	Type colonne	Description
<i>Joueur_id</i>	INTEGER FOREIGN KEY	Indique l'id du joueur propriétaire de la carte chance
<i>chance_id</i>	INTEGER FOREIGN KEY	Indique l'id de la carte chance possédée par le joueur