14/12/2022



**Rapport Projet Base de données**

L2-INFO

Réalisée par :

* Abdullah SWAILEM
* Lina BOUNOUAR
* Emad BA GUBAIR
* Ismael MATEYA

Notre projet porte sur le jeu Grand Theft Auto 5 ou communément appelé «GTA5».

Il s’agit en bref d’un jeu sorti en 2013 par Rockstar Games dans lequel on retrouve 3 personnages qui, dans la ville de Los Santos (ville fictif qui représente à la base Los Angeles), vont s’apprendre à se connaître et vont réaliser la plupart du temps des missions tel que des braquages, vol, trafic, meurtres, etc.

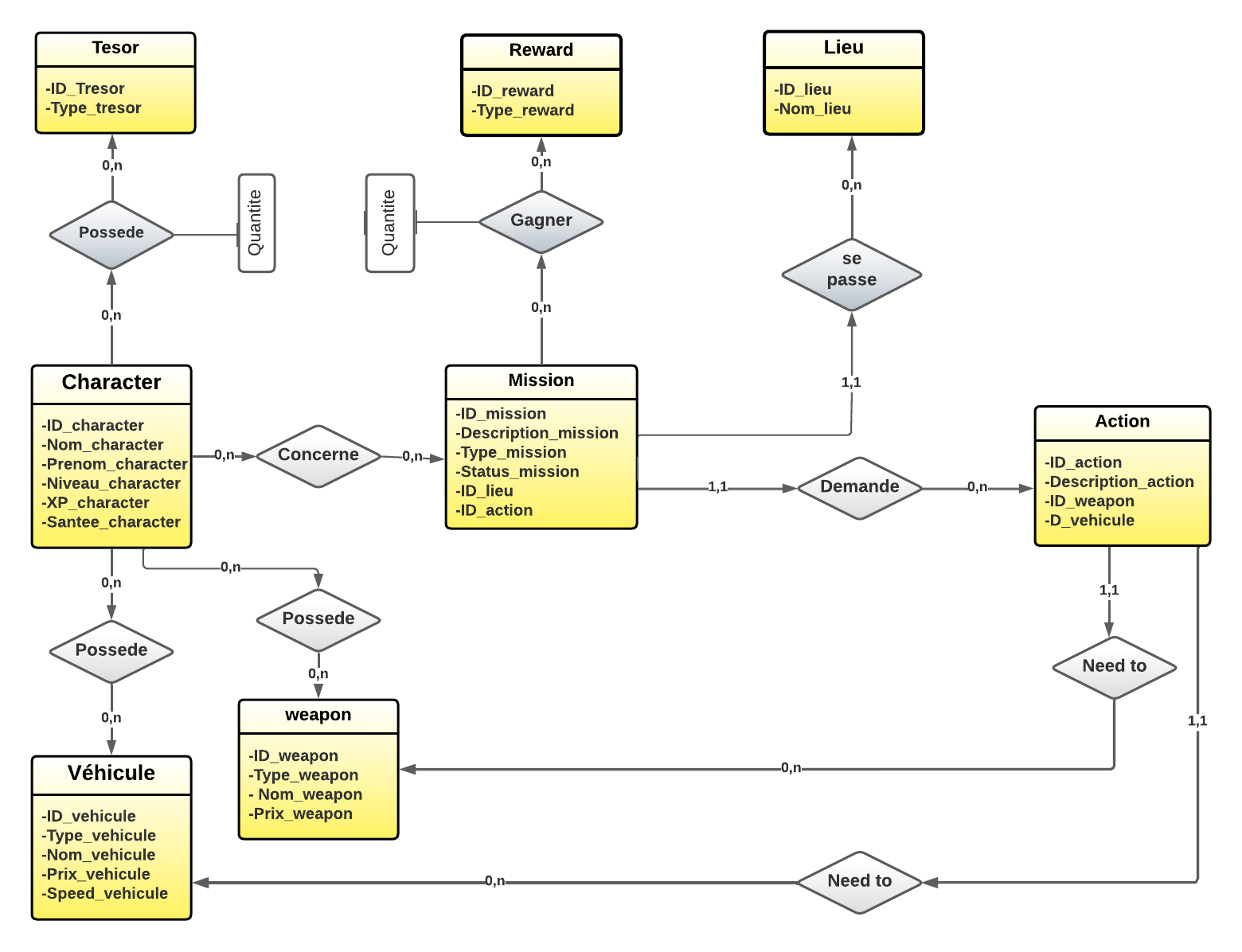
En revanche, ce qui nous intéresse c’est la représentation de ce jeu sous forme de base de données. Notre objectif principal a été de faire vivre nos bases données dans lequel plusieurs scénarios sont possibles. Par exemple, un personnage peut finir millionnaire après avoir réussi avec succès plusieurs missions ou tout simplement mourir en cours de mission.

**Comment avons-nous procéder à cela ?**

On a tout d’abord commencé par le Modèle Entité Association.

Ensuite, on s’est répartit les tâches et on s’est entraidés.

**Voici le modèle relationnel du MEA avec mise à jour :**



**Explication de quelque table:**

**Table Character :** Les personnages sont identifiés par un numéro d’identification, pour chaque personnage on connait son nom, prénom, son niveau, son état de santé, son expérience, un personnage peut posséder zéro ou plusieurs armes et véhicules.

**Table véhicule :** Les véhicules sont identifiés par un numéro d’identification, pour chaque véhicule on connait sa marque, son prix et sa vitesse. Le véhicule peut être conduit par plusieurs personnes.

**Table Weapon** : Les armes sont identifiées par un numéro, pour chaque arme on connaît son nom, sa marque et son prix. Chaque arme peut être utilisée par plusieurs personnes.

**Table Mission** : Les missions sont identifiées par un numéro, pour chaque mission on connait le type mission, son statut, son lieu , sa description et son action.

**Table Action :** : : Les action sont identifiées par un numéro, pour chaque action on connait son description , son weapon et véhicule nécessaire .

**Relation avec spécification en langage naturel :**

* **Character** (ID\_character, Nom\_character, Prenom\_character, Niveau\_character, XP\_character, Sante\_character) ;

/\* < i, n, p, N, x, s > ϵ character **⇔** le character identifier par son numéro i, son nom n et son prénom p, son niveau N, son niveau d’expérience x et par son état de santé s\*/

* **Vehicule** (ID\_vehicule, Type\_vehicule, Nom\_vehicule, Prix\_vehicule, Speed\_vehicule)

/\* < i, t, n, p, v > ϵ Vehicule  **⇔** le vehicule est identifier par son numéro i, type de vehicule t, le nom de vehicule n, son prix p et sa vitesse v \*/

* **Weapon** (ID\_weapon, Type\_weapon, Nom\_weapon, Prix\_weapon)

/\* < i, t, n, p > ϵ weapon  **⇔** le weapon est identifier par son numéro i, type de weapon t, son nom n, et par son prix p \*/

* **Tresor** (ID\_tresor, Type\_tresor)

/\* < i, t > ϵ Tresor  **⇔** le trésor est identifier par son numéro i, et son type t \*/

* **Mission** (ID\_mission, Description\_mission, Type\_mission, Status\_mission, # ID\_lieu, #ID\_action)

/\* < i, d, t, s,l,a > ϵ Mission  **⇔** lamission est identifier par un numéro i, description de la mission d, le type de la mission t, son statuts s , le lieu de la mission l , et l’action de la mission a \*/

* **Reward** (ID\_reward, Type\_reward)

/\* < i, t > ϵ Reward  **⇔** le reward est identifier par un numéro i, et par son type t\*/

* **Action** (ID\_action, Description\_action,# ID\_weapon ,#ID\_vehicule)

/\* < i, d,w,v > ϵ Action  **⇔** l’action est identifier par un numéro i, par sa description d , son weapon w et son véhicule v\*/

* **Lieu** (ID\_lieu, Nom\_ lieu)

/\* < i, n > ϵ Lieu  **⇔** le lieu est identifier par un numéro i, et un nom n\*/

* Possede\_v (#ID\_character, # ID\_vehicule)
* Possede\_w (#ID\_character, # ID\_weapon)
* Possede\_t (#ID\_character, # type\_tresor, Quantite)
* Gagner (#ID\_mission, #ID\_reward, Quantite)
* Concerne (#ID\_character, #ID\_mission)
* Char\_concerne\_miss (#ID\_character, #ID\_mission)

**Contraintes de domaine :**

* **CREATE DOMAIN dom\_type\_mission AS VARCHAR(10) CHECK (VALUE IN ('légal','illégal'));**

Il s’agit du type de la mission, une mission est soit légal ou illégal. Par exemple, faire une mission où le personnage est un chauffeur de taxi est une mission légale. Une mission où on devra tuer une personne est une mission illégale.

* **CREATE DOMAIN dom\_status\_mission AS VARCHAR(10) CHECK (VALUE IN ('ouvert','fermé','complet'));**

Une mission est au départ ouverte, les autres missions sont fermées. Alors, lorsque la mission ouverte a été complétée, la mission suivante est ouverte (autrement dit disponible) et ainsi de suite.

* **CREATE DOMAIN dom\_santé AS INT CHECK (VALUE BETWEEN 1 AND 100);**

Ceci définit la santé du personnage. La santé du personnage commence à 100 et en fonction des dégâts que le personnage reçoit (chute, explosion, tirs reçus, etc.), la santé diminue voire chuter à 0 brusquement (dans ce cas, le personnage meurt). En revanche, il est impossible d’augmenter la vie du personnage dans notre base de données, alors le joueur n’a qu’à bien se tenir.

* **CREATE DOMAIN dom\_type\_vehicule AS VARCHAR(20) CHECK (VALUE IN(**

**'Boats', 'Cars', 'Helicopters', 'Motorcycles', 'Emergency', 'Sports', 'Cycles', 'Vans'));**

C’est une des raisons du succès de GTA 5, on peut trouver tous types de véhicules que ça soit du vélo de sport à l’hélicoptère d’attaque, le joueur a toute une panoplie à sa disposition pour lors des missions ou encore s’amuser avec dans la ville de Los Santos.

* **CREATE DOMAIN dom\_type\_weapon AS VARCHAR(15) CHECK (VALUE IN ('hand','melee\_weapon','thrown\_weapon','pistol', 'shotgun', 'machine\_gun', 'assault\_rifle',' heavy\_weapon','sniper\_rifle'));**

De même pour les véhicules, le joueur a aussi plusieurs types d’armes disponibles que ça soit une arme blanche jusqu’aux fusils de précision.

**Dictionnaire des données :**

1. ID\_character

2. Nom\_character

3. Prenom\_character

4. Niveau\_character

5. XP\_character

6. Sante\_character

7. ID\_vehicule

8. Type\_vehicule

9. Nom\_vehicule

10. Prix\_vehicule

11. Speed\_vehicule

12. ID\_weapon

13. Type\_weapon

14. Prix\_weapon

15. Nom\_weapon

16. ID\_tresor

17. Type\_tresor

18. ID\_Mission

19. Description\_mission

20. Type\_mission

21. Status\_mission

22. ID\_lieu

23. Nom\_lieu

24. ID\_action

25. Description\_action

26. ID\_reward

27. Type\_reward

28. Quantite

**Contraintes d’intégrité référentielle :**

CREATE TABLE possède\_v(id\_character INT REFERENCES character(id\_character),id\_vehicule INT REFERENCES vehicule(id\_vehicule),PRIMARY KEY(id\_character,id\_vehicule));

Nous n’avons pas fait de contraintes car nous avons déjà écrit directement celui-ci, comme l’indique l’exemple ci-dessus.

**Types de données :**

DOMAIN (dom\_type\_mission) = DOMAIN (dom\_status\_mission) = DOMAIN (dom\_type\_vehicule) = DOMAIN (dom\_type\_weapon) = **chaîne de caractères.**

DOMAIN (dom\_santé) = **Type entier entre 1 et 100.**

**Les triggers et les foncions :**

1. **Trigger :**CREATE TRIGGER acheter\_weapon  
       BEFORE INSERT OR UPDATE ON possède\_w  
       FOR EACH ROW EXECUTE FUNCTION acheter\_weapon();

Ce trigger permet d,achter un weapon après il vérifie que la personne ait assez d’argent.

1. **Trigger :**  CREATE TRIGGER acheter\_vehicule

BEFORE INSERT OR UPDATE ON possède\_v

FOR EACH ROW

EXECUTE FUNCTION acheter\_vehicule()

Ce trigger permet d,achter un véhicule après il vérifie que la personne ait assez d’argent.

1. **Trigger :**  CREATE TRIGGER gagner\_reward

AFTER INSERT OR UPDATE ON char\_concerne\_miss

FOR EACH ROW

EXECUTE FUNCTION gagner\_reward()

Ce trigger permet de donner la personne les rewards après il fini la mission.

4.**Trigger :**  TRIGGER check\_mission\_ouvert

BEFORE INSERT OR UPDATE ON char\_concerne\_miss

FOR EACH ROW

EXECUTE FUNCTION check\_mission\_ouvert();

Ce trigger vérifie si la personne peux faire cette mission.

5.**Trigger** CREATE TRIGGER check\_mission\_action\_vehicule

BEFORE INSERT OR UPDATE ON char\_concerne\_miss

FOR EACH ROW

EXECUTE FUNCTION check\_mission\_action\_vehicule();

Ce trigger vérifie que l’acion a besoin de quel vehicule.

6.**Trigger** CREATE TRIGGER check\_mission\_action\_ weapon

BEFORE INSERT OR UPDATE ON char\_concerne\_miss

FOR EACH ROW

EXECUTE FUNCTION check\_mission\_action\_ weapon ();

Ce trigger vérifie que l’acion ait besoin de quel weapon.

7.**Trigger** CREATE TRIGGER change\_status\_mission

AFTER INSERT OR UPDATE ON char\_concerne\_miss

FOR EACH ROW

EXECUTE FUNCTION change\_status\_mission()

Ce trigger change l’etat de la mision.

8.**Fonction** CREATE FUNCTION echager\_or\_to\_argent(VARCHAR,int)

Cette fonction permet convertir l’or en argent (chaque lingot d’or = 100000 $ ).

9.**Fonction** CREATE FUNCTION do\_m(VARCHAR,INT)

Cette fonction prend le nom de la personnage er ID de la mission ,après elle va faire

tous les suites de travaill.