PROJET DE BASE DE DONNEE

OUARDIA LABBACI

ABDERRAHMANE HADJABDELKADER

Master CCI 2023-2024

RÉSUMÉ

Le projet vise à développer un système de réservation pour les théâtres de Marseille. Ce système gérera plusieurs aspects, y compris la gestion des théâtres, des salles, des places et des spectacles. Il permettra aux spectateurs de réserver des places pour une représentation spécifique, avec le prix variant en fonction de plusieurs facteurs. Le système sera flexible et permettra une programmation annuelle, facilitant ainsi les réservations des utilisateurs. De plus, il fournira des analyses sur la fréquentation des spectacles, ce qui aidera à améliorer la communication et à mener une politique culturelle ambitieuse à long terme. Ce projet est une étape importante vers la réalisation de l'ambition de Marseille de devenir une capitale culturelle.

SUJET: MARSEILLE 2013

Dans le prolongement de l'événement "Marseille Capitale de la culture 2013", la ville de Marseille a souhaité se doter d'un système de réservations gérant toutes les salles de théâtre de la ville. Il y a plusieurs théâtres (La Criée, la Minoterie, Le Gymnase, etc.). Chacun d'eux comporte une ou plusieurs salles. Chaque salle a une capacité donnée. Dans ces salles, il y a des places de diverses catégories, A, B, C. Pour certains spectacles, le placement dans une catégorie fixée est libre. Pour d'autres spectacles, les places sont numérotées. La répartition des places selon leur catégorie varie selon la salle. Par exemple, dans l'unique salle de la Minoterie, toutes les places sont de catégorie B, alors que la grande salle de la Criée offre 60% de places en catégorie A, 30% en catégorie B et le reste en catégorie C. Les spectacles sont programmés dans une seule salle, ils sont caractérisés par leur intitulé, leur genre (théâtre classique, comédie, spectacle musical, comédie musicale, cirque, ballet, one-(wo)man show, etc.), la catégorie du public ciblé (grand public, théâtre, expérimental, pour enfants, etc.), leur durée et leur période de programmation. Pour un même spectacle, il y a au plus une représentation par jour durant la période de programmation. L'heure de début varie selon les représentations. Les spectateurs sont invités à faire leur réservation en choisissant une représentation donnée. Lors d'une même réservation, ils peuvent demander plusieurs places (dans la limite des places disponibles), mais dans une même catégorie. Le prix à payer pour une personne pour assister à une représentation dépend de la salle dans laquelle elle a lieu, de la catégorie de la place, et du type de spectacle. Les spectacles potentiellement très demandés (comédiens célèbres) voient leur tarification augmentée. Ceux ayant lieu l'après-midi bénéficient d'un rabais. Le système doit être flexible et permettre aux services culturels d'établir la programmation à l'année et aux utilisateurs d'effectuer des réservations. Le système devra permettre d'analyser la fréquentation des spectacles (nombre de spectateurs, âge, ...) et faciliter la communication, ce qui permettrait à plus long terme de mener une politique culturelle ambitieuse.

LE DICTIONNAIRE DES DONNÉES ET DES PROPRIÉTÉS

Table	Attribut	Type	Contrainte	Longueur	Description	Règle
HALO_Spectateur	idSpectateur	int	PK	5	Identifiant technique	Création - Autoincremant
	nom	varchar	NOT NULL	40	Nom du spectateur	Création
	prenom	varchar	NOT NULL	40	Prenom du spectateur	Création
	age	int	NOT NULL	2	Age du spectateur	Création
HALO_Periode	idPeriode	int	PK	2	Identifiant technique	Création - Autoincremant
	libelePeriode	varchar	NOT NULL	3	(WK, FA, PM,PN,,,etc)	Création par programmeur évenement
	pourcentageAugmentation	double	NOT NULL	5	exemple: 10%	Création
	pourcentageReduction	double	NOT NULL	5	exemple : 15,20 %	Création
HALO_Genre	idgenre	int	PK	3	Identifiant technique	Création - Autoincremant
	typeSpectacle	varchar	NOT NULL	20	(Romantique, comédie musicale..	Création
	prix_genre	double	NOT NULL	5	Exemple: 12,89	Création
HALO_CategorieSpectacle	idCategoriePublic	int	PK	5	Identifiant technique	Création - Autoincremant
	categoriePublic	varchar	NOT NULL	10	Adult,enfant ,,,,	Création
HALO_Theatre	idTheatre	int	PK	5	Identifiant technique	Création - Autoincremant
	nomTheatre	varchar	NOT NULL	40	Nom du théatre	Création
HALO_Salle	idSalle	int	PK	5	Identifiant technique	Création - Autoincremant
	nom	varchar	NOT NULL	40	Nom de salle	Création
	capacite	int	NOT NULL	3	100	Création
	pourcentageA	double	NOT NULL	5	exemple: 10 %	Création
	pourcentageB	double	NOT NULL	5	exemple : 15,20 %	Création
	pourcentageC	double	NOT NULL	5	exemple: 10 %	Création
	prix_place	double	NOT NULL	5	Exemple: 15,89	Création

HALO_Representation	idRepresentation	int	PK	5	Identifiant technique	Création - Autoincremant
	heureDebut	timestamp	NOT NULL	19	YYYY-MM-DD HH:MM	Création
HALO_Acteur	actor_id	int	PK	5	Identifiant technique	Création - Autoincremant
	nom	varchar	NOT NULL	40	Nom d'acteur	Création
	prenom	varchar	NOT NULL	40	Prénom d'acteur	Création
	popularity	varchar		20	haute,NULL,moyenne	Création
	poucentageAugmentation	double	NOT NULL	5	exemple : 15,20 %	Création
HALO_CatgoriePlace	IdCategorie	int	PK	5	Identifiant technique	Création - Autoincremant
	Type	String	NOT NULL	1	"A","B","C"	Définie à la création
HALO_Place	idPlace	int	PK	5	Identifiant technique	Création - Autoincremant
	numero	int		5	numéro de place	Création
HALO_Reservation	idReservation	int	PK	5	Identifiant technique	Création - Autoincremant
	prixTotal	double	NOT NULL	5	prix_place*number_places* (1+PlaceAugmentation/100)*(1-PourcetationReduction/100)	Céation
HALO_Spectacle	titre	double	NOT NULL	5	nom du spectacle	Céation
	specId	int	PK	5	Identifiant technique	Création - Autoincremant
	duree	double	NOT NULL	4	Duree en hh:min	creation
	placementCategorie	bool	NOTNULL	5	true or false	création

TOTAL = (Prix spectacle+ prix du genre + (nombre de réservations * prix de place)) * (Pourcentage augmentation periode + Pourcentage augmentation acteur - pourcentage reduction)

LES RÈGLES DE GESTION(AJOUTES DES EXEMPLES À LA FIN)

1)Un spectateur peut avoir un seul et unique Id selon son nom, prénom et âge.

2) Une période est un attribut qui contribue au calcul du prix final et varie selon les jours de la semaine, les journées fériés, les jours de fêtes, weekend, période matinale ou soir. Selon ces variation une augmentation ou une réduction de prix peut être appliquée.

3)Les Genre de spectacles sont définis par un type de spectacle et un prix de base

4) Les catégories de spectacles ayant un seul Id permettent de définir le type de publicité à projeter lors du visionnage.

5) Un théâtre possède un nom et un Id uniques .

6) Les salles de théâtres sont différencier par un ID, nom et capacité; chaque salle contient un pourcentage différent de places A, B et C ainsi qu'un prix de place propre à elle .

7)Représentation : définie par un id et une heure de début de spectacle.

8) Acteur: un acteur est identifié par son Id, peut participer à plusieurs spectacles, un critère de popularité qui participe au calcul du prix du spectacle.

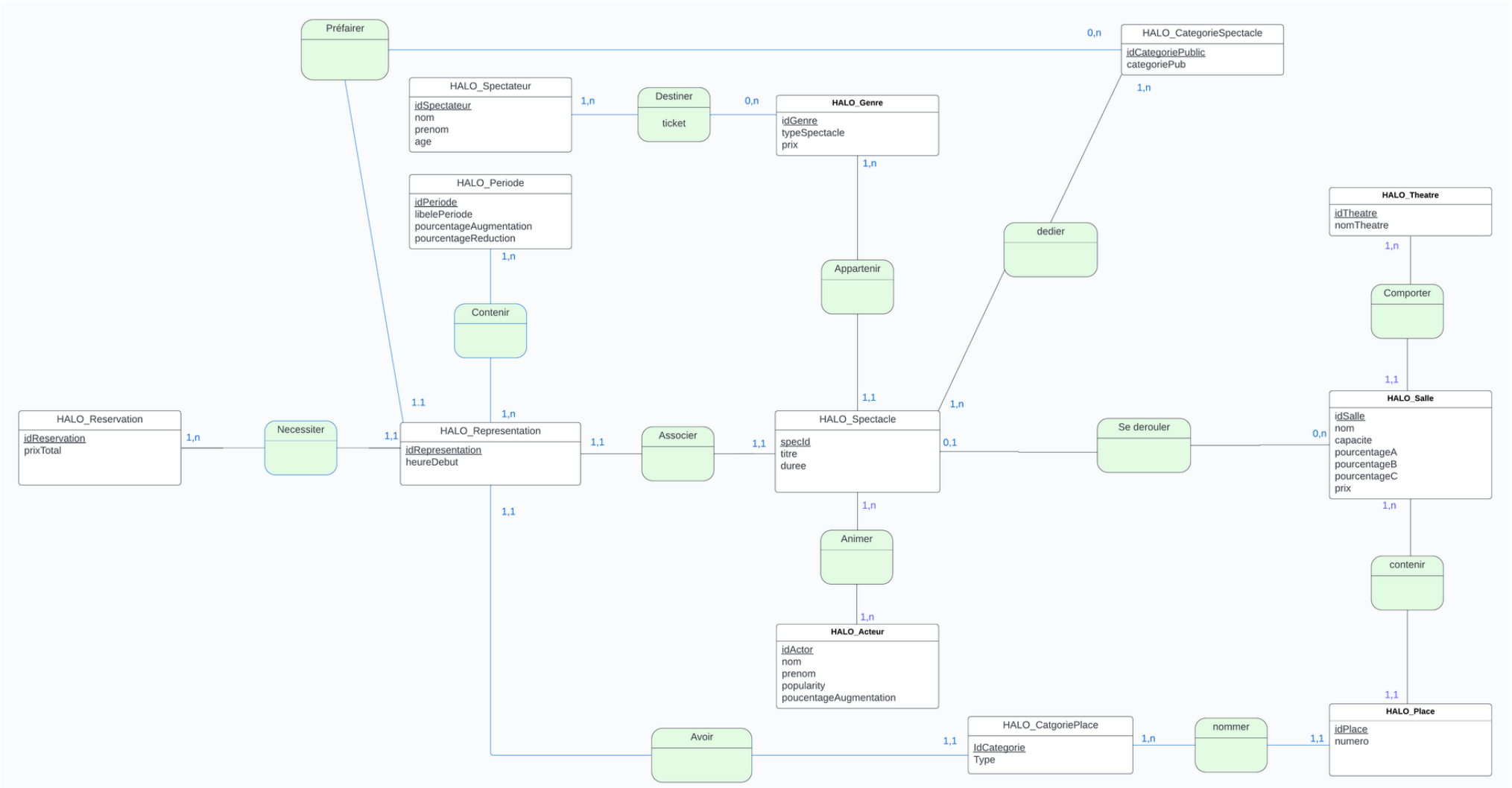
9) CatgoriePlace: Chaque place de chaque salle a son propre type "A", "B" ou "C". Les types varient selon la situation du siège dans la salle, "A" pour carré d'or par exemple.

10) Place : une place est identifié par un numéro et un id pour chaque billet de spectacle vendu et pour chaque salle.

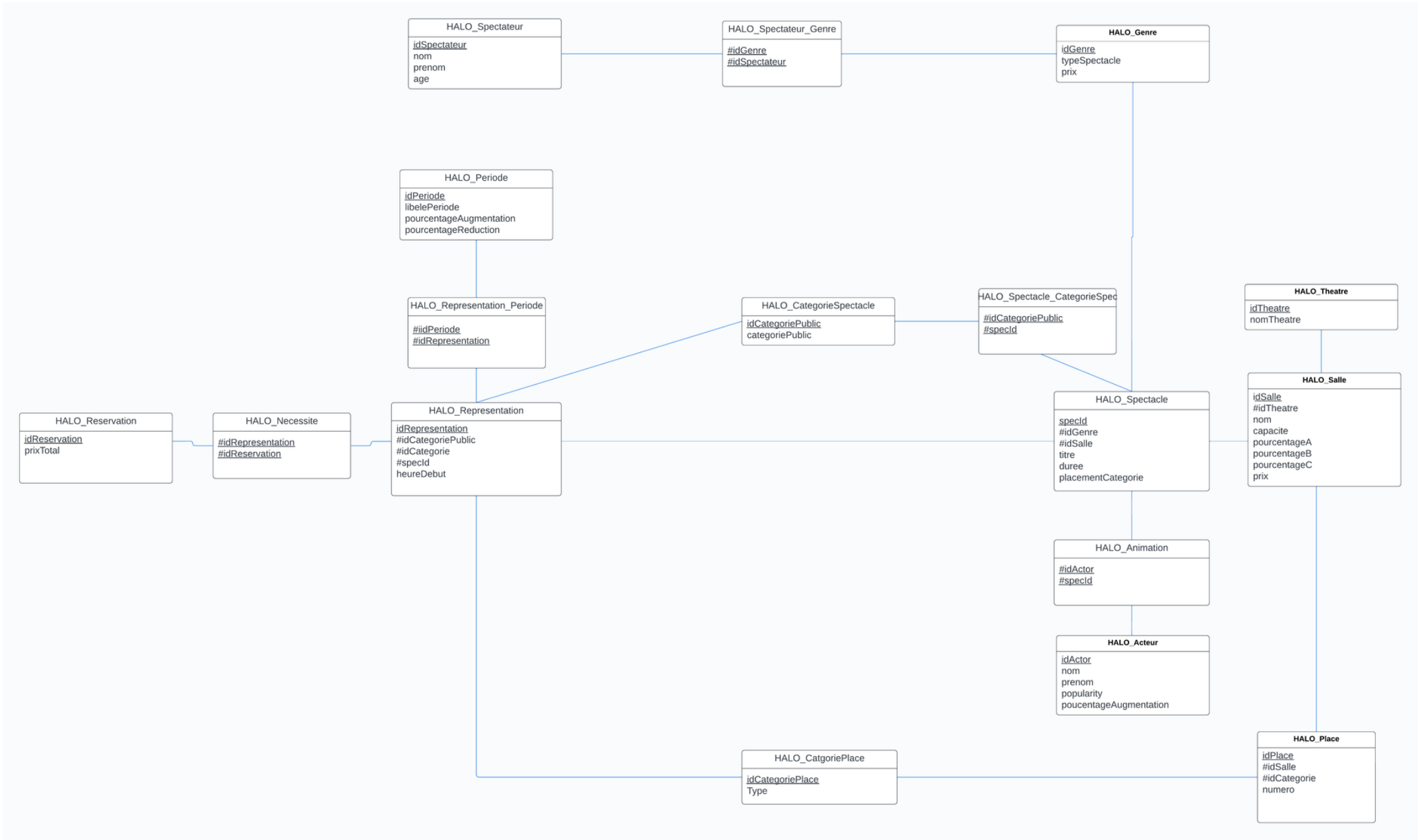
11) Reservation : Chaque réservation retourne le prix total du/des billets achetés ainsi qu'un id unique.

12) Spectacle : Un spectacle est différencié par un id, un titre, une durée de visionnage.

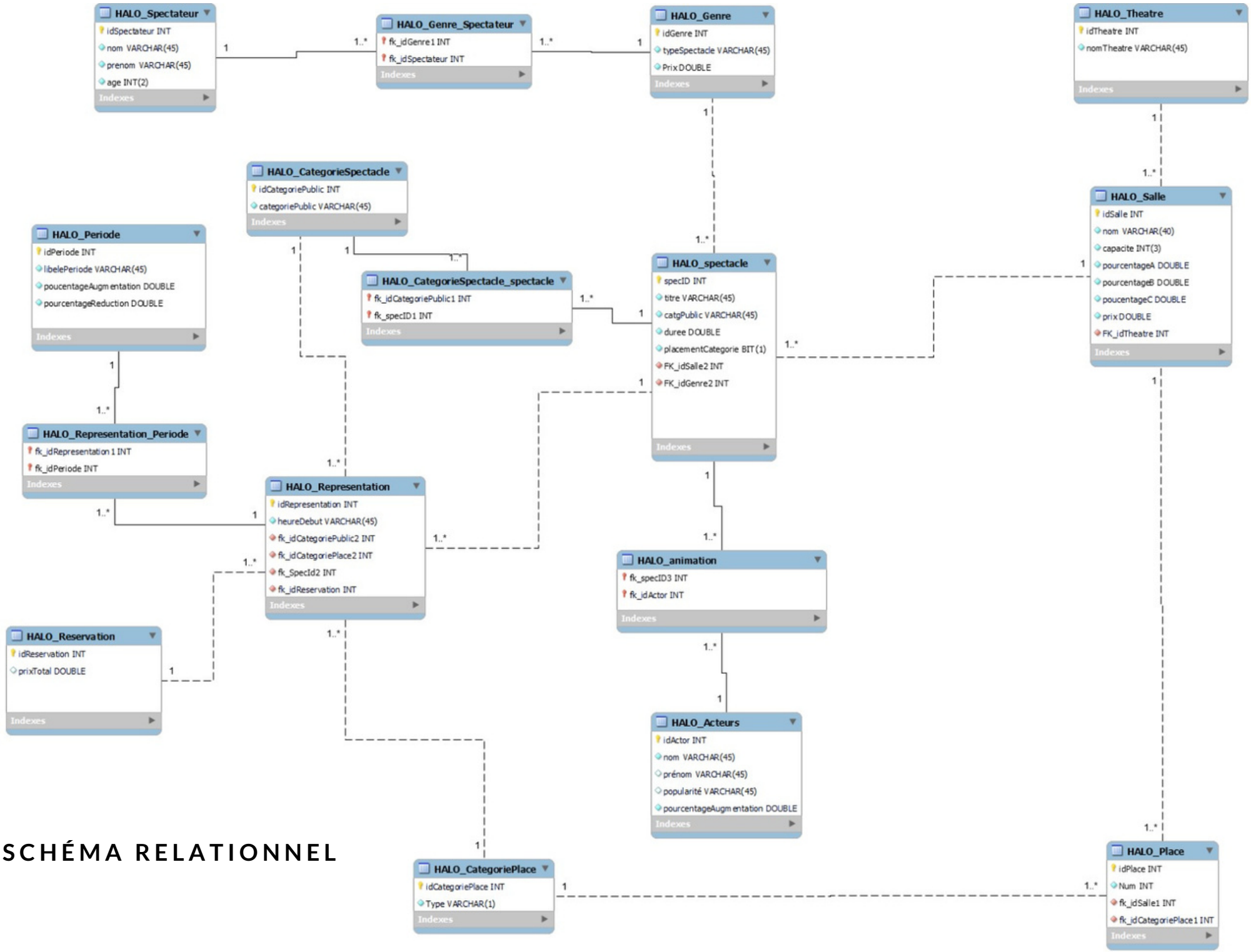
SCHÉMA MCD


Préfairer
0,n
HALO_CategorieSpectacle
idCategoriePublic
categoriePub
1,n
HALO_Spectateur
idSpectateur
nom
prenom
age
1,n
Destiner
ticket
0,n
HALO_Genre
idGenre
typeSpectacle
prix
1,n
HALO_Periode
idPeriode
libelePeriode
pourcentageAugmentation
pourcentageReduction
1,n
Contenir
Appartenir
dedier
HALO_Theatre
idTheatre
nomTheatre
1,n
Comporter
1,1
HALO_Salle
idSalle
nom
capacite
pourcentageA
pourcentageB
pourcentageC
prix
1,n
contenir
1,1
HALO_Place
idPlace
numero
1,1
HALO_Reservation
idReservation
prixTotal
1,n
Necessiter
1,1
HALO_Representation
idRepresentation
heureDebut
1,1
1,n
Associer
1,1
HALO_Spectacle
specId
titre
duree
1,n
0,1
Se derouler
0,n
1,1
1,n
Animer
1,n
HALO_Acteur
idActor
nom
prenom
popularity
poucentageAugmentation
1,1
Avoir
1,1
HALO_CatgoriePlace
IdCategorie
Type
1,n
nommer
1,1

LE SCHÉMA MLD



LE SCHÉMA RELATIONNEL



EXEMPLES

Exemples de sélection

Exemple 1 :

```
SELECT
    rs.idReservation,rep.heureDebut,rep.idRepresentation,per.libelePeriode
FROM
    halo_reservation AS rs
INNER JOIN halo_representation AS rep ON rs.idReservation = rep.fk_idReservation
LEFT JOIN halo_representation_periode AS reper ON rep.idRepresentation =reper.fk_idRepresentation1
LEFT join halo_periode as per ON reper.fk_idPeriode=per.idPeriode

WHERE rs.idReservation = 2;
```

t Grid | Filter Rows: | Export: | Wrap Cell Content:

idReservation	heureDebut	idRepresentation	libelePeriode
	2024-04-10 22:00:00	2	bf
	2024-01-10 18:00:00	5	NULL

- Récupération des données spécifiques : La requête vise à extraire des informations spécifiques de la base de données.
- Filtrage sur une réservation particulière : La clause **WHERE** avec la condition rs.idReservation = 2 restreint les résultats à la réservation avec l'identifiant 2.
- Colonnes sélectionnées : Les colonnes sélectionnées dans la clause **SELECT** (rs.idReservation, rep.heureDebut, rep.idRepresentation, per.libelePeriode) indiquent les données spécifiques nécessaires pour cette requête.
- Utilisation de jointures : Les jointures **INNER JOIN** et **LEFT JOIN** sont utilisées pour combiner les données provenant de différentes tables (halo_reservation, halo_representation, halo_representation_periode, halo_periode) en fonction des relations spécifiées par les clés étrangères

Exemple 2 :

```
1  SELECT
2      *
3  FROM
4      halo_reservation
5          INNER JOIN
6      halo_representation ON halo_reservation.idReservation = halo_representation.fk_idReservation
7  WHERE
8      halo_reservation.idReservation = 2;
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

idReservation	prixTotal	idRepresentation	heureDebut	fk_idCategoriePublic2	fk_idCategoriePlace2	fk_SpecId2	fk_idReservation
2	0	2	2024-04-10 22:00:00	4	1	2	2
2	0	5	2024-01-10 18:00:00	5	1	1	2

Récupération des données liées à la réservation : La requête vise à extraire toutes les colonnes de données associées à une réservation spécifique.

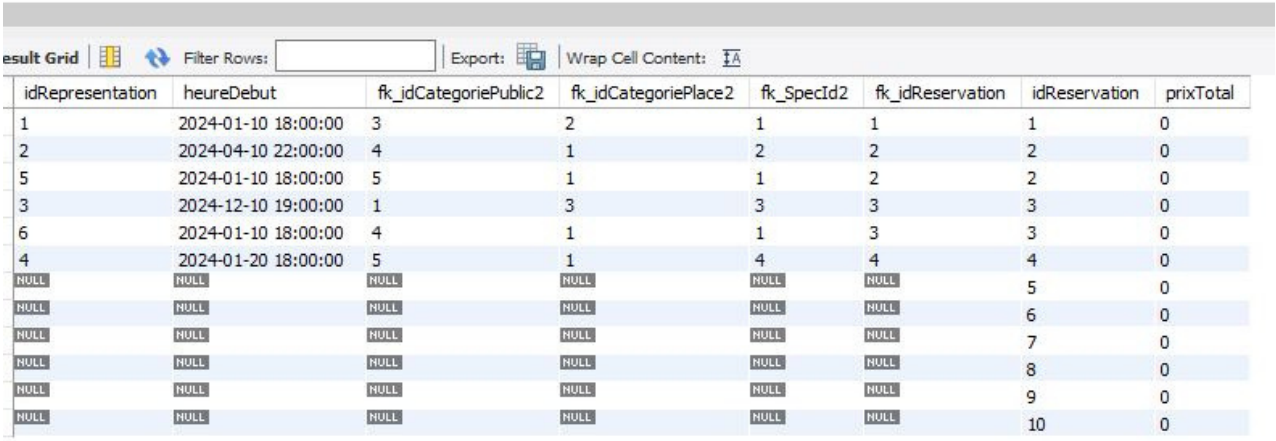
Filtrage sur une réservation particulière : La clause **WHERE** avec la condition halo_reservation.idReservation = 2 restreint les résultats à la réservation avec l'identifiant 2.

Utilisation de **l'INNER JOIN : L'INNER JOIN** avec la table halo_representation est utilisé pour combiner les données de la table halo_reservation avec celles de la table halo_representation en se basant sur la condition de clé étrangère halo_reservation.idReservation = halo_representation.fk_idReservation. Cela permet de récupérer les données liées à la représentation pour la réservation spécifique.

Sélection de toutes les colonnes : La clause **SELECT** * est utilisée pour récupérer toutes les colonnes des tables impliquées dans la requête (halo_reservation et halo_representation).

Exemple 3 :

```
1 •  select * from halo_representation
2    RIGHT JOIN halo_reservation ON halo_reservation.idReservation = halo_representation.fk_idReservation;
```


esult Grid | Filter Rows: | Export: | Wrap Cell Content:

idRepresentation	heureDebut	fk_idCategoriePublic2	fk_idCategoriePlace2	fk_SpecId2	fk_idReservation	idReservation	prixTotal
1	2024-01-10 18:00:00	3	2	1	1	1	0
2	2024-04-10 22:00:00	4	1	2	2	2	0
5	2024-01-10 18:00:00	5	1	1	2	2	0
3	2024-12-10 19:00:00	1	3	3	3	3	0
6	2024-01-10 18:00:00	4	1	1	3	3	0
4	2024-01-20 18:00:00	5	1	4	4	4	0
NULL	NULL	NULL	NULL	NULL	NULL	5	0
NULL	NULL	NULL	NULL	NULL	NULL	6	0
NULL	NULL	NULL	NULL	NULL	NULL	7	0
NULL	NULL	NULL	NULL	NULL	NULL	8	0
NULL	NULL	NULL	NULL	NULL	NULL	9	0
NULL	NULL	NULL	NULL	NULL	NULL	10	0

Cette requête récupère toutes les colonnes des tables halo_reservation et halo_representation, en incluant toutes les réservations (et leurs représentations associées le cas échéant) de la table halo_reservation, même si elles n'ont pas de correspondance dans la table halo_representation. Les colonnes de halo_representation pour les réservations sans représentations associées seront remplies de valeurs NULL dans le résultat.

Insertion:

Une requête d'insertion en SQL est utilisée pour ajouter de nouvelles lignes de données dans une table. Voici la syntaxe générale pour une requête d'insertion :

INSERT INTO nom_de_la_table (colonne1, colonne2, colonne3, ...)
VALUES (valeur1, valeur2, valeur3, ...)**;**

Example dans notre cas:
INSERT INTO halo_salle (nom, capacite, pourcentageA, pourcentageB, poucentageC, prix, FK_idTheatre) VALUES
('Salle A La Criée', 30, 0.4, 0.3, 0.3, 20.0, 15),
('Salle B La Criée', 20, 0.5, 0.4, 0.1, 15.0, 15);

Création de tables:
la création des tables se fait avec cette requête:

CREATE TABLE nom_de_la_table (
  colonne1 type_de_donnée_contraintes,
  colonne2 type_de_donnée_contraintes,
  colonne3 type_de_donnée_contraintes,
  ...
  contraintes_de_table
    clés etrangers,
    Reference
    ON DELETE ,
    ON UPDATE

);

-Reference:
La référence représente la table que la clé étrangers et en relation.

ON DELETE:
Cette clause spécifie l'action à entreprendre lorsqu'une ligne référencée dans la table parent est supprimée.

ON UPDATE :
Cette clause spécifie l'action à entreprendre lorsqu'une colonne référencée dans la table parent est mise à jour.

Exemple:

```
CREATE TABLE IF NOT EXISTS `HALO_DB`.`HALO_CategoriePlace` (
 `idCategoriePlace` INT(11) NOT NULL AUTO_INCREMENT,
 `Type` VARCHAR(1) NOT NULL,
 PRIMARY KEY (`idCategoriePlace`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
```

```
CREATE TABLE IF NOT EXISTS `HALO_DB`.`HALO_Theatre` (
 `idTheatre` INT(11) NOT NULL AUTO_INCREMENT,
 `nomTheatre` VARCHAR(45) NOT NULL,
 PRIMARY KEY (`idTheatre`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
```

CREATE SCHEMA IF NOT EXISTS `HALO_DB` DEFAULT CHARACTER SET utf8 ;

Puisque on est sur MYSQL workbench on doit crée un schémas avant la création des tables

si on est pas sur cette plateforme ou une autre similaire on se contente de la création d'une base de donnée avec la syntaxe:

```
CREATE DATABASE IF NOT EXISTS `HALO_DB`;
```

Calcul prix Total:

Pour le calcul du prix total de la réservation sa sera du coté application en utilisant un Transact-SQL (ou encore appelé T-SQL) est un langage de programmation procédural database. Il est le monopole de Microsoft, qui est utilisé dans SQL Serve ou d'autre langages du même type.

*Si on le calcul avec une requête elle sera beaucoup plus longue et compliquée.