



# Corrigés des Exercices Système d'Information

Licence Ingénierie et Conception Système d'Information

# Cas 1 : GESTION DES CONSULTATIONS MEDICALES

## Dictionnaire de Données

	Intitulé	Code	Type	Contrainte	Calcul
1	Code Réservation	Cd_Res	S	Identifiant	
2	Date Rendez Vous	D_RDV	D	Not Null	
3	Horaire Rendez Vous	H_RDV	T	Not Null	
4	Id Malade	Id_Mal	S	Identifiant	
5	Nom Malade	Nom_Mal	AV(30)	Not Null	
6	Prénom Malade	Prenom_Mal	AV(30)	Not Null	
7	Date Naissance Malade	DN_Mal	D	Not Null	
8	Id Symptôme	Id_Sympt	S	Identifiant	
9	Description Symptôme	Desc_Sympt	AV(30)	Not Null, Unique	
10	Id Maladie	Id_Mal	S	Identifiant	
11	Description Maladie	Desc_Mal	AV(30)	Not Null, Unique	
12	Id Consultation	Id_Consult	S	Identifiant	
13	Date Consultation	D_Consult	D	Not Null	
14	Heure Consultation	H_Consult	T	Not Null	
15	Prix Consultation	Prix_Consult	N(8,2)	Not Null	

## Matrice de dépendance Fonctionnelle

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1														
2	1	1													
3	1		1												
4				1											
5				1	1										
6				1		1									
7				1			1								
8								1							
9								1	1						
10										1					
11										1	1				
12												1			
13												1	1		
14												1		1	
15												1			1

MALADIE			
<u>Id_Mali</u>	<pi>	Entier	<O>
Desc_Mali		Caractère variable (30)	<O>
Identifiant_1	<pi>		

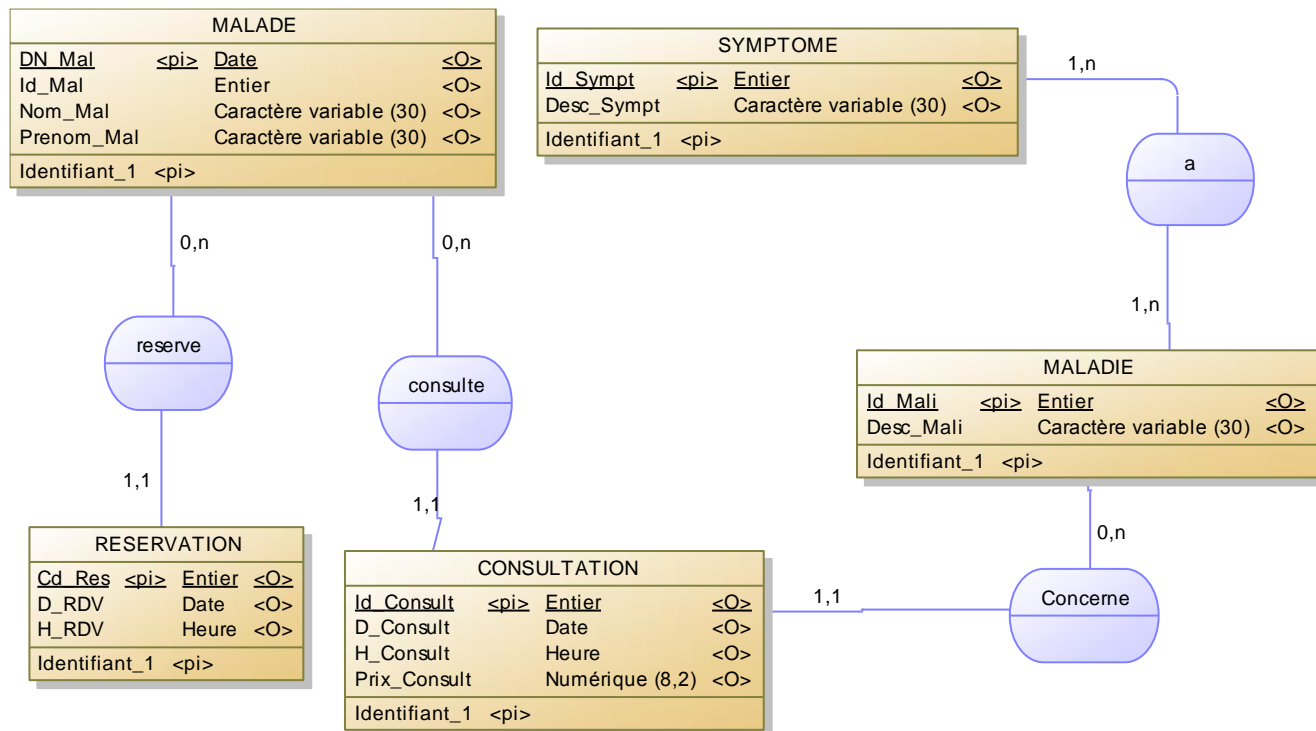
SYMPTOME			
<u>Id_Sympt</u>	<pi>	Entier	<O>
Desc_Sympt		Caractère variable (30)	<O>
Identifiant_1	<pi>		

RESERVATION			
<u>Cd_Res</u>	<pi>	Entier	<O>
D_RDV		Date	<O>
H_RDV		Heure	<O>
Identifiant_1	<pi>		

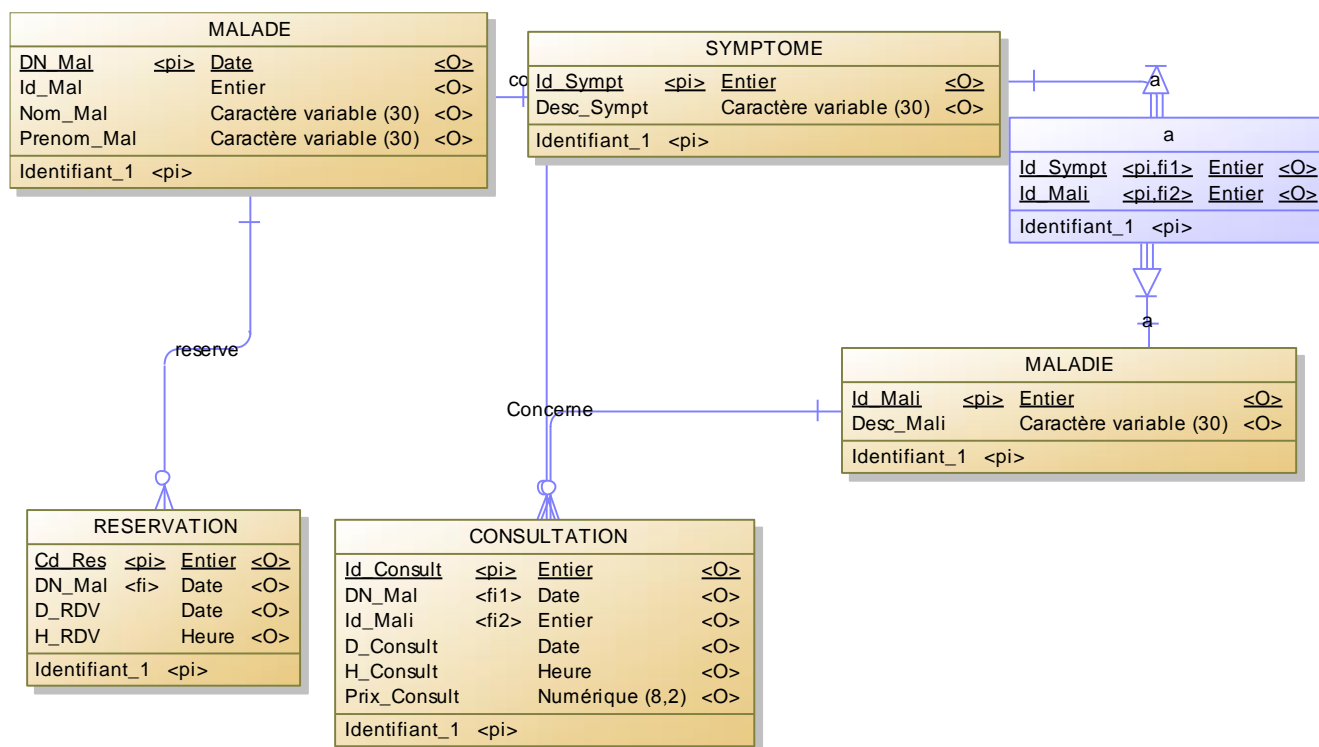
CONSULTATION			
Id_Consult		Entier	
D_Consult		Date	
H_Consult		Heure	
Prix_Consult		Numérique (8,2)	
Identifiant_1	<pi>		

MALADE			
<u>DN_Mal</u>	<pi>	Date	<O>
Id_Mal		Entier	<O>
Nom_Mal		Caractère variable (30)	<O>
Prenom_Mal		Caractère variable (30)	<O>
Identifiant_1	<pi>		

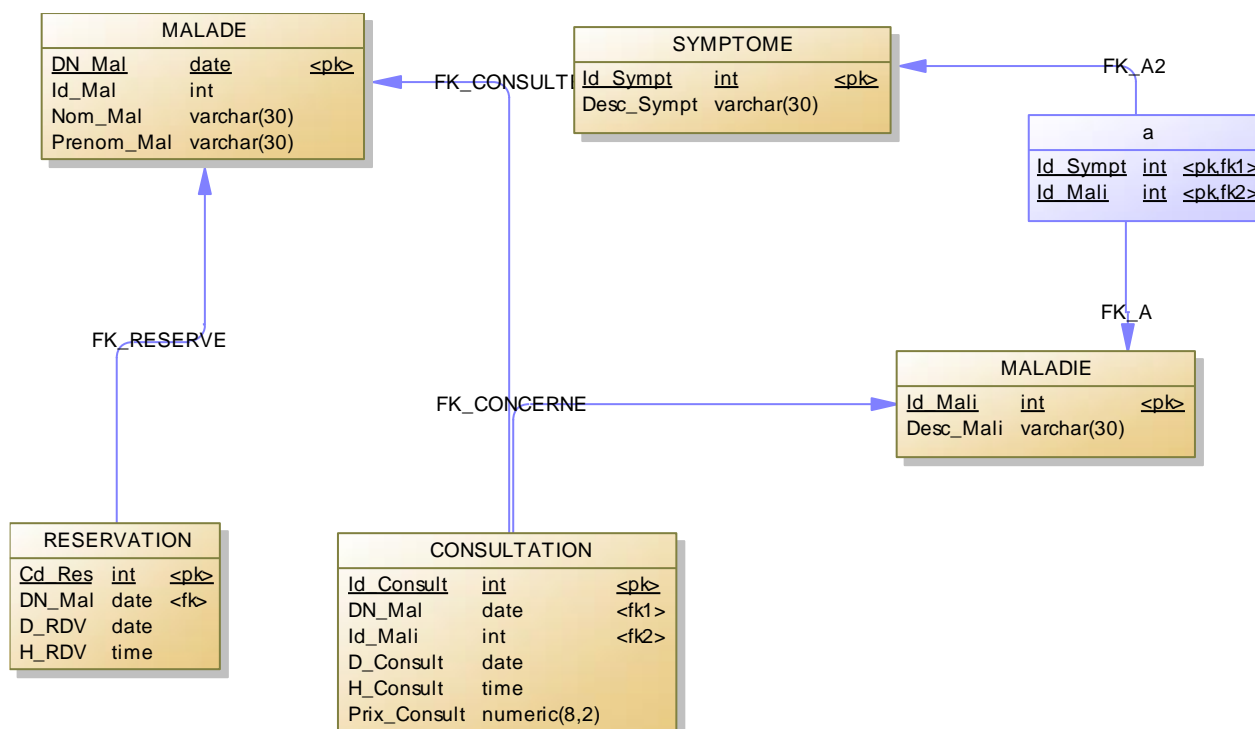
## Modèle Conceptuel de Données



## Modèle Logique de Données



## Modèle Physique de Données



## Script de création de base de données

```

/*=====*/
/* Nom de SGBD : MySQL 5.0 */
/* Date de création : 29/11/2019 22:46:28 */
/*=====*/

```

```

drop table if exists A;
drop table if exists CONSULTATION;
drop table if exists MALADE;
drop table if exists MALADIE;
drop table if exists RESERVATION;
drop table if exists SYMPTOME;

```

```

/*=====*/
/* Table : A                                     */
/*=====*/
create table A
(
    ID_SYMPT      int not null,
    ID_MALI       int not null,
    primary key (ID_SYMPT, ID_MALI)
);

/*=====*/
/* Table : CONSULTATION                         */
/*=====*/
create table CONSULTATION
(
    ID_CONSULT     int not null,
    DN_MAL         date not null,
    ID_MALI        int not null,
    D_CONSULT      date not null,
    H_CONSULT      time not null,
    PRIX_CONSULT   numeric(8,2) not null,
    primary key (ID_CONSULT)
);

/*=====*/
/* Table : MALADE                               */
/*=====*/
create table MALADE
(
    DN_MAL         date not null,
    ID_MAL         int not null,
    NOM_MAL        varchar(30) not null,
    PRENOM_MAL     varchar(30) not null,
    primary key (DN_MAL)
);

/*=====*/
/* Table : MALADIE                             */
/*=====*/
create table MALADIE
(
    ID_MALI        int not null,
    DESC_MALI      varchar(30) not null,
    primary key (ID_MALI)
);

/*=====*/
/* Table : RESERVATION                         */
/*=====*/
create table RESERVATION
(
    CD_RES         int not null,
    DN_MAL         date not null,
    D_RDV          date not null,
    H_RDV          time not null,
    primary key (CD_RES)
);

/*=====*/
/* Table : SYMPTOME                           */
/*=====*/
create table SYMPTOME
(
    ID_SYMPT      int not null,
    DESC_SYMPT    varchar(30) not null,
    primary key (ID_SYMPT)

```

```
);  
  
alter table A add constraint FK_A foreign key (ID_MALI)  
references MALADIE (ID_MALI) on delete restrict on update restrict;  
  
alter table A add constraint FK_A2 foreign key (ID_SYMPT)  
references SYMPTOME (ID_SYMPT) on delete restrict on update restrict;  
  
alter table CONSULTATION add constraint FK_CONCERNE foreign key (ID_MALI)  
references MALADIE (ID_MALI) on delete restrict on update restrict;  
  
alter table CONSULTATION add constraint FK_CONSULTE foreign key (DN_MAL)  
references MALADE (DN_MAL) on delete restrict on update restrict;  
  
alter table RESERVATION add constraint FK_RESERVE foreign key (DN_MAL)  
references MALADE (DN_MAL) on delete restrict on update restrict;
```

# Cas 2 : GESTION DE STOCKS DES PIECES AUTO

## Dictionnaire de Données

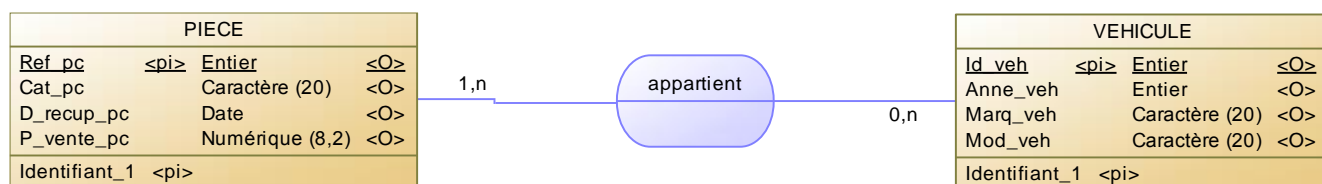
	Intitulé	Code	Type	Contrainte	Calcul
1	Référence pièce	Ref_pc	S	Identifiant	
2	Catégorie pièce	Cat_pc	AV(20)	Not Null	
3	Date récupération pièce	D_Recup_pc	D	Not Null	
4	Prix vente pièce	P_vente_pc	N(8,2)	Not Null	
5	Id vehicule	Id_veh	S	Identifiant	
6	Marque véhicule	Marq_veh	AV(16)	Not Null	
7	Modèle véhicule	Mod_veh	AV(16)	Not Null	
8	Année véhicule	Annee_veh	D	Not Null	

## Matrice de dépendance Fonctionnelle

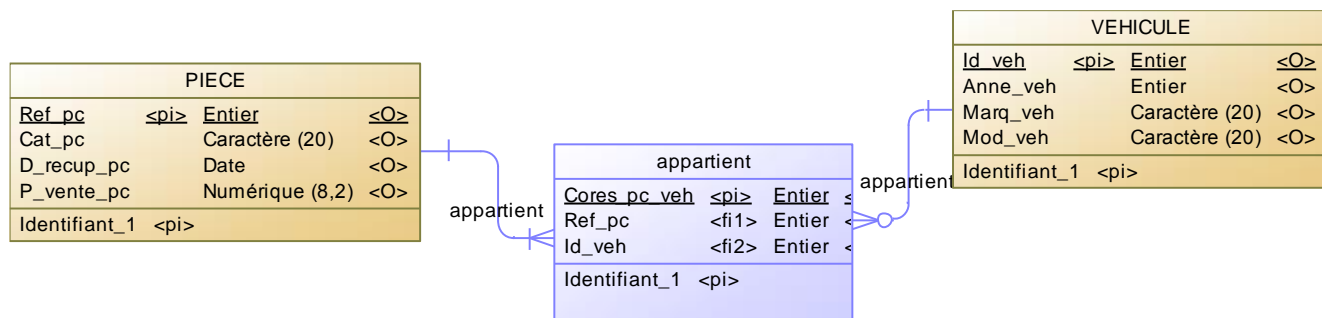
	1	2	3	4	5	6	7	8
1	1							
2	1	1						
3	1		1					
4	1			1				
5					1			
6					1	1		
7					1		1	
8					1			1

PIECE			
<u>Ref_pc</u>	<pi>	Entier	<O>
Cat_pc		Caractère (20)	<O>
D_recup_pc		Date	<O>
P_vente_pc		Numérique (8,2)	<O>
Identifiant_1	<pi>		

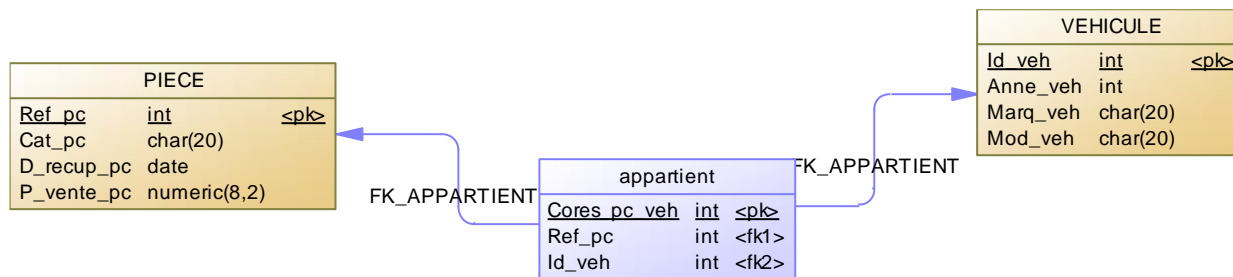
VEHICULE			
<u>Id_veh</u>	<pi>	Entier	<O>
Anne_veh		Entier	<O>
Marq_veh		Caractère (20)	<O>
Mod_veh		Caractère (20)	<O>
Identifiant_1	<pi>		



## Modèle Logique de Données



## Modèle Physique de Données



## Script de création de base de données

```

/*=====*/
/* Nom de SGBD : MySQL 5.0 */
/* Date de création : 30/11/2019 00:08:30 */
/*=====*/

drop table if exists APPARTIENT;
drop table if exists PIECE;
drop table if exists VEHICULE;

/*=====*/
/* Table : APPARTIENT */
/*=====*/
create table APPARTIENT
(
    CORES_PC_VEH int not null,
    REF_PC int not null,
    ID_VEH int not null,
    primary key (CORES_PC_VEH)
);

/*=====*/
/* Table : PIECE */
/*=====*/
create table PIECE
(
    REF_PC int not null,
    CAT_PC char(20) not null,
    D_RECUP_PC date not null,
    P_VENTE_PC numeric(8,2) not null,
    primary key (REF_PC)
);

/*=====*/
/* Table : VEHICULE */
/*=====*/
create table VEHICULE
(
    ID_VEH int not null,
    ANNE_VEH int not null,
    MARQ_VEH char(20) not null,
    MOD_VEH char(20) not null,
    primary key (ID_VEH)
);

alter table APPARTIENT add constraint FK_APPARTIENT foreign key (ID_VEH)
references VEHICULE (ID_VEH) on delete restrict on update restrict;

alter table APPARTIENT add constraint FK_APPARTIENT2 foreign key (REF_PC)
references PIECE (REF_PC) on delete restrict on update restrict;

```



# Cas 3 : GESTION DES ANIMAUX D'UN AQUARIUM

## Dictionnaire de Données

	Intitulé	Code	Type	Contrainte	Calcul
1	Immatricule poisson	Immat_poiss	S	Identifiant	
2	Id soin	Id_soin	S	Identifiant	
3	Date soin	D_soin	D	Not Null	
4	Nature de soin	Nat_soin	AV(20)	Not Null	
5	Id Bassin	Id_bassin	S	Identifiant	
6	Id Pièce	Id_pièce	S	Identifiant	
7	Nom Pièce	Nom	AV(20)	Not Null	
8	Date migration bassin	D_mig_bassin	AV(16)	Not Null	
9	Id Ordre	Id_ordre	S	Identifiant	
10	Description Ordre	Desc_ordre	AV(20)	Not Null	
11	Id Famille	Id_ordre	S	Identifiant	
12	Description Famille	Desc_famille	AV(20)	Not Null	
13	Id Genre	Id_genre	S	Identifiant	
14	Description Genre	Desc_genre	AV(20)	Not Null	
15	Id Espèce	Id_espece	S	Identifiant	
16	Description Espèce	Desc_espece	AV(20)	Not Null	
17	Nom poisson	Nom_poiss	AV(20)	Not null	
18	Nom Bassin	Nom_bassin	AV(20)	Not Null	

## Matrice de dépendance Fonctionnelle

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1																	
2		1																
3			1	1														
4			1		1													
5					1													
6						1												
7							1											
8								1										
9									1									
10									1	1								
11											1							
12											1	1						
13													1					
14													1	1				
15															1			
16															1	1		
17	1																1	
18					1													1

POISSON	
<u>Imm_poiss</u> <pi> Entier	
Nom_poiss	Caractère variable (20)
Identifiant_1	<pi>

SOIN	
<u>D_soin</u> <pi> Date	
Id_soin	Entier
Nat_soin	Caractère variable (20)
Identifiant_1	<pi>

BASSIN	
<u>Id_bassin</u> <pi> Entier	
Nom_bassin	Caractère variable (20)
Identifiant_1	<pi>

PIECE	
<u>Id_piece</u> <pi> Entier	
Nom_piece	Caractère variable (20)
Identifiant_1	<pi>

FAMILLE	
<u>Id_famille</u> <pi> Entier	
Desc_famille	Caractère variable (20)
Identifiant_1	<pi>

ORDRE	
<u>Id_ordre</u> <pi> Entier	
Desc_ordre	Caractère variable (20)
Identifiant_1	<pi>

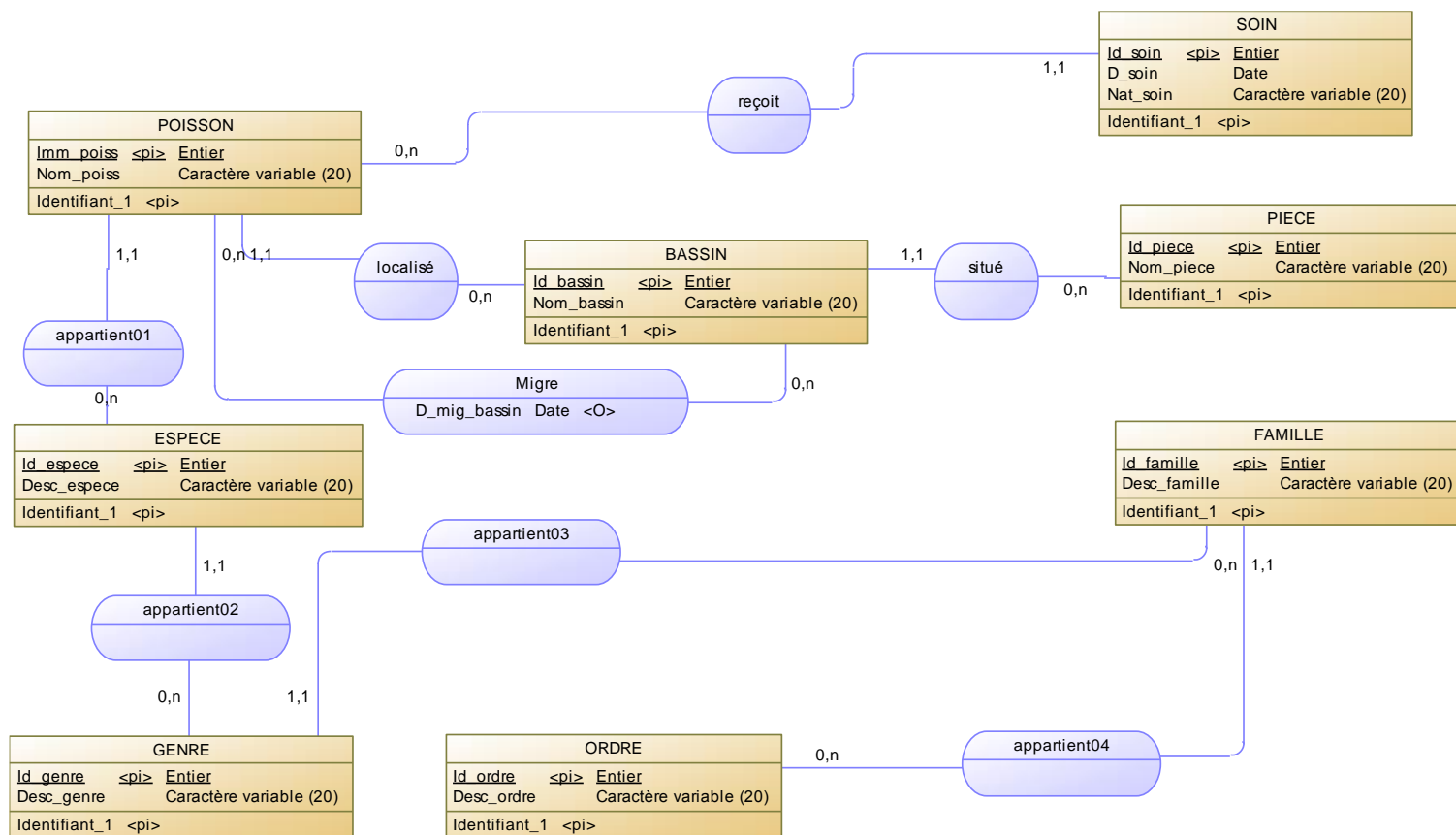
GENRE	
<u>Id_genre</u> <pi> Entier	
Desc_genre	Caractère variable (20)
Identifiant_1	<pi>

ESPECE	
<u>Id_espece</u> <pi> Entier	
Desc_espece	Caractère variable (20)
Identifiant_1	<pi>

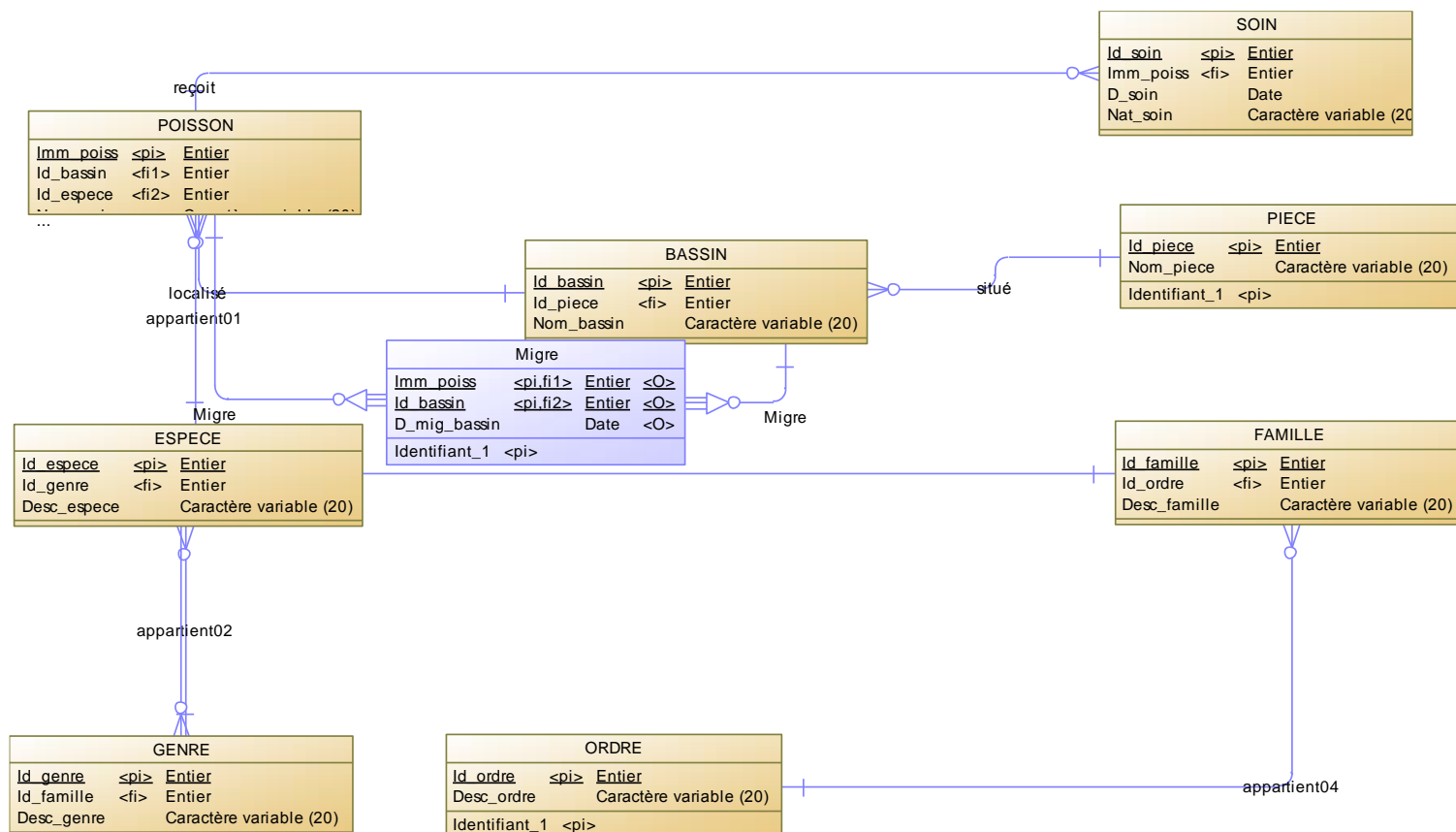
On Remarque que le premier passage n'a pas permis de caser tous les attributs dans les entités, il reste néanmoins un attribut non casé dans les entités (D\_mig\_bassin).

Une deuxième analyse de la matrice de dépendance fonctionnelle, permet de caser cet attribut au niveau de l'association <migre>

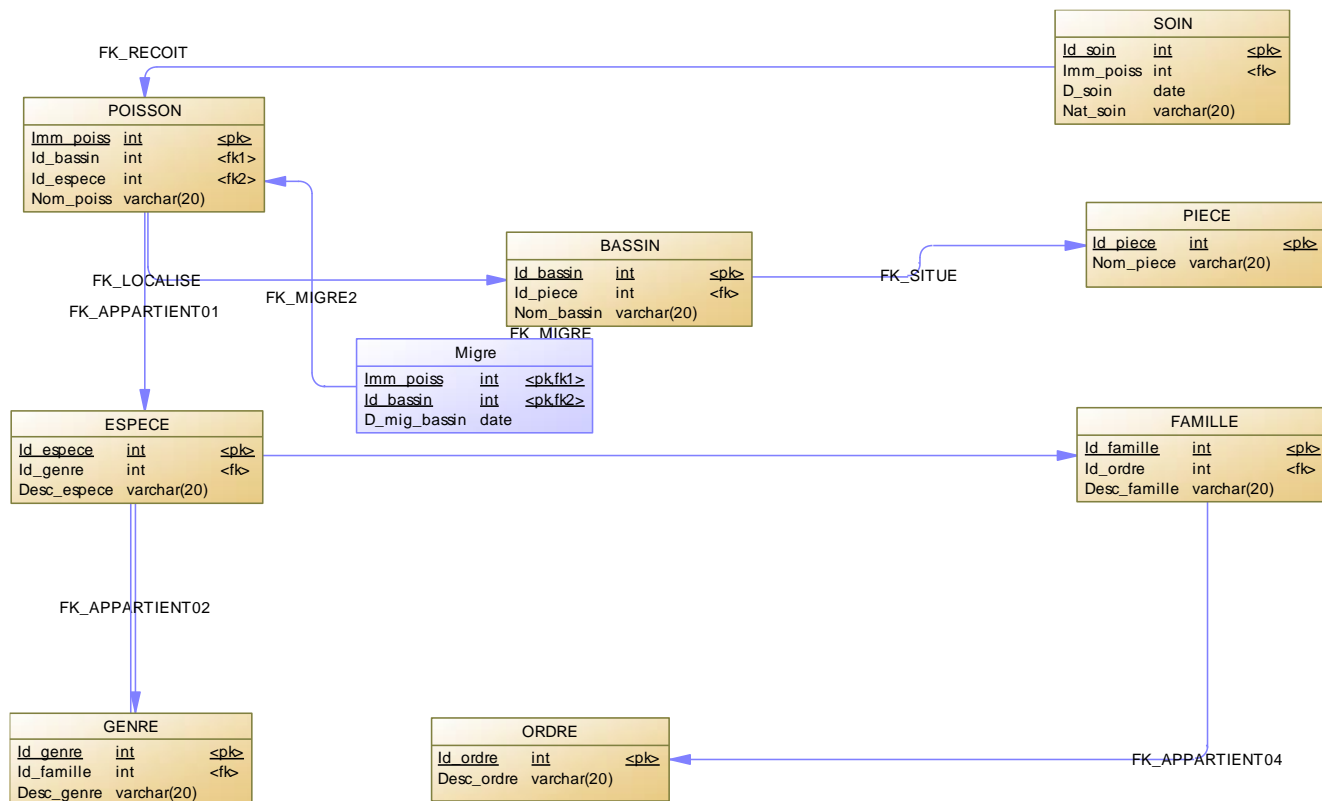
## Modèle conceptuel de Données



## Modèle Logique de Données



## Modèle Physique de Données



## Script de création de base de données

! \* Nom de SGBD : MySQL 5.0  
! \* Date de création : 06/12/2019 16:24:31

\*/  
\*/

drop table if exists BASSIN;  
 drop table if exists ESPECE;  
 drop table if exists FAMILLE;  
 drop table if exists GENRE;  
 drop table if exists MIGRE;  
 drop table if exists ORDRE;  
 drop table if exists PIECE;  
 drop table if exists POISSON;  
 drop table if exists SOIN;

/\* Table : BASSIN \*/

```
create table BASSIN
(
  ID_BASSIN      int not null,
  ID_PIECE       int not null,
  NOM_BASSIN     varchar(20) not null,
  primary key (ID_BASSIN)
);
```

/\* Table : ESPECE \*/

```
create table ESPECE
(
  ID_ESPECE      int not null,
  ID_GENRE       int not null,
  DESC_ESPECE    varchar(20) not null,
  primary key (ID_ESPECE)
);
```

/\* Table : FAMILLE \*/

```
create table FAMILLE
(
  ID_FAMILLE     int not null,
  ID_ORDRE       int not null,
  DESC_FAMILLE   varchar(20) not null,
  primary key (ID_FAMILLE)
);
```

/\* Table : GENRE \*/

```
create table GENRE
(
  ID_GENRE       int not null,
  ID_FAMILLE     int not null,
  DESC_GENRE     varchar(20) not null,
  primary key (ID_GENRE)
);
```

/\* Table : MIGRE \*/

```
create table MIGRE
(
  IMM_POISS      int not null,
  ID_BASSIN      int not null,
  D_MIG_BASSIN   date not null,
  primary key (IMM_POISS, ID_BASSIN)
);
```

/\* Table : ORDRE \*/

```
create table ORDRE
(
  ID_ORDRE       int not null,
```

```
DESC_ORDRE      varchar(20) not null,
primary key (ID_ORDRE)
);
```

*/\* Table : PIECE \*/*

```
create table PIECE
(
  ID_PIECE      int not null,
  NOM_PIECE     varchar(20) not null,
primary key (ID_PIECE)
);
```

*/\* Table : POISSON \*/*

```
create table POISSON
(
  IMM_POISS     int not null,
  ID_BASSIN     int not null,
  ID_ESPECE     int not null,
  NOM_POISS     varchar(20) not null,
primary key (IMM_POISS)
);
```

*/\* Table : SOIN \*/*

```
create table SOIN
(
  ID_SOIN       int not null,
  IMM_POISS     int not null,
  D_SOIN        date not null,
  NAT_SOIN      varchar(20) not null,
primary key (ID_SOIN)
);
```

```
alter table BASSIN add constraint FK_SITUE foreign key (ID_PIECE)
references PIECE (ID_PIECE) on delete restrict on update restrict;
alter table ESPECE add constraint FK_APPARTIENT02 foreign key (ID_GENRE)
references GENRE (ID_GENRE) on delete restrict on update restrict;
alter table FAMILLE add constraint FK_APPARTIENT04 foreign key (ID_ORDRE)
references ORDRE (ID_ORDRE) on delete restrict on update restrict;
alter table GENRE add constraint FK_APPARTIENT03 foreign key (ID_FAMILLE)
references FAMILLE (ID_FAMILLE) on delete restrict on update restrict;
alter table MIGRE add constraint FK_MIGRE foreign key (ID_BASSIN)
references BASSIN (ID_BASSIN) on delete restrict on update restrict;
alter table MIGRE add constraint FK_MIGRE2 foreign key (IMM_POISS)
references POISSON (IMM_POISS) on delete restrict on update restrict;
alter table POISSON add constraint FK_APPARTIENT01 foreign key (ID_ESPECE)
references ESPECE (ID_ESPECE) on delete restrict on update restrict;
alter table POISSON add constraint FK_LOCALISE foreign key (ID_BASSIN)
references BASSIN (ID_BASSIN) on delete restrict on update restrict;
alter table SOIN add constraint FK_RECOIT foreign key (IMM_POISS)
references POISSON (IMM_POISS) on delete restrict on update restrict;
```