

# Énoncé

On veut informatiser la gestion d'un lieu culturel en créant une base de données relationnelle. Après un entretien avec les gestionnaires de l'association du lieu culturel, voici les informations dont on dispose :

Notre espace culturel est composé de 3 salles de spectacle de différentes capacités (une de 300 personnes, une de 100 personnes et une de 50 personnes). Des événements sont organisés dans ces 3 salles, parfois dans le cadre d'événements communs, type festival, et parfois en utilisant une ou deux salles uniquement. Des événements et spectacles différents peuvent avoir lieu au même moment s'ils n'utilisent pas les mêmes salles.

Nous accueillons différents types de spectacles : musique, théâtre, performances, lectures, etc. Chaque événement peut proposer un ou plusieurs spectacle. Chaque artiste reçoit un cachet prédéfini pour sa performance, et chaque spectacle a un ou plusieurs tarifs pour le public (tarif plein, tarif réduit, tarif bénévole, tarif jeune, etc.). Un artiste reçoit son paiement après sa performance, il faut savoir si un artiste a été payé ou non. Les artistes sont représentés à minima par leurs nom et description. On veut conserver un historique des événements, spectacles et artistes qui se sont produits dans notre lieu.

Chaque spectacle a un nombre de places limité en fonction de la salle dans laquelle il se déroule. Les personnes du public peuvent réserver des tickets en avance si elles ont un compte dans la base de données. Les tickets réservés pourront être payés à l'avance ou sur place. Si une personne ayant réservé un ticket ne se présente pas 15 minutes au plus tard avant le début du spectacle, alors on considère que son ticket peut être revendu sur place (Les places disponibles ne sont donc plus nécessairement les mêmes 15min avant le début du spectacle). Un événement peut-être annulé. Les tickets peuvent alors être remboursés, mais on veut conserver un historique de ceux-ci.

De nombreux bénévoles participent à la vie du lieu culturel. Ils peuvent être bénévoles "génériques" ou bien avoir de spécificités comme aide à la comptabilité, connaissances techniques musicales, aide à la sécurité etc. Ils ont des tarifs préférentiels sur les événements et les consommations. Ils peuvent avoir des missions longues (sur l'année, par exemple pour choisir la programmation), et ils peuvent aussi s'inscrire comme bénévole pour les événements, pour préciser qu'ils sont disponibles pour aider.

Certains bénévoles peuvent être désignés responsables d'un pôle (programmation, organisation, sécurité, nettoyage, etc.) et aussi d'un événement. Lors d'événements, on pourra leur assigner une liste de bénévoles pour qu'ils puissent gérer les équipes des autres bénévoles participant à un événement.

Nous avons aussi une dizaine de salariés qui occupent différents postes dans l'association (technicien, administratif, sécurité, etc.) en plus des membres de la présidence de l'association. Certains membres et salariés sont responsables les uns des autres. Les salariés ont un salaire mensuel prédéfini.

On va considérer que toute personne peut être identifiée à minima par ses nom, prénom, date de naissance, adresse, numéro de téléphone et adresse email.

# Partie 1 - Conception (MCD)

## 1. Extraire les propositions pertinentes de l'énoncé:

- ❖ **l'espace culturel est composé de salles de spectacle**
  - Notre espace culturel est composé de 3 salles de spectacle.
  - parfois en utilisant une ou deux salles uniquement.
- ❖ **Des événement sont organisés**
  - Des événements sont organisés dans ces 3 salles,
  - Chaque événement peut proposer un ou plusieurs spectacle
  - Un événement peut-être annulé
- ❖ **des spectacles sont organisés**
  - Des événements et spectacles différents peuvent avoir lieu
  - Nous accueillons différents types de spectacles
  - chaque spectacle a un ou plusieurs tarifs pour le public
  - Chaque spectacle a un nombre de places limité en fonction de la salle dans laquelle il se déroule
- ❖ **des artiste qui anime les spectacle**
  - Chaque artiste reçoit un cachet prédéfini pour sa performance
  - il faut savoir si un artiste a été payé ou non
  - Les artistes sont représentés à minima par leurs nom et description
- ❖ **On veut conserver un historique des événements, spectacles et artistes qui se sont produits dans notre lieu.**
  - il y a une date pour les événements, spectacles et artistes .
- ❖ **les personne du publique qui ont assisté à l'événement**
  - Les personnes du public peuvent réserver des tickets .

❖ **des tickets pour pouvoir assisté à l'événement**

- réserver des tickets en avance si elles ont un compte dans la base de données.
- Les tickets réservés pourront être payés à l'avance ou sur place
- on considère que son ticket peut être revendu sur place
- Les tickets peuvent alors être remboursés
- on veut conserver un historique du ticket

❖ **des bénévoles participe la vie du lieu culturel**

- De nombreux bénévoles participent à la vie du lieu culturel
- Ils ont des tarifs préférentiels sur les événements et les consommations
- Ils peuvent avoir des missions longues
- ils peuvent aussi s'inscrire comme bénévole pour les événements
- Certains bénévoles peuvent être désignés responsables d'un pôle

❖ **on a des Employés de l'association**

- Nous avons aussi une dizaine de salariés qui occupent différents postes dans l'association
- Certains membres et salariés sont responsables les uns des autres
- Les salariés ont un salaire mensuel prédéfini.

❖ **toute personne peut être identifiée à minima par ses nom, prénom, date de naissance, adresse, numéro de téléphone et adresse email.**

**2. Proposer, sous forme de textes courts, les entités, attributs, associations et cardinalités:**

**une salle** de spectacle à id\_salle , une nbplace (nombre de place ) et nom\_salle .plusieur **événements et spectacles** sont organisés dans des salle .

**un événement** à idévénement , un nom ,d'un type de la période de l' organisation et de la date du debut et de fin de l'événement et sont état es que annulé ou maintenu , à venir. un événement est organisé dans une ou plusieurs **salle** .un événement propose un ou plusieurs **spectacles** . Un événement peut accueillir plusieurs **bénévoles**.

**un spectacle** à un idspectacle, un titre, un type, nombre de place consacré ,la date de début et date de fin de déroulement du spectacle et le tarif et tarif préf 1 et 2 pour les remises .Un spectacle peut être réalisé dans **un événement**. un spectacle contient un ou plusieurs **tickets**. un spectacle est animé par un ou plusieurs **artistes**.

**un artiste** à un id artiste, une description ,. des artistes peuvent participer à plusieurs **spectacles** .

**listArtistSpect** à la remuneration des Artiste et payé représente l'état de paiement

**un Ticket** à un ID ticket , une date, un prix , un état ( remboursé ou non). Plusieurs tickets peuvent être réservés par **les personnes** du public. Un seul ticket représente **un spectacle**.

**une personne** du public (client, employeur, artiste, bénévole) possède ID personne , un nom , prénom, date de naissance, adresse, numéro de téléphone et adresse email. Plusieurs personnes peuvent réserver des **Ticket**.

**un Employé** à un id Employer, id chef, date fin contrat, salaire, poste, rôle. Un salarié gère un ou plusieurs **événements**.

**un bénévole** possède un id bénévole , une spécificités, date debut, date fin. Les bénévoles peuvent participer à plusieurs **événements**.

### 3. Identifier les dépendances fonctionnelles:

id salle : capacité de la salle

id salle : événement

id événement : nom de l'événement

id événement : type de l'événement

id événement : Date de début

id événement : Date de fin

id événement : état de l'événement

id spectacle : titre du spectacle

id spectacle : type du spectacle

id spectacle : nombre de place du spectacle

id spectacle : date de début du spectacle

id spectacle : date de fin du spectacle

id spectacle : tarif pref1 du spectacle

id spectacle : tarif pref2 du spectacle

id spectacle : idevenement

id spectacle : idsalle

id artiste : description

id ticket : prix du ticket

id ticket : état du ticket

id ticket : id\_personne

id ticket : idspectacle

id personne : nom de la personne

id personne : prénom de la personne

id personne : date de naissance de la personne

id personne : adresse de la personne

id personne : adresse email de la personne

id personne : numéro de téléphone de la personne

id employee : poste employé  
id employee : idChef  
id employee : salaire  
id employee : date fin contrat Date

id bénévole : spécificité du bénévole  
id bénévole : date\_debut bénévole  
id bénévole : date de fin du bénévole

#### 4- schéma entité-association:

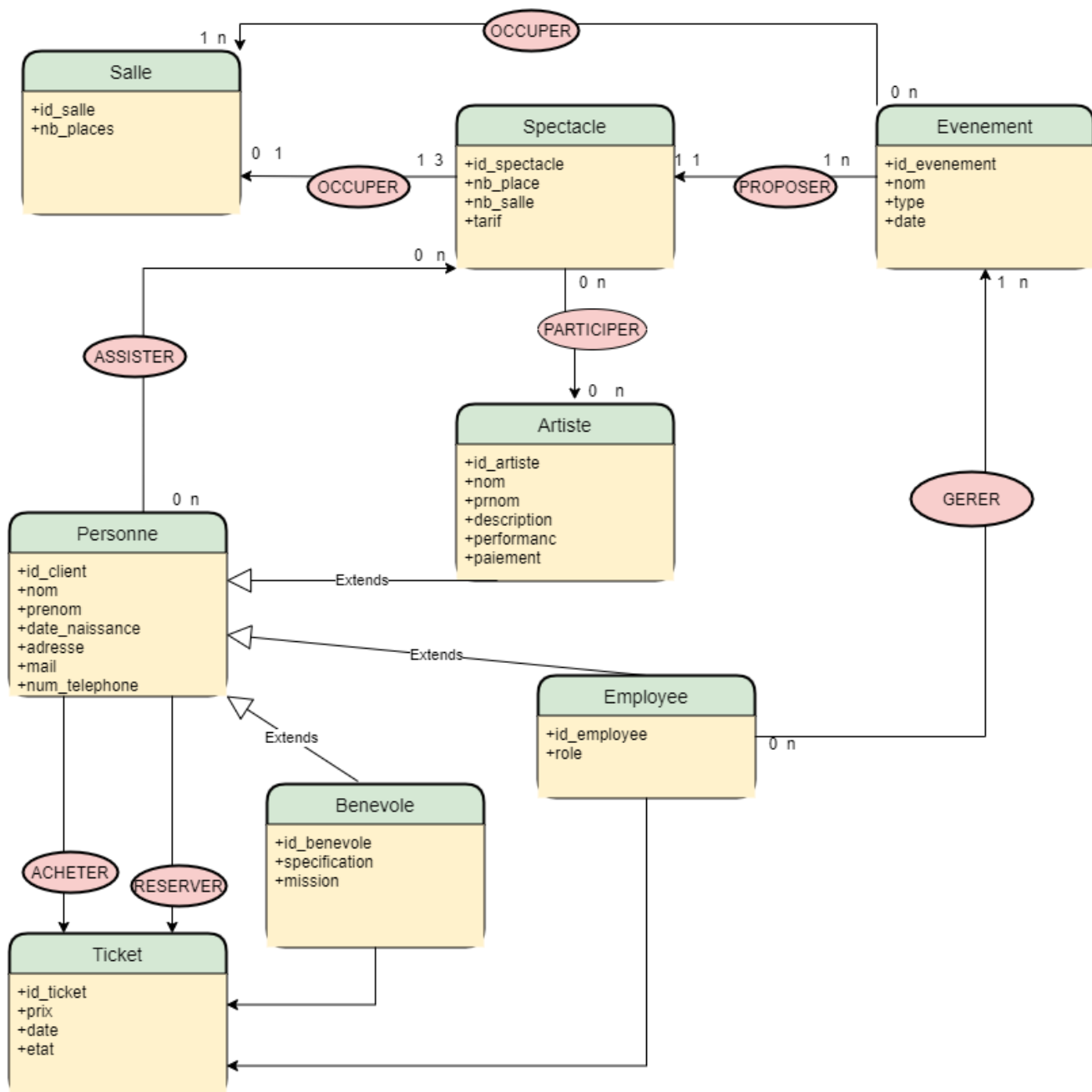


Schéma entité-association à l'aide du logiciel Lucidchart

## 5- justification que les entités sont en 3FN :

**salle** (idsalle, nbplace, nom\_salle)

**evenement**( idévenement, nom, typeEvenement, date\_debut, date\_fin, etat)

**spectacle**( idspectacle, titr\_spectacle, type\_spectacle, nbplace\_dispo, date\_debut, tarif\_base, date\_fin, idevenement, idsalle, tarif\_pref1, tarif\_pref2) .

**artiste** (idartiste, description) .

**ticket**( idticket, prix, état )

**personne**( idpersonne, nom, prénom, date\_naissance, adresse, mail, téléphone).

**employee** (idemployee, post, salaire, date\_debut\_contrat, date\_fin\_contrat)

**benevole** (idbénévole, specificite, date\_debut, date\_fin)

**Remarque :** les dépendances transitives ainsi que les **groupes répétitifs** ne sont pas présentes dans les relations.

Et vu qu'il n'y a pas de clé primaire composée, on déduit que les attributs dépendent de la totalité de la clé.

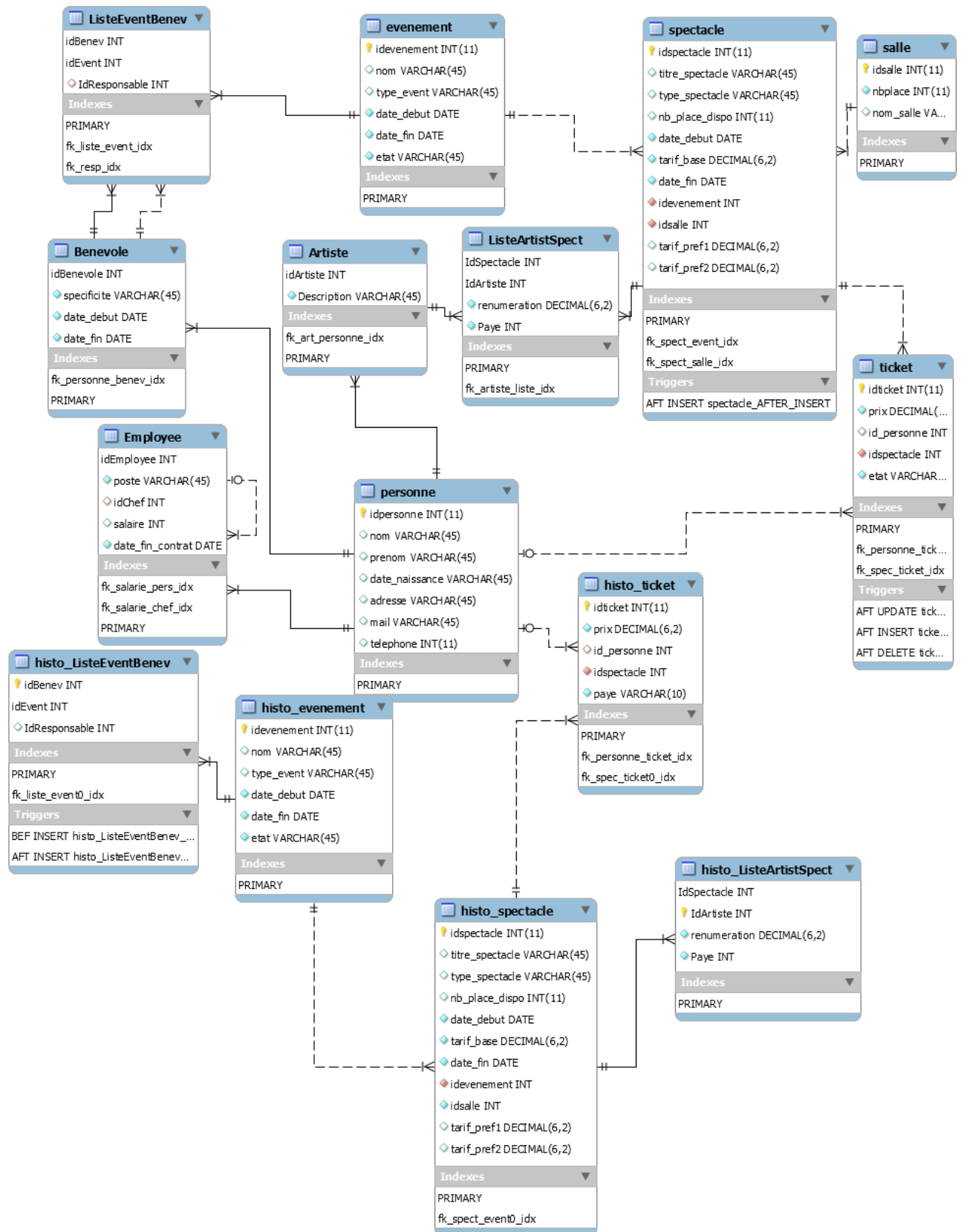
- On peut alors se rendre compte que les tables sont bien en 3ème forme normale.

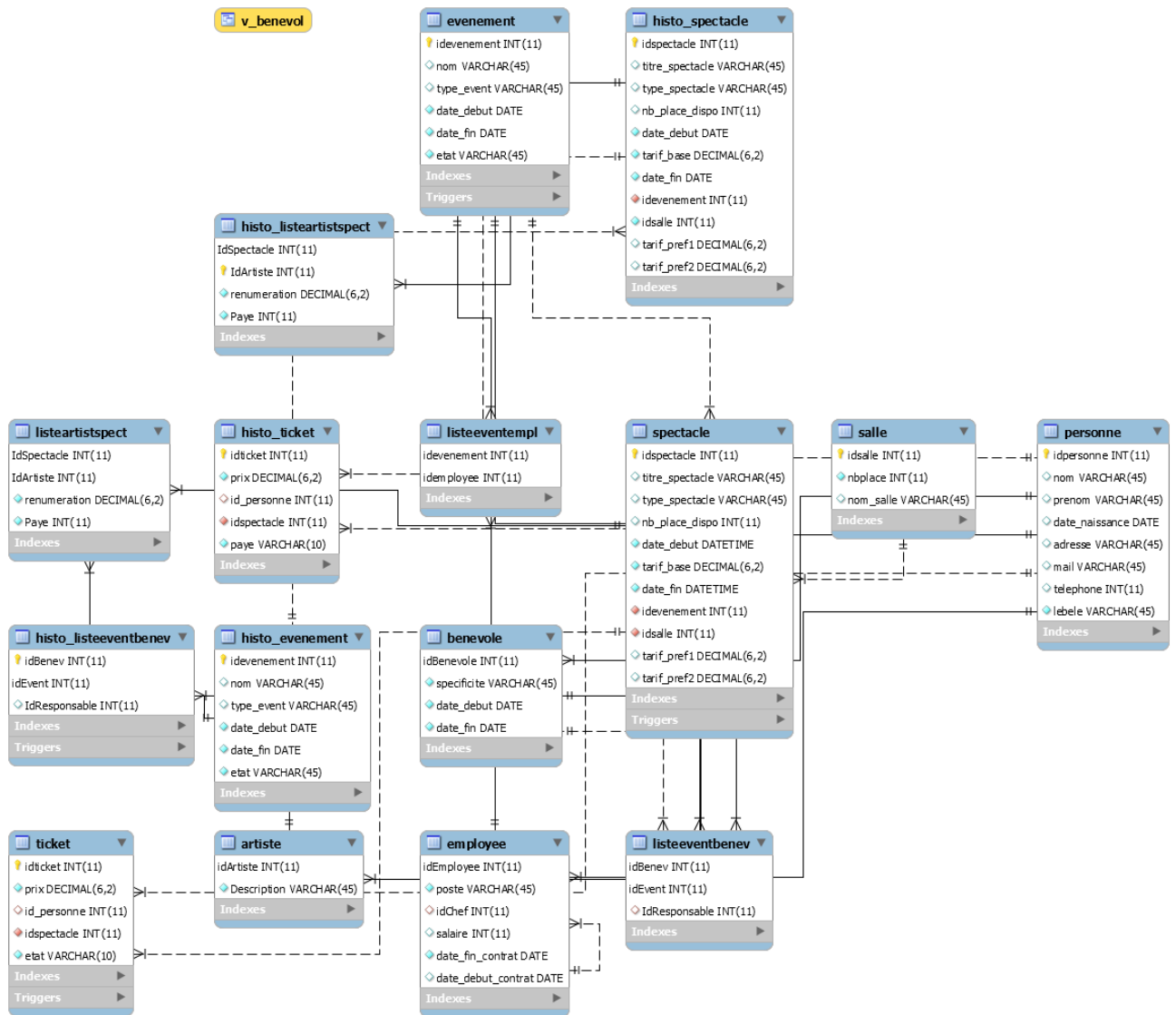
## Quelques erreurs et difficultés :





# Partie 2 - Production (MLD)





**remarque:** ce diagramme est généré à partir de notre database reversed pour avoir tout les triggers, routines et fonctions ajoutées,

# Partie 3- Requêtes (sql)

## 1- la liste des bénévoles

/\*1 Liste des bénévoles et de leur âge, ainsi que les événements auxquels ils ont participés\*/

```
select b.nom, b.prenom, floor(datediff(current_date(),date_naissance)/365) as age,e.nom as  
nom_evenement  
from v_benevol b  
left join listeventbenev l  
on b.idBenevole = l.idBenev  
left join evenement e on e.idEvenement = l.idEvent;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
nom	prenom	age	nom_evenement
e	f	8	journée des parent
d	e	32	jourée de l'amitié
c	d	21	journée des parent
b	c	20	journén de velo
a	b	31	NULL

/\*2 Montant des prestations réglées aux artistes entre deux dates données\*/

/\* on a choisie les 2 date suivant '2020-01-01' ET '2023 -07-05'\*/

```
select sum(renumeration) as paiement_artiste  
from listartistspect l, spectacle s  
where l.IdSpectacle = s.IdSpectacle  
and s.date_fin > '2020-01-01'  
and s.date_fin < '2023-07-05'  
and l.Payé= 1;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
paiement_artiste			
7200.00			

/\*3 nombre de places disponibles pour un événement \*/

```
select sum(nb_place_dispo)  
from spectacle  
where idevenement = '1';
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	sum(nb_place_dispo)	idevenement		
▶	296	1		

/\*4 Afficher le nombre de minutes, d'heures, de jours, de mois et d'années avant le début d'un événement\*/

/\* on a choisie idevenement = '7' \*/

```
SELECT TIMESTAMPDIFF(YEAR,current_date(),date_debut ) AS nb_annees,  
TIMESTAMPDIFF(MONTH,current_date(),date_debut ) AS nb_mois,  
TIMESTAMPDIFF(DAY,current_date(),date_debut) AS nb_jours,  
TIMESTAMPDIFF(HOUR,current_date(),date_debut ) AS nb_heures,  
TIMESTAMPDIFF(MINUTE,current_date(),date_debut ) AS nb_minuts,  
idevenement  
from evenement  
where idevenement = '7';
```

Result Grid

Filter Rows:

Export:

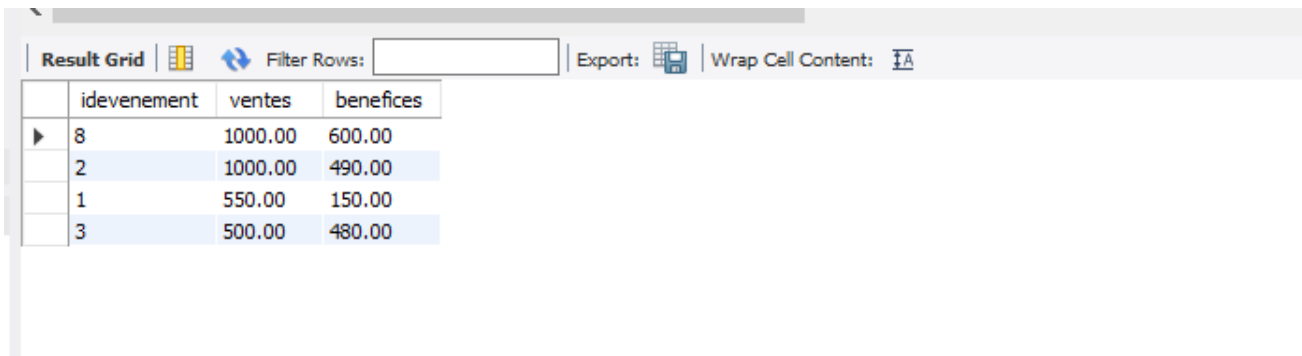
Wrap Cell Content:

	nb_annees	nb_mois	nb_jours	nb_heures	nb_minuts	idevenement
▶ 1	14	432	10368	622080	7	

*/\*5\*/*

*/\*Afficher les événements par ordre de meilleures ventes, avec les sommes collectées au total d'une part, et le bénéfice réalisé d'autre part (après paiement des artistes)  
rappel : ticket payé à la valeur PAYE dans etat\*/*

```
select idevenement,sum(prix) as ventes, sum(prix)-sum(renumeration) as benefices  
from ticket t,spectacle s, listeartistspect l  
where s.idspectacle = t.idspectacle  
and l.IdSpectacle = s.IdSpectacle  
and t.etat = 'PAYE'  
group by idevenement  
order by sum(prix) desc;
```



	idevenement	ventes	benefices
▶	8	1000.00	600.00
	2	1000.00	490.00
	1	550.00	150.00
	3	500.00	480.00

*/\*6*

1. Proposer une requête d'insertion d'un événement composé de 7 spectacles qui utilisent les 3 salles. Elle peut être composée de plusieurs requêtes et ne doit pas imposer d'identifiant ( == elle doit pouvoir fonctionner sans tenir compte des identifiants des autres événements ou spectacles)\*/

```
insert into evenement(idevenement,nom,type_event,date_debut,date_fin)  
select max(idevenement)+1,'event_test','cinema','2021-07-01','2021-07-07'  
FROM evenement;
```

*/\*spect 01\*/*

```
insert into spectacle(idspectacle,idevenement,date_debut,date_fin,tarif_base,idsalle)  
select max(idspectacle)+1,(select max(idevenement) FROM evenement),'2021-07-01  
18:00','2021-07-01 22:00',10,1  
FROM spectacle;
```

*/\*spect 02\*/*

```
insert into spectacle(idspectacle,idevenement,date_debut,date_fin,tarif_base,idsalle)  
select max(idspectacle)+1,(select max(idevenement) FROM evenement),'2021-07-02  
18:00','2021-07-02 22:00',10,2  
FROM spectacle;
```

*/\*spect 03\*/*

```
insert into spectacle(idspectacle,idevenement,date_debut,date_fin,tarif_base,idsalle)  
select max(idspectacle)+1,(select max(idevenement) FROM evenement),'2021-07-01  
18:00','2021-07-01 22:00',10,3
```

```

FROM spectacle;
/*spect 04*/
insert into spectacle(idspectacle,idevenement,date_debut,date_fin,tarif_base,idsalle)
select max(idspectacle)+1,(select max(idevenement) FROM evenement),'2021-07-01
14:00','2021-07-01 16:00',10,1
FROM spectacle;
/*spect 05*/
insert into spectacle(idspectacle,idevenement,date_debut,date_fin,tarif_base,idsalle)
select max(idspectacle)+1,(select max(idevenement) FROM evenement),'2021-07-03
18:00','2021-07-03 22:00',10,2
FROM spectacle;
/*spect 06*/
insert into spectacle(idspectacle,idevenement,date_debut,date_fin,tarif_base,idsalle)
select max(idspectacle)+1,(select max(idevenement) FROM evenement),'2021-07-04
18:00','2021-07-04 22:00',10,3
FROM spectacle;
/*spect 07*/
insert into spectacle(idspectacle,idevenement,date_debut,date_fin,tarif_base,idsalle)
select max(idspectacle)+1,(select max(idevenement) FROM evenement),'2021-07-05
18:00','2021-07-05 22:00',10,1
FROM spectacle;
/**/

```

	idevenement	nom	type_event	date_debut	date_fin	etat
	3	tour de france	musical	2021-06-26	2021-07-18	ok
	4	jourée de l'amitié	musical	2021-07-30	2021-07-30	ok
	5	journée mondial de tourism	culturel	2021-08-27	2021-08-29	ok
	6	journée des enseignants	culturel	2021-10-05	2021-10-05	ok
	7	journée de vélo	social	2022-06-03	2022-06-06	ok
	8	event_test	cinema	2021-07-01	2021-07-07	A VENIR
	9	event_test	cinema	2021-07-01	2021-07-07	A VENIR
	10	event_test	cinema	2021-07-01	2021-07-07	A VENIR
*	NULL	NULL	NULL	NULL	NULL	NULL

	idspectacle	type_spectacle	titre_spectacle	nom	idevenement
▶	1	music	concert1	event_test	10
	2	théâtre	concert2	event_test	10
	3	fete	concert3	event_test	10
	4	fete	concert4	event_test	10
	5	fete	concert5	event_test	10
	6	music	concert6	event_test	10
	7	style	concert7	event_test	10
	8	dance	concert8	event_test	10
	9	cinema	concert9	event_test	10

listeeventbenev 4    Result 13 x

Output :

/\*7

1. Les salaires annuels des salariés pour une année donnée\*/

/\*pour l'année 2019\*/

```
select sum(salaire*least(timestampdiff(MONTH,date_debut_contrat,'2019-12-31'),12))  
from employee  
where date_fin_contrat > '2019-12-31'  
and date_debut_contrat < '2019-12-31'
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
salaires			
91200			

/\*8

Les ventes d'un événements classées par tarifs (tarifs plein, jeune, etc.)

\*/

```
select s.idevenement,sum(if(t.prix=s.tarif_base,1,0)) as nb_tick_tarif_base,  
sum(if(t.prix=s.tarif_pref1,1,0)) as nb_ticket_tarif_pref1,  
sum(if(t.prix=s.tarif_pref2,1,0)) as nb_ticket_tarif_pref2  
from ticket t, spectacle s  
where t.idspectacle = s.idspectacle  
group by s.idevenement;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
idevenement	nb_tick_tarif_base	nb_ticket_tarif_pref1	nb_ticket_tarif_pref2
1	2	1	0
2	0	0	0
3	0	0	0
4	0	1	0
8	1	0	1

/\*9

Les pourcentages de types de spectacle pour une année donnée

\*/

```
select type_spectacle,round(100*count(*)/(select count(*) from spectacle where date_debut >  
'2021-01-01'  
and date_fin < '2022-01-01'),2) as 'nb%'  
from spectacle s  
where date_debut > '2021-01-01'
```

**and date\_fin < '2022-01-01'**  
**group by type\_spectacle;**

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
type_spectacle	nb%		
NULL	10.00		
cinema	10.00		
dance	10.00		
fete	30.00		
music	20.00		
style	10.00		
théâtre	10.00		

/\*10

Les événements annulés ainsi que le montant des tickets remboursés

\*/

**select e.idevenement,e.nom,sum(t.prix) as montant\_rembourse**  
**from evenement e, spectacle s, ticket t**  
**where e.etat = 'ANNULE'**  
**and s.idevenement = e.idevenement**  
**and t.idspectacle = s.idspectacle**  
**/\*and t.etat = 'REMBOURSE'\*/**  
**group by e.idevenement,e.nom;**

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
idevenement	nom	montant_rembourse	
1	journée des parent	1118.00	

La condition sur l'état du ticket a été commenté car on suppose que cette requête est faite sans passage d'une routine pour modifier les états des tickets du PAYE à REMBOURSE. Si une telle routine existait, la condition doit être mise.



## Partie 4- Contraintes et Triggers -

*/\*1 Un événement peut-être annulé, mais il ne peut pas être supprimé de la base de données, et un ticket vendu peut-être remboursé, mais pas supprimé*

*\*/*

faire un trigger (before\_delete) sur les table pour interdire la suppression

```
CREATE DEFINER='root'@'localhost' TRIGGER `events`.`evenement_BEFORE_DELETE`  
BEFORE DELETE ON `evenement` FOR EACH ROW  
BEGIN  
  DECLARE msg varchar(255);  
  SET msg = 'Interdit de supprimer un évènement';  
  SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;  
END
```

--pour le ticket, le même principe s'applique mais il faut pouvoir supprimer les tickets non payés

```
CREATE DEFINER='root'@'localhost' TRIGGER `events`.`ticket_BEFORE_DELETE` BEFORE  
DELETE ON `ticket` FOR EACH ROW  
BEGIN  
  DECLARE msg varchar(255);  
  if(old.etat!='RESERVE') then  
    SET msg = 'Interdit de supprimer un ticket';  
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;  
  end if;  
END
```

Une autre solution serait d'insérer les lignes dans les tables historiques juste avant de les supprimer.

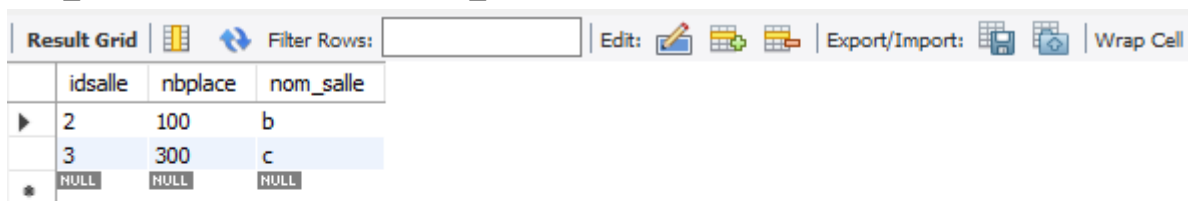
/\*2 Vérifier qu'un événement ou un spectacle ne peut pas être créé si un autre événement ou spectacle utilise déjà une des salles sur le même créneau

\*/

Fonction pour vérifier disponibilité des salles entre deux dates

La requête suivante nous renvoie les salles disponibles entre deux dates. Avec ce résultat, on peut décider si un événement peut être planifié entre ces deux dates ou pas.

```
select *  
from salle s  
where not exists (select null from spectacle where idsalle = s.idsalle and  
date_debut > '2021-06-01' and date_fin < '2021-06-07');
```



	idsalle	nbplace	nom_salle
▶	2	100	b
	3	300	c
*	NULL	NULL	NULL

/\*3 Un bénévole ne peut-être assigné qu'à un seul responsable par événement\*/

La clé primaire de la table listeventBenev est la combinaison des colonne idBenev et idEvent. Ce couple ne peut donc jamais être répété et donc un bénévole ne peut avoir plus qu'un responsable par événement

/\*4  
\*/  
une contrainte de type check ne permet pas que le nombre de places disponibles (déjà géré par trigger) garantit l'impossibilité de réserver plus que disponible

```
alter table spectacle add constraint chk_nbPlace check (nb_place_dispo >= 0);
```

/\*5 Les tickets réservés mais non payés doivent redevenir disponibles 15min avant le début d'un événement\*/

voir la routine AnnulTicketsImpaye

```
CREATE DEFINER='root'@'localhost' PROCEDURE `AnnulTicketsImpaye`()  
BEGIN  
delete from ticket  
where etat = 'RESERVE'  
and exists(select null from spectacle where idspectacle = t.idspectacle and date_debut -  
15/1440 < now());  
END
```

**remarques:**

- des modifications ont été ajoutées aux tables pour améliorer la base de données 'events';
- le code initial de création des tables synchronisées ne contient pas les triggers et les routines mais on les a ajoutés dans les réponses.
- les données sont pas beaucoup mais sont des événements réels de la calendrier des événements annuel .
- les prix des tickets sont très élevés pour rendre les bénéfices positifs .
- pour la génération des données factice avec fillDB on a réussi à charger les données mais on a rencontré un problème lors de l'importation sur Mysql alors on a opté à créer des données manuellement pour pouvoir tester nos requêtes.

pour la répartition du travail on a travaillé ensemble sur le projet sur des réunions zoom et discord