

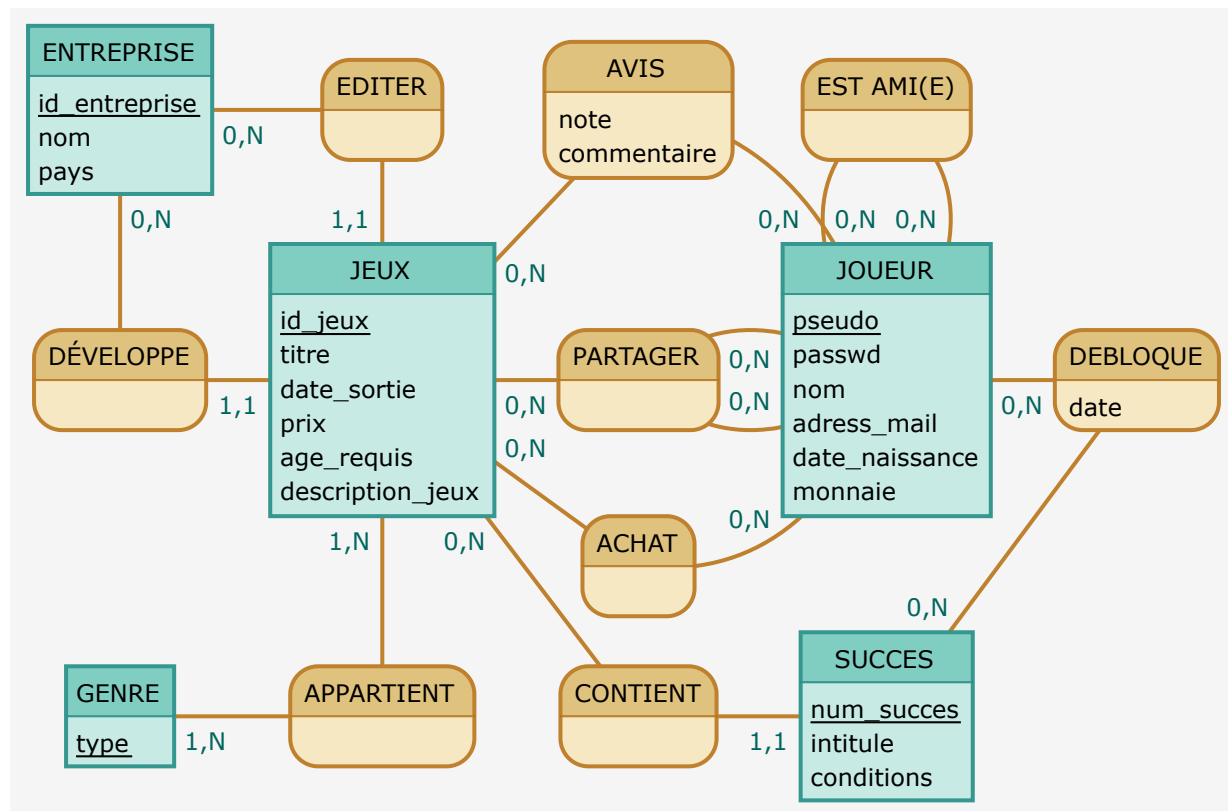
Projet : Nuage

Présentation

Le client souhaite construire une plateforme de vente de jeux vidéos en ligne. Baptisée **Nuage**, cette plateforme permettra aux joueurs d'acheter des jeux, de suivre leur progression, et de partager avec les autres joueurs leurs avis et commentaires sur les jeux auxquels ils ont joué.

Partie 1 : Modélisation

Modéliser sous la forme d'un schéma entité-association une base de données répondant aux attentes du cahier des charges du client. Ci-dessous, les différentes entités, leurs attributs et les associations :



Hypothèse

Nous avons eu comme idée de répartir les succès par type.

Contraintes non garanties par le schéma entité-association :

Un joueur ne peut pas partager un même jeu avec plusieurs amis simultanément.

Compléments apportés directement par le client

Il n'était pas précisé dans le cahier des charges si un joueur ne pouvait laisser qu'une note (Seuls les commentaires avait eu le droit à cette précision). Notre client nous a donc confirmé qu'un joueur ne peut laisser qu'une note par jeux.

Partie 2 : Implémentation

- Traduction du schéma entité-association vers le modèle relationnel.

ENTREPRISE (id_entreprise, nom, pays)

AVIS (pseudo, id_jeux, note, commentaire)

EST AMI(E) (pseudo_joueur_1, pseudo_joueur_2)

JEUX (id_jeux, titre, date_sortie, prix, age_requis, description_jeux, id_editeur, id_developpeur)

PARTAGER (pseudo_joueur_1, id_jeux, pseudo_joueur_2)

JOUEUR (pseudo, passwd, nom, adress_mail, date_naissance, monnaie)

DEBLOQUE (num_succes, pseudo, date)

ACHAT (id_jeux, pseudo)

APPARTIENT (id_jeux, type)

SUCCES (num_succes, intitule, conditions, id_jeux)

FK :

---> **AVIS**(pseudo) fait référence à **JOUEUR**(pseudo)

---> **AVIS**(id_jeux) fait référence à **JEUX**(id_jeux)

---> **EST AMI(E)**(pseudo_joueur_1) fait référence à **JOUEUR**(pseudo)

---> **EST AMI(E)**(pseudo_joueur_2) fait référence à **JOUEUR**(pseudo)

---> **JEUX**(id_editeur) fait référence à **ENTREPRISE**(id_entreprise)

---> **JEUX**(id_developpeur) fait référence à **ENTREPRISE**(id_entreprise)

---> **PARTAGER**(pseudo_joueur_1) fait référence à **JOUEUR**(pseudo)

---> **PARTAGER**(pseudo_joueur_2) fait référence à **JOUEUR**(pseudo)

---> **PARTAGER**(id_jeux) fait référence à **JEUX**(id_jeux)

---> **DEBLOQUE**(num_succes) fait référence à **SUCCES**(num_succes)

---> **DEBLOQUE**(pseudo) fait référence à **JOUEUR**(pseudo)

---> **ACHAT** (id_jeux) fait référence à **JEUX**(id_jeux)

- > **ACHAT** (*pseudo*) fait référence à **JOUEUR**(*pseudo*)
- > **APPARTIENT**(*id_jeux*) fait référence à **JEUX**(*id_jeux*)
- > **SUCCE**(*id_jeux_*) fait référence à **JEUX**(*id_jeux*)
- Implémentation du modèle relationnel sous PostgreSQL.

```
-- Create necessary tables

-- Entreprise(id_entreprise, nom, pays)
CREATE TABLE Entreprise(
    id_entreprise serial primary key,
    nom varchar(255),
    pays varchar(100)
);

-- Jeux(id_jeux, titre, date_sortie, prix, age_requis, description_jeux, id_editeur,
id_developpeur)
CREATE TABLE Jeux(
    id_jeux serial primary key,
    titre varchar(255),
    date_sortie date,
    prix int,
    age_requis int,
    description_jeux text,
    id_editeur int references Entreprise(id_entreprise),
    id_developpeur int references Entreprise(id_entreprise)
);
-- Ne connaissant pas la taille moyenne d'une description d'un jeu vidéo,
-- nous avons choisit le type "text" pour le champ "description_jeux" permettant de
réservé de la place dynamiquement.
-- De plus cela permet de ne pas réserver de l'espace en trop.

-- Joueur(pseudo, passwd, nom, adress_mail, date_naissance, monnaie)
CREATE TABLE Joueur(
    pseudo varchar(100) primary key,
    passwd varchar(255) not null,
    nom varchar(100),
    adress_mail varchar(100) not null,
    date_naissance date not null,
    monnaie int default 0
);

-- Succes(num_succes, intitule, conditions, type_succes)
CREATE TABLE Succes(
    num_succes serial primary key,
    intitule varchar(255),
    conditions text,
    id_jeux int references Jeux(id_jeux)
);

-- Appartient (id_jeux, type_genre)
CREATE TABLE Appartient(
    id_jeux int references Jeux(id_jeux),
    type_genre varchar(100),
```

```

        primary key(id_jeux, type_genre)
    );

CREATE TABLE Est_ami_e(
    pseudo_joueur_1 varchar(100) references Joueur(pseudo),
    pseudo_joueur_2 varchar(100) references Joueur(pseudo),
    primary key(pseudo_joueur_1, pseudo_joueur_2)
);

-- Achat(id_jeux, pseudo)
CREATE TABLE Achat(
    id_jeux int references Jeux(id_jeux),
    pseudo varchar(100) references Joueur(pseudo),
    primary key(id_jeux, pseudo)
);

CREATE TABLE Avis (
    pseudo VARCHAR(100) references Joueur(pseudo),
    id_jeux INTEGER references Jeux(id_jeux),
    note INTEGER,
    commentaire VARCHAR(500),
    primary key(pseudo, id_jeux)
);

-- Partager(id_jeux, pseudo_joueur_1, pseudo_joueur_2)
CREATE TABLE Partager(
    id_jeux int references Jeux(id_jeux),
    pseudo_joueur_1 varchar(100) references Joueur(pseudo),
    pseudo_joueur_2 varchar(100) references Joueur(pseudo),
    primary key(id_jeux, pseudo_joueur_1, pseudo_joueur_2),
    constraint uniquePartage unique(id_jeux, pseudo_joueur_1)
);

-- TODO: Test à faire

-- Debloque(num_succes, pseudo, date_deblocage)
CREATE TABLE Debloque(
    num_succes int references Succes(num_succes),
    pseudo varchar(100) references Joueur(pseudo),
    date_deblocage date not null,
    primary key(num_succes, pseudo)
);

```

Partie 3 : Maquette du site web

Page de connexion

LOGO

Boutique


Communautés

Connexion

CONNEXION

CONNEXION

[Mot de passe oublié ?](#)
[Créer un compte](#)



Requête

```
select passwd
from joueur
where pseudo='john';
```

Vérifie que le pseudo de l'utilisateur existe et correspond au mot de passe.

Page d'inscription

LOGO

Boutique

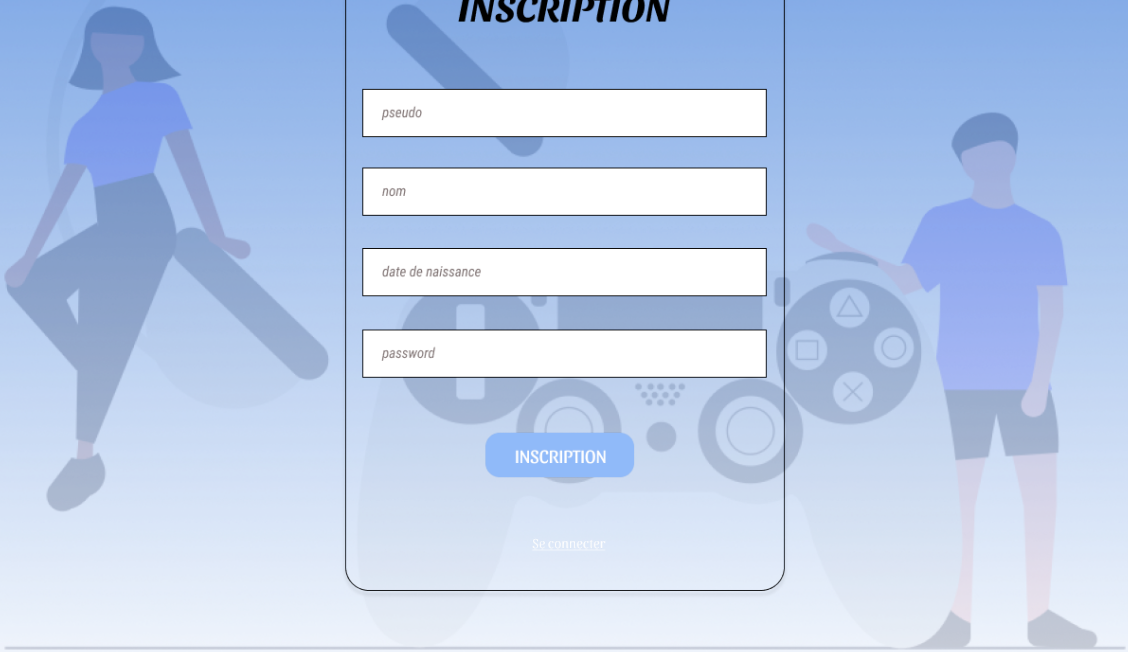
Communautés

Connexion

INSCRIPTION

INSCRIPTION

Se connecter



Requête

```
insert into Joueur(pseudo, passwd, nom, adress_mail, date_naissance)
values ('john', 'xtvfn$.dfq', 'Doe', 'john@nuage.com', '1990-01-01');
```

Crée un nouvel utilisateur.


Page de boutique

LOGO

Boutique


Communautés

Support

kenedy4Obama 

recherher un jeu go


Trier : [Date](#) [Ventes](#) [Note](#)



Battle Field 2042

Lorem ipsum is a good game in where a soldier have a mission to kill all her enemies ipsum ...


4,2 ★



Fortnite

Lorem ipsum is a good game in where a soldier have a mission to kill all her enemies ipsum ...


4,5 ★



GTA 5

Lorem ipsum is a good game in where a soldier have a mission to kill all her enemies ipsum ...


4,8 ★



CyberPunk

Lorem ipsum is a good game in where a soldier have a mission to kill all her enemies ipsum ...


3,9 ★



Valorant

Lorem ipsum is a good game in where a soldier have a mission to kill all her enemies ipsum ...


4,2 ★



Call of Duty: Modern War

Lorem ipsum is a good game in where a soldier have a mission to kill all her enemies ipsum ...


4,2 ★



Call of Duty: Vanguard

Lorem ipsum is a good game in where a soldier have a mission to kill all her enemies ipsum ...

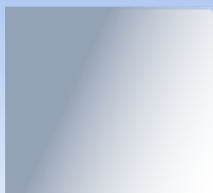
4,2 ★



GTA 4

Lorem ipsum is a good game in where a soldier have a mission to kill all her enemies ipsum ...

4,2 ★



Call of Duty: Modern War

Lorem ipsum is a good game in where a soldier have a mission to kill all her enemies ipsum ...

4,2 ★

Requête (par défaut)

```
select titre, description_jeux
from Jeux
order by titre;
```

Requête par défaut pour l'affichage des jeux de la boutique, triés par ordre alphabétique.

Requête (par date de parution)

```
select titre, description_jeux
from Jeux
order by date_sortie;
```

Requête pour l'affichage des jeux triés par ordre de date de parution.

Requête (par nombre de ventes)


```
select titre, description_jeux
from Jeux natural join Achat
group by titre, description_jeux
order by count(*) desc;
```

Requête pour l'affichage des jeux triés par ordre de nombre de ventes.

Requête (par note moyenne)

```
select titre, description_jeux
from Jeux natural left join Avis
group by titre, description_jeux
order by avg(note) desc;
```

Requête pour l'affichage des jeux triés par ordre de notes moyennes.

Page de recherche

Voici quelques exemples de requêtes pour une page de recherche.

Requête (par genre)

```
select id_jeux, titre, description_jeux
from Jeux natural join Appartient
where type_genre like '%action%';
```

Affiche les jeux dont le genre commence par le texte renseigner.

Requête (par éditeur)


```
select id_jeux, titre, description_jeux
from Jeux join Entreprise on Jeux.id_editeur = Entreprise.id_entreprise
where nom like '%microsoft%';
```


Affiche les jeux dont le nom de l'editeur commence par le texte renseigner.

Page de jeu détaillé

LOGO

BoutiqueCommunautésSupport

kenedy4Obama 



Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here,'making it look like readable English.

25.99 € Acheter

4,2 ★ [Editeur: Microsoft](#) [Développeur: Sony](#)

Commentaires

laPetiteMerde 29/01/2022 - 15:35
There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable. If you are going to

Requête (information nécessaire)

```
select id_jeux, titre, date_sortie, prix, age_requis, description_jeux,
editeur.nom as editeur, developpeur.nom as developpeur, avg(note) as
note_moyenne
from Jeux join Entreprise developpeur on
Jeux.id_developpeur=developpeur.id_entreprise join Entreprise editeur on
Jeux.id_editeur=editeur.id_entreprise natural join Avis
where titre='titre'
group by id_jeux, titre, date_sortie, prix, age_requis, description_jeux,
editeur.nom, developpeur.nom;
```

Récupère les informations du jeux vidéo de la page ainsi que sa note moyenne.

Requête (succès)

```
select intitule, conditions
from Jeux natural join Succes natural left join Debloque
where titre='titre' and pseudo='pseudo';
```

Sert a afficher les succès d'un jeu si l'utilisateur le détient.

Page de profil personnel

Maquette non réalisé

Requête (donner un avis)

```
insert into Avis(pseudo, id_jeux, note, commentaire)
select "pseudo", id_jeux, 4, "Le graphisme du jeu est très bien fait.
Félicitations aux développeurs !"
from Jeux
where titre='titre';
```

Cette requête permet au joueur de donner un avis.

Requête (mise à jour commentaire)

```
update Avis
set commentaire="Cette nouvelle version a ete une catastrophe niveau graphisme.
Je suis très déçu."
where pseudo='pseudo' and id_jeux=1;
```

Cette requête permet au joueur de mettre à jour l'avis précédemment donné.

Organisation du travail

Ce projet a été très intéressant et très fun à faire. Les premières parties du projet, tels que la conception du schéma d'entité-association, ont été réalisé ensemble afin de pouvoir se mettre d'accord sur la conception de notre site web. Cependant, par la suite nous avons réalisé différents travaux chacun de notre côté pour ensuite les mettre en commun. C'est ce qui nous a permis de réaliser ce travail a temps. Léo a fait la modélisation du cahier des charges. Je me suis chargé de l'implémentation et de la mise en place des requêtes. J'ai fait les maquettes et Léo s'est occupé de réaliser les requêtes nécessaires pour chaque maquette.