

Rapport de projet Base de Données

Licence Informatique 3ème année (L3)

Réservation de salle d'arcade

Luxon JEAN-PIERRE & Pierre DIBO

Sommaire

I Objectif du projet.....	3
II Modélisation.....	4
1/ Modèle relationnel.....	5
2/ Schéma relationnel.....	6
3/ Explication.....	7
III Requêtes.....	8
1/ Table intermédiaire arcadeplusfrequente.....	9
2/ Requête numéro 6.....	10
IV Problèmes rencontrés/Points faibles.....	11
1/ Contrainte d'insertion.....	11
2/ Absence de déclencheur et utilisation d'une variable calculée.....	11
3/ Requête imposée numéro 9.....	11
4/ Les jours fériés.....	12

I Objectif du projet

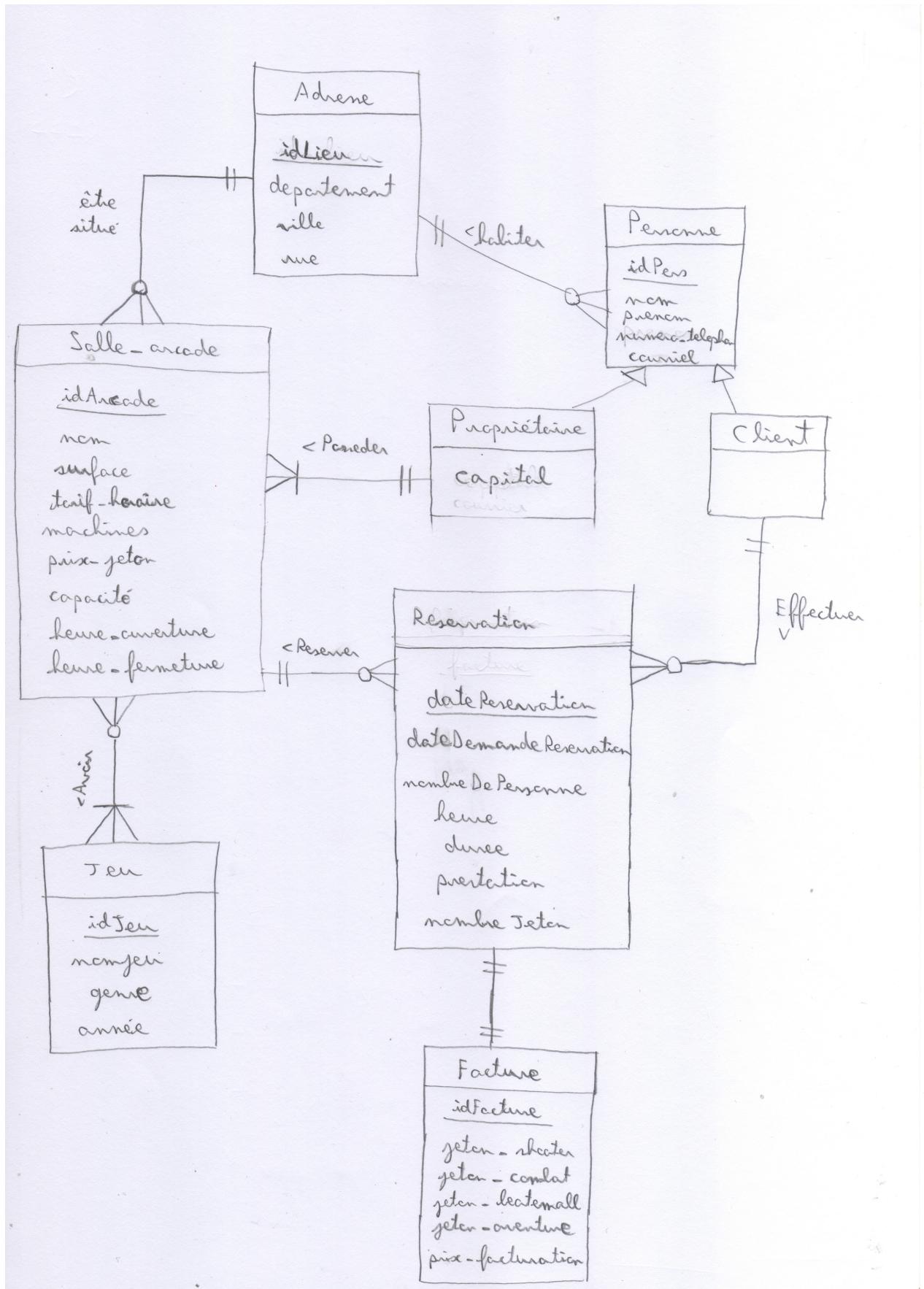
Dans une salle d'arcade, on achète des jetons pour pouvoir jouer sur des bornes d'arcade. Chaque salle d'arcade a un certain nombre de bornes et de jeux. Dans certaines salles, il est possible de réserver le lieu pour organiser un événement.

Le projet a pour but de créer une base de données relative au système de réservation d'une salle d'arcade. Le tarif d'une réservation est calculé en fonction du tarif horaire, défini par une salle d'arcade, de la durée de la réservation, et dans une certaine mesure, des jetons consommés.

II Modélisation

Cette partie traite de la modélisation d'une base de données relative au système de réservation d'une salle d'arcade.

1/ Modèle relationnel



2/ Schéma relationnel

Ici sont indiqués les différentes tables :

Table adresse :

adresse(idLieu, departement, ville ,rue)

Table jeu :

jeu(idJeu, nomjeu, genre, annee)

Table proprietaire :

proprietaire(idProprietaire, nom, prenom, capital, numero_telephone, courriel, #adresse)

Table client :

client(idClient, nom, prenom, numero_telephone, courriel, #adresse)

Table salle d'arcade :

salle_arcade(idArcade, nom, surface, tarif_horaire, machines, prix_jeton, capacite, heure_ouverture, heure_fermeture, #adresse, #proprietaire)

Table possede :

possede(#arcade,#jeu)

Table reservation :

reservation(#client, #arcade,#facture, dateReservation, dateDemandeReservation, nombreDePersonne, heure, duree, prestation, nombreJeton)

Table facture :

facture(idFacture , jeton_shooter , jeton_action, jeton_combat, jeton_beatemall, jeton_aventure,prix_facturation)

3/ Explication

Une salle d'arcade fait l'objet d'une réservation par un client, à condition que la salle soit disponible à l'heure demandées par le client et que la plage de réservation n'entre pas en conflit avec celle d'une autre réservation effectuée sur la même salle.

A partir d'une réservation, on peut connaître la salle concernée ainsi que le client. Chaque réservation contient des informations relatives à la date de celle-ci, l'heure, et la durée. On a également une facture qui est calculée dès la création de la réservation. Nous avons fait le choix de le stocker dans cette table car, d'une part, le tarif varie selon la salle, et d'autre part, le tarif d'une réservation peut varier pour une même salle dans l'année. L'existence de la facture est une erreur de conception dans la mesure où elle est calculée.

Toutes les salles d'arcade ont un(e) propriétaire. Le/La propriétaire peut aussi bien être une personne physique qu'une personne morale. Plusieurs salle d'arcade peuvent appartenir à une seule personne

Une salle possède aussi une bibliothèque de jeux jouables sur les bornes dédiée. Les jeux sont catégorisés par genre. Un jeu peut être présent dans plusieurs salles.

III Requêtes

Ici sont abordés les explications relatives à certaines requêtes personnelles. Elles concernent la table intermédiaire *arcadePlusfrequente* et la requête numéro 6.

1/ Table intermédiaire *arcadeplusfrequente*

La table intermédiaire *arcadePlusfrequente* permet d'obtenir l'identifiant de la salle arcade qui a été la plus fréquentée durant une année demandée par l'utilisateur. Cette table, obtenu à partir de la requête suivante (\$1 est le paramètre de la requête préparée) :

```
SELECT arcade as id, count(arcade) as nbReservation
FROM projet.reservation
WHERE $1<=extract(year from (select current_date)) and $1=extract(year from dateReservation)
GROUP BY id
ORDER BY nbReservation desc
LIMIT 1
```

est utilisée par les requêtes personnelles numéro 3 et 4. Ces requêtes permettaient respectivement de récupérer le nom de la salle et que les jeux qui sont dans la salle associée à l'identifiant récupéré.

2/ Requête numéro 6

Question :

- Quels sont les clients qui ont réservé une salle donnée à une année donnée ?

La salle et l'année sont donnée en paramètre. L'année ne peut pas dépasser l'année en cours.

Solution :

Trois tables sont sollicités : reservation, client, et salle_arcade. L'utilisation d'un SELECT par table dans le FROM permettait de ne garder que les informations nécessaires à la suite de la requête pour la clause WHERE.

En outre, l'utilisation de cette méthode permet d'optimiser le calcul du produit cartésien entre les trois tables dans la mesure où elles minorent la taille de la table résultante sur laquelle sera effectuée la clause WHERE.

Requête :

```
PREPARE requete_6(VARCHAR, INTEGER) AS
SELECT DISTINCT nom, prenom
FROM (SELECT idarcade, nom AS nomArcade FROM projet.salle_arcade WHERE nom=upper($1)) AS arcade2
CROSS JOIN (SELECT arcade, client AS idclt, dateReservation FROM projet.reservation
            WHERE $2<=EXTRACT(year FROM (SELECT current_date))
              AND EXTRACT(year FROM dateReservation)=$2 ) AS reserv
CROSS JOIN (SELECT idClient, nom, prenom FROM projet.client) AS client2
AND idarcade=arcade
AND idClient=idclt;
```

IV Problèmes rencontrés/Points faibles

1/ Contrainte d'insertion

Le premier problème rencontré dans le projet était le problème de conflit au niveau des plages horaires. En effet, pour deux réservations effectuées le même jour, il fallait s'assurer que les horaires, ainsi que les durées soient compatibles et ne provoquent pas de chevauchement entre les deux plages. Pour résoudre ce problème, nous avons dû faire l'usage d'une fonction qui vérifie les contraintes.

2/ Absence de déclencheur et utilisation d'une variable calculée

Lors de l'insertion d'une ligne de réservation, le contrôle des paramètres devait se faire en entrée. Cela impliquait l'utilisation d'un déclencheur (trigger). Or ce trigger n'a pu être défini à cause d'un problème de conception. En effet, dans le modèle relationnel, on peut voir une table Facture censée représenter la facture relative à une réservation. Cette facture étant calculée, elle n'avait pas lieu d'être dans le modèle. De plus, la fonction d'insertion dans la table reservation insérait en interne une ligne dans la table facture, ce qui montre bien que la facture est calculée dynamiquement à l'insertion, alors qu'elle devrait être établie via une requête SELECT.

3/ Requête imposée numéro 9

La réalisation de la requête imposée numéro 9 est tout bonnement impossible à satisfaire au vu de la modélisation de la base de donnée. En effet, cette question implique qu'il y ait un ordre de prestation selon un type. Or dans notre base de données, on définit pas d'ordre dans les types de prestations, chaque prestation étant relative à une réservation. Cette question est trop particulière et spécifique pour être appliquée à tous les types de système de réservation.

4/ Les jours fériés

Dans le projet, il y a eu un problème de communication ayant amené à une gestion des jours fériés. Ce cas ne correspond pas à la réalité. En effet, il existe des salles d'arcades qui sont ouvertes les jours fériés. Notre système n'est donc pas adapté aux salles concernés. Il a donc fallu supprimer ce cas inutile. Pour ce qui est des salles d'arcade qui ne sont pas ouverts les jours fériés, c'est à l'utilisateur de faire attention aux données qu'il entre.