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 ?

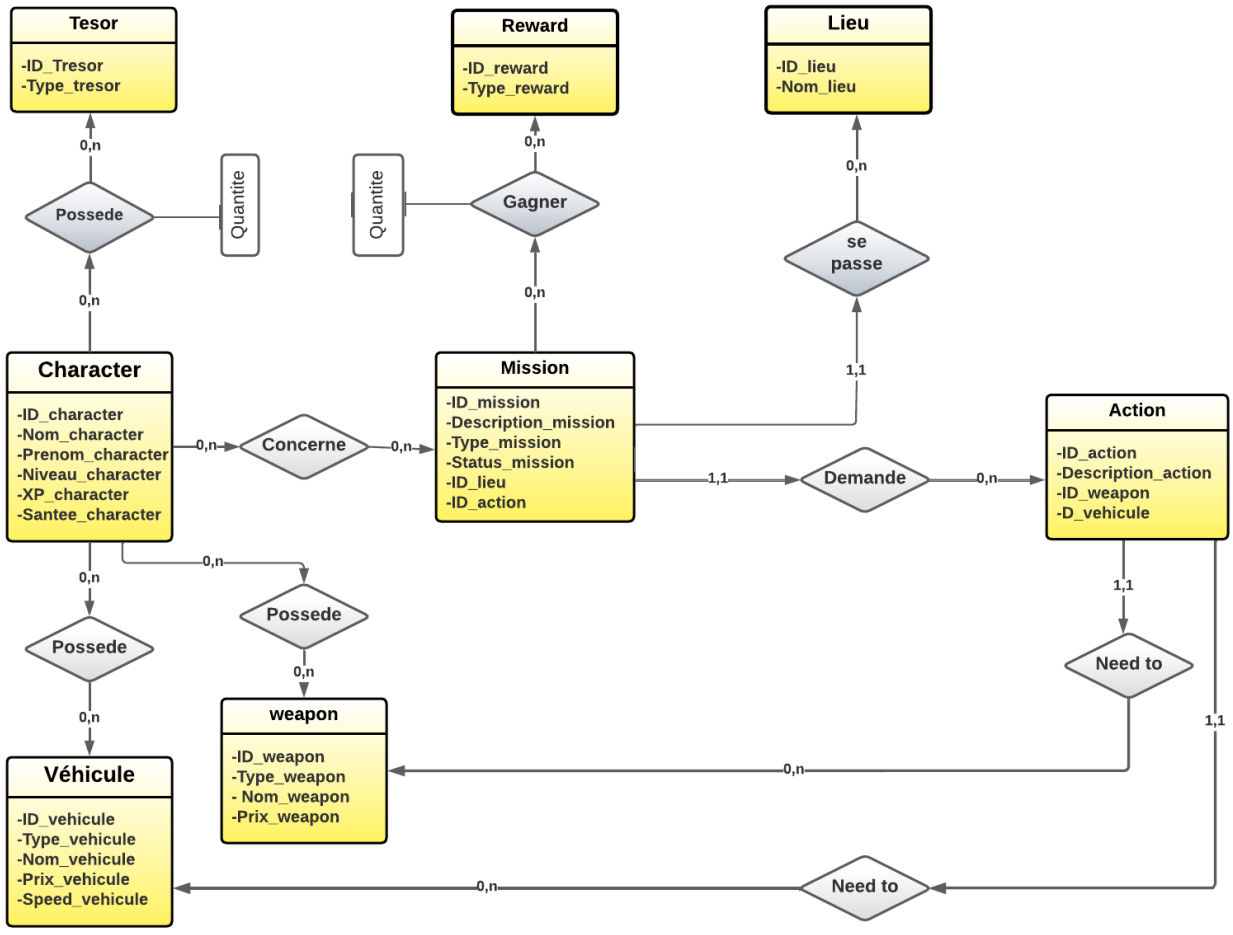
On a tout d'abord commencé par le Modèle Entité Association.
Ensuite, on s'est répartit les tâches et on s'est entraidés.

<u>Voici le modèle relationnel du MEA avec mise à jour :</u>

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 ⇔ la mission 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.
2. **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.

3. **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.