

# RPG DnD DATABASE

## Data Dictionary

2023-10-10





## Table of contents

MIAGE@localhost .....	8
1. HERITAGE PERSONNAGE .....	9
1.1. Tables .....	10
1.1.1. Table: ALLIE .....	10
1.1.2. Table: CLASSE .....	11
1.1.3. Table: CLASSE_RACE .....	12
1.1.4. Table: COMBAT .....	13
1.1.5. Table: MONSTRE .....	14
1.1.6. Table: PERSONNAGE .....	16
1.1.7. Table: PJ .....	19
1.1.8. Table: PNJ .....	20
1.1.9. Table: RACE .....	21
1.2. Views .....	22
1.2.1. View: V_ALLIE .....	22
1.2.2. View: V_MONSTRE .....	24
1.2.3. View: V_PJ .....	26
1.2.4. View: V_PNJ .....	28
2. HERITAGE SORT .....	32
2.1. Tables .....	33
2.1.1. Table: BUFF .....	33
2.1.2. Table: BUFF_PERSONNAGE .....	34
2.1.3. Table: CLASSE_COMPETENCE .....	35
2.1.4. Table: COMPETENCE .....	36
2.1.5. Table: ECOLEMAGIE .....	37
2.1.6. Table: GRIMOIRE .....	38
2.1.7. Table: POUVOIR .....	39
2.1.8. Table: RACE_RACIAL .....	40
2.1.9. Table: RACIAL .....	41
2.1.10. Table: SORT .....	42
2.2. Views .....	45
2.2.1. View: V_BUFF .....	45
2.2.2. View: V_COMPETENCE .....	47
2.2.3. View: V_CONSUMMABLE .....	49
2.2.4. View: V_POUVOIR .....	51
2.2.5. View: V_RACIAL .....	53
2.2.6. View: V_RESSOURCE .....	55
3. HERITAGE ITEM .....	58
3.1. Tables .....	59
3.1.1. Table: ARME .....	59
3.1.2. Table: ARME_TYPEARME .....	60
3.1.3. Table: ARMURE .....	61
3.1.4. Table: ARMURE_TYPEARMURE .....	62
3.1.5. Table: CLASSE_TYPEARME .....	63
3.1.6. Table: CLASSE_TYPEARMURE .....	64
3.1.7. Table: CONSUMMABLE .....	65

3.1.8. Table: DIVERS .....	66
3.1.9. Table: EVENEMENT_ITEM .....	67
3.1.10. Table: INVENTAIRE .....	68
3.1.11. Table: ITEM .....	69
3.1.12. Table: ITEM_RECETTE .....	71
3.1.13. Table: ITEMQUETE .....	72
3.1.14. Table: ITEMQUETE_QUETE .....	73
3.1.15. Table: METIER_OUTIL .....	74
3.1.16. Table: OUTIL .....	75
3.1.17. Table: RESSOURCE .....	76
3.1.18. Table: TYPEARME .....	77
3.1.19. Table: TYPEARMURE .....	78
3.1.20. Table: TYPECONSUMMABLE .....	79
3.1.21. Table: TYPERESSOURCE .....	80
3.2. Views .....	81
3.2.1. View: V_ARME .....	81
3.2.2. View: V_ARMURE .....	83
3.2.3. View: V_CONSUMMABLE .....	85
3.2.4. View: V_DIVERS .....	87
3.2.5. View: V_ITEMQUETE .....	89
3.2.6. View: V_OUTIL .....	91
3.2.7. View: V_RESSOURCE .....	93
4. Other .....	96
4.1. Tables .....	96
4.1.1. Table: APPARENCE .....	96
4.1.2. Table: APPARENCE_CLASSE .....	99
4.1.3. Table: APPARENCE_EVENEMENT .....	100
4.1.4. Table: APPARENCE_FACTION .....	101
4.1.5. Table: APPARENCE_HAUTFAIT .....	102
4.1.6. Table: APPARENCE_ITEM .....	103
4.1.7. Table: APPARENCE_LIEUX .....	104
4.1.8. Table: APPARENCE_METIER .....	105
4.1.9. Table: APPARENCE_PERSONNAGE .....	106
4.1.10. Table: APPARENCE_QUETE .....	107
4.1.11. Table: APPARENCE_RACE .....	108
4.1.12. Table: APPARENCE_SORT .....	109
4.1.13. Table: APPARENCE_TITRE .....	110
4.1.14. Table: APPARENCEPOSSIBLE_CLASSE .....	111
4.1.15. Table: APPARENCEPOSSIBLE_EVENEMENT .....	112
4.1.16. Table: APPARENCEPOSSIBLE_FACTION .....	113
4.1.17. Table: APPARENCEPOSSIBLE_HAUTFAIT .....	114
4.1.18. Table: APPARENCEPOSSIBLE_ITEM .....	115
4.1.19. Table: APPARENCEPOSSIBLE_LIEUX .....	116
4.1.20. Table: APPARENCEPOSSIBLE_METIER .....	117
4.1.21. Table: APPARENCEPOSSIBLE_PERSONNAGE .....	118
4.1.22. Table: APPARENCEPOSSIBLE_QUETE .....	119

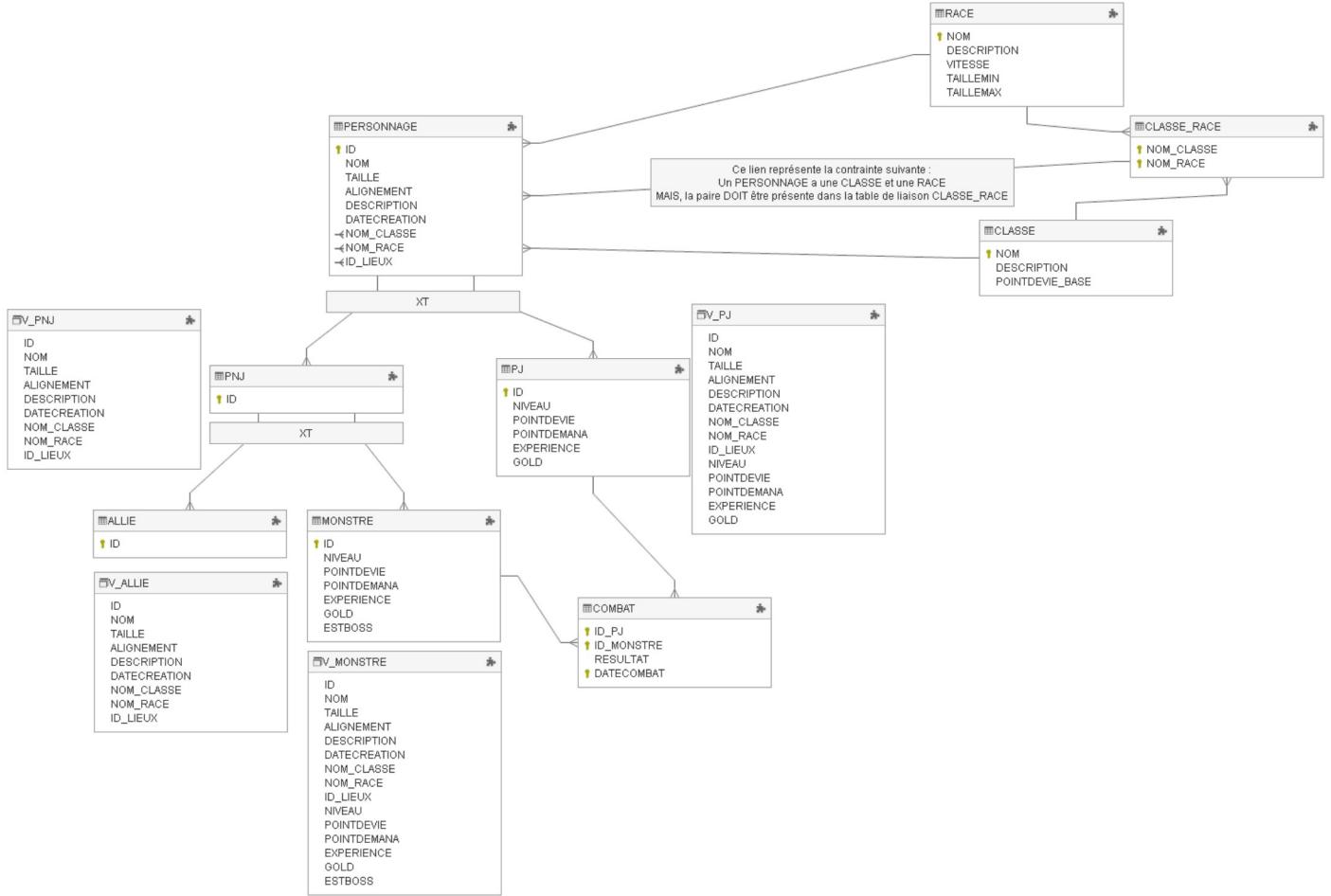
4.1.23. Table: APPARENCEPOSSIBLE_RACE .....	120
4.1.24. Table: APPARENCEPOSSIBLE_SORT .....	121
4.1.25. Table: APPARENCEPOSSIBLE_TITRE .....	122
4.1.26. Table: CALENDRIER_EVENEMENT .....	123
4.1.27. Table: EVENEMENT .....	124
4.1.28. Table: EVENEMENT_MONSTRE .....	126
4.1.29. Table: EVENEMENT_QUETE .....	127
4.1.30. Table: FACTION .....	128
4.1.31. Table: FACTION_PERSONNAGE .....	129
4.1.32. Table: HAUTFAIT .....	130
4.1.33. Table: HAUTFAIT_PERSONNAGE .....	131
4.1.34. Table: LANGUE .....	132
4.1.35. Table: LANGUE_PERSONNAGE .....	133
4.1.36. Table: LANGUE_RACE .....	134
4.1.37. Table: LIEUX .....	135
4.1.38. Table: METIER .....	137
4.1.39. Table: METIER_PERSONNAGE .....	139
4.1.40. Table: METIER_RECETTE .....	140
4.1.41. Table: METIER_TYPERESSOURCE .....	141
4.1.42. Table: MONSTRE_TYPEMONSTRE .....	142
4.1.43. Table: PERSONNAGE_QUETE .....	143
4.1.44. Table: PERSONNAGE_QUETEJOURNALIERE .....	144
4.1.45. Table: PERSONNAGE_RECETTE .....	145
4.1.46. Table: PERSONNAGE_TITRE .....	146
4.1.47. Table: QUETE .....	147
4.1.48. Table: RECETTE .....	149
4.1.49. Table: SUITEQUETE .....	151
4.1.50. Table: TITRE .....	152
4.1.51. Table: TYPELIEUX .....	153
4.1.52. Table: TYPEMONSTRE .....	154
4.1.53. Table: TYPEQUETE .....	155
4.2. Procedures .....	156
4.2.1. Procedure: CLEANALL .....	156

## Legend

- 🔑 Primary key
- 🔑 Primary key disabled
- 🔑 User-defined primary key
- 🔑 Unique key
- 🔑 Unique key disabled
- 🔑 User-defined unique key
- ⚡ Active trigger
- ⚡ Disabled trigger
- Many to one relationship
- User-defined many to one relationship
- ← One to many relationship
- ← User-defined one to many relationship
- ↔ Many to many relationship
- ↔ User-defined many to many relationship
- One to one relationship
- User-defined one to one relationship
- ↗ Input
- ↘ Output
- ↗ Input/Output
- ↗ Uses dependency
- ↗ User-defined uses dependency
- ↘ Used by dependency
- ↘ User-defined used by dependency

MIAGE@localhost

# 1. HERITAGE PERSONNAGE



La table PERSONNAGE est la grand-mère.

Elle possède deux filles :

- PNJ
- PJ

PNJ est elle-même la table mère de deux filles :

- ALLIE
- MONSTRE

Cela permet des traitements différents sur les différentes filles/mère/grand-mère.

Pour représenter cet héritage en base on crée des vues pour chaque enfant.

Les vues réunissent les data des mères et des filles.

On ajoute des TRIGGER INSTEAD OF pour les INSERT, UPDATE, et DELETE.

Et on fait les I, U et D sur les views et non sur les tables directement.

De ce fait un INSERT sur une petite-fille aura des INSERT en chaîne jusqu'à la doyenne de la famille.

## 1.1. Tables

### 1.1.1. Table: ALLIE

#### Columns

	Name	Data type	Description / Attributes
 	ID	NUMBER	<b>References:</b> PNJ

#### Links to

	Table	Join	Title / Name / Description
	PNJ	ALLIED = PNJID	FK_PERSONNAGE_ALLIE

#### Unique keys

	Columns	Name / Description
	ID	SYS_C0040573

#### Uses

	Name
	ALLIE
	PNJ

#### Used By

	Name
	ALLIE
	V_ALLIE
	DELETE_ALLIE_TRIGGER
	NEW_ALLIE_TRIGGER

## 1.1.2. Table: CLASSE

### Columns

		Name	Data type	Description / Attributes
█	█	NOM	VARCHAR2(255 BYTE)	
█		DESCRIPTION	CLOB	<b>Nullable</b>
█		POINTDEVIE_BASE	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 1000

### Linked from

Table		Join	Title / Name / Description
→	APPARENCE_CLASSE	<b>CLASSENOM</b> = APPARENCE_CLASSENOM_CLASSE	FK_CLASSE_AC
→	APPARENCEPOSSIBLE_CLASSE	<b>CLASSENOM</b> = APPARENCEPOSSIBLE_CLASSENOM_CLASSE	FK_CLASSE_APC
→	CLASSE_COMPETENCE	<b>CLASSENOM</b> = CLASSE_COMPETENCENOM_CLASSE	FK_CLASSE_CC
→	CLASSE_RACE	<b>CLASSENOM</b> = CLASSE_RACENOM_CLASSE	SYS_C0040558
→	CLASSE_TYPEARME	<b>CLASSENOM</b> = CLASSE_TYPEARMENOM_CLASSE	FK_CLASSE_CLASSE_TYPEARME
→	CLASSE_TYPEARMURE	<b>CLASSENOM</b> = CLASSE_TYPEARMURENOM_CLASSE	FK_CLASSE_CT
→	PERSONNAGE	<b>CLASSENOM</b> = PERSONNAGENOM_CLASSE	FK_CLASSE_PERSONNAGE

### Unique keys

Columns		Name / Description
█	NOM	SYS_C0040503

### Used By

Name	
█	CLASSE
→	APPARENCE_CLASSE
→	APPARENCEPOSSIBLE_CLASSE
→	CLASSE_COMPETENCE
→	CLASSE_RACE
→	CLASSE_TYPEARME
→	CLASSE_TYPEARMURE
→	PERSONNAGE

### 1.1.3. Table: CLASSE\_RACE

#### Columns

Name		Data type	Description / Attributes
目	🔑 NOM_CLASSE	VARCHAR2(255 BYTE)	References: CLASSE
目	🔑 NOM_RACE	VARCHAR2(255 BYTE)	References: RACE

#### Links to

Table		Join	Title / Name / Description
→	CLASSE	CLASSE_RACENOM_CLASSE = CLASSENOM	SYS_C0040558
→	RACE	CLASSE_RACENOM_RACE = RACENOM	SYS_C0040559

#### Linked from

Table		Join	Title / Name / Description
←	PERSONNAGE	CLASSE_RACENOM_CLASSE = PERSONNAGENOM_CLASSE, CLASSE_RACENOM_RACE = PERSONNAGENOM_RACE	FK_CALSSERACE_PERSONNAGE

#### Unique keys

Columns		Name / Description
🔑	NOM_CLASSE, NOM_RACE	SYS_C0040557

#### Uses

Name	
█	CLASSE_RACE
→	CLASSE
→	RACE

#### Used By

Name	
█	CLASSE_RACE
←	PERSONNAGE

## 1.1.4. Table: COMBAT

### Columns

		Name	Data type	Description / Attributes
█	█	ID_PJ	NUMBER(5, 0)	<b>References:</b> PJ
█	█	ID_MONSTRE	NUMBER(5, 0)	<b>References:</b> MONSTRE
█		RESULTAT	VARCHAR2(255 BYTE)	<b>Nullable</b> <b>Default:</b> 'victoire'
█	█	DATECOMBAT	DATE	<b>Default:</b> SYSDATE

### Links to

Table		Join	Title / Name / Description
→	MONSTRE	<b>COMBAT</b> ID_MONSTRE = MONSTREID	FK_MONSTRE_COMBAT
→	PJ	<b>COMBAT</b> ID_PJ = PJID	FK_PJ_COMBAT

### Unique keys

Columns		Name / Description
█	ID_PJ, ID_MONSTRE, DATECOMBAT	SYS_C0040634

### Uses

		Name
█	COMBAT	
→	MONSTRE	
→	PJ	

## 1.1.5. Table: MONSTRE

### Columns

		Name	Data type	Description / Attributes
█	█	ID	NUMBER	<b>References:</b> PNJ
█		NIVEAU	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 1
█		POINTDEVIE	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 1000
█		POINTDEMANA	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 1000
█		EXPERIENCE	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 0
█		GOLD	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 0
█		ESTBOSS	NUMBER(1, 0)	<b>Nullable</b> <b>Default:</b> 0

### Links to

	Table	Join	Title / Name / Description
→	PNJ	MONSTREID = PNJID	FK_PERSONNAGE_MONSTRE

### Linked from

	Table	Join	Title / Name / Description
←	COMBAT	MONSTREID = COMBATID_MONSTRE	FK_MONSTRE_COMBAT
←	EVENEMENT_MONSTRE	MONSTREID = EVENEMENT_MONSTREID_MONSTRE	FK_MONSTRE_EVENEMENT_MONSTRE
←	MONSTRE_TYPEMONSTRE	MONSTREID = MONSTRE_TYPEMONSTREID_MONSTRE	FK_MONSTRE_MT

### Unique keys

	Columns	Name / Description
█	ID	SYS_C0040575

### Uses

	Name
█	MONSTRE
→	PNJ

### Used By

	Name
█	MONSTRE
█	V_MONSTRE
⚡	DELETE_MONSTRE_TRIGGER
⚡	NEW_MONSTRE_TRIGGER

Name
⚡ UPDATE_MONSTRE_TRIGGER
→ COMBAT
→ EVENEMENT_MONSTRE
→ MONSTRE_TYPEMONSTRE

## 1.1.6. Table: PERSONNAGE

### Columns

		Name	Data type	Description / Attributes
█	█	ID	NUMBER	
█		NOM	VARCHAR2(255 BYTE)	
█		TAILLE	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 180
█		ALIGNEMENT	VARCHAR2(255 BYTE)	<b>Nullable</b>
█		DESCRIPTION	CLOB	<b>Nullable</b>
█		DATECREATION	DATE	<b>Nullable</b> <b>Default:</b> SYSDATE
█		NOM_CLASSE	VARCHAR2(255 BYTE)	<b>References:</b> CLASSE_RACE, CLASSE
█		NOM_RACE	VARCHAR2(255 BYTE)	<b>References:</b> CLASSE_RACE, RACE
█		ID_LIEUX	NUMBER(5, 0)	<b>References:</b> LIEUX

### Links to

	Table	Join	Title / Name / Description
→	CLASSE	<b>PERSONNAGENOM_CLASSE</b> = CLASSENOM	FK_CLASSE_PERSONNAGE
→	CLASSE_RACE	<b>PERSONNAGENOM_CLASSE</b> = CLASSE_RACENOM_CLASSE, <b>PERSONNAGENOM_RACE</b> = CLASSE_RACENOM_RACE	FK_CALSSERACE_PERSONNAGE
→	LIEUX	<b>PERSONNAGEID_LIEUX</b> = LIEUXID	FK_LIEUX_PERSONNAGE
→	RACE	<b>PERSONNAGENOM_RACE</b> = RACENOM	FK_RACE_PERSONNAGE

### Linked from

	Table	Join	Title / Name / Description
←	APPARENCE_PERSONNAGE	<b>PERSONNAGEID</b> = APPARENCE_PERSONNAGEID_PERSONNAGE	FK_PERSONNAGE_AP
←	APPARENCEPOSSIBLE_PERSONNAGE	<b>PERSONNAGEID</b> = APPARENCEPOSSIBLE_PERSONNAGEID_PERSONNAGE	FK_PERSONNAGE_APP
←	BUFF_PERSONNAGE	<b>PERSONNAGEID</b> = BUFF_PERSONNAGEID_PERSONNAGE	FK_PERSONNAGE_BUFF_PERSONNAGE
←	FACTION_PERSONNAGE	<b>PERSONNAGEID</b> = FACTION_PERSONNAGEID_PERSONNAGE	FK_PERSONNAGE_FP
←	GRIMOIRE	<b>PERSONNAGEID</b> = GRIMOIREID_PERSONNAGE	FK_PERSONNAGE_GRIMOIRE
←	HAUTFAIT_PERSONNAGE	<b>PERSONNAGEID</b> = HAUTFAIT_PERSONNAGEID_PERSONNAGE	FK_PERSONNAGE_HP
←	INVENTAIRE	<b>PERSONNAGEID</b> = INVENTAIREID_PERSONNAGE	FK_PERSONNAGE_INVENTAIRE

Table		Join	Title / Name / Description
←	LANGUE_PERSONNAGE	<b>PERSONNAGEID</b> = LANGUE_PERSONNAGEID_PERSONNAGE	FK_PERSONNAGE_LP
←	METIER_PERSONNAGE	<b>PERSONNAGEID</b> = METIER_PERSONNAGEID_PERSONNAGE	FK_PERSONNAGE_MP
←	PERSONNAGE_QUETE	<b>PERSONNAGEID</b> = PERSONNAGE_QUETEID_PERSONNAGE	FK_PERSONNAGE_PERSONNAGE_QUETE
←	PERSONNAGE_QUETEJOURNALIERE	<b>PERSONNAGEID</b> = PERSONNAGE_QUETEJOURNALIEREID_PERSONNAGE	FK_PERSONNAGE_PQJ
←	PERSONNAGE_RECETTE	<b>PERSONNAGEID</b> = PERSONNAGE_RECETTEID_PERSONNAGE	FK_PERSONNAGE_PR
←	PERSONNAGE_TITRE	<b>PERSONNAGEID</b> = PERSONNAGE_TITREID_PERSONNAGE	FK_PERSONNAGE_PT
←	PJ	<b>PERSONNAGEID</b> = PJID	FK_PERSONNAGE_PJ
←	PNJ	<b>PERSONNAGEID</b> = PNJID	FK_PERSONNAGE_PNJ

## Unique keys

Columns		Name / Description
🔑	ID	SYS_C0040564

## Triggers

	Name	When	Description
⚡	NEW_PERSONNAGE_TRIGGER	Before Insert	
<pre>TRIGGER New_Personnage_trigger BEFORE INSERT ON PERSONNAGE FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := PERSONNAGE_seq.NEXTVAL; END;</pre>			

## Uses

	Name
grid	PERSONNAGE
→	CLASSE
→	CLASSE_RACE
→	LIEUX
→	RACE
⚡	NEW_PERSONNAGE_TRIGGER
grid	PERSONNAGE

## Used By

	Name
grid	PERSONNAGE

Name
V_PJ
V_PNJ
⚡ DELETE_PJ_TRIGGER
⚡ DELETE_PNJ_TRIGGER
⚡ NEW_PERSONNAGE_TRIGGER
⚡ NEW_PJ_TRIGGER
⚡ NEW_PNJ_TRIGGER
⚡ UPDATE_PJ_TRIGGER
⚡ UPDATE_PNJ_TRIGGER
→ APPARENCE_PERSONNAGE
→ APPARENCEPOSSIBLE_PERSONNAGE
→ BUFF_PERSONNAGE
→ FACTION_PERSONNAGE
→ GRIMOIRE
→ HAUTFAIT_PERSONNAGE
→ INVENTAIRE
→ LANGUE_PERSONNAGE
→ METIER_PERSONNAGE
→ PERSONNAGE_QUETE
→ PERSONNAGE_QUETEJOURNALIERE
→ PERSONNAGE_RECETTE
→ PERSONNAGE_TITRE
→ PJ
→ PNJ

## 1.1.7. Table: PJ

### Columns

		Name	Data type	Description / Attributes
■	🔑	ID	NUMBER	<b>References:</b> PERSONNAGE
■		NIVEAU	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 1
■		POINTDEVIE	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 1000
■		POINTDEMANA	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 1000
■		EXPERIENCE	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 0
■		GOLD	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 10

### Links to

	Table	Join	Title / Name / Description
→	PERSONNAGE	PJID = PERSONNAGEID	FK_PERSONNAGE_PJ

### Linked from

	Table	Join	Title / Name / Description
←	COMBAT	PJID = COMBATID_PJ	FK_PJ_COMBAT

### Unique keys

	Columns	Name / Description
🔑	ID	SYS_C0040569

### Uses

	Name
█ PJ	
→ PERSONNAGE	

### Used By

	Name
█ PJ	
↳ V_PJ	
⚡ DELETE_PJ_TRIGGER	
⚡ NEW_PJ_TRIGGER	
⚡ UPDATE_PJ_TRIGGER	
← COMBAT	

## 1.1.8. Table: PNJ

### Columns

	Name	Data type	Description / Attributes
	ID	NUMBER	<b>References:</b> PERSONNAGE

### Links to

	Table	Join	Title / Name / Description
	PERSONNAGE	PNJID = PERSONNAGEID	FK_PERSONNAGE_PNJ

### Linked from

	Table	Join	Title / Name / Description
	ALLIE	PNJID = ALLIEDID	FK_PERSONNAGE_ALLIE
	MONSTRE	PNJID = MONSTREID	FK_PERSONNAGE_MONSTRE

### Unique keys

	Columns	Name / Description
	ID	SYS_C0040571

### Uses

	Name
	PNJ
	PERSONNAGE

### Used By

	Name
	PNJ
	V_PNJ
	DELETE_PNJ_TRIGGER
	NEW_PNJ_TRIGGER
	ALLIE
	MONSTRE

## 1.1.9. Table: RACE

### Columns

		Name	Data type	Description / Attributes
■	🔑	NOM	VARCHAR2(255 BYTE)	
■		DESCRIPTION	CLOB	Nullable
■		VITESSE	NUMBER(5, 0)	Nullable Default: 15
■		TAILLEMIN	NUMBER(5, 0)	Nullable Default: 90
■		TAILLEMAX	NUMBER(5, 0)	Nullable Default: 250

### Linked from

Table		Join	Title / Name / Description
→	APPARENCE_RACE	RACENOM = APPARENCE_RACENOM_RACE	FK_RACE_AR
→	APPARENCEPOSSIBLE_RACE	RACENOM = APPARENCEPOSSIBLE_RACENOM_RACE	FK_RACE_APR
→	CLASSE_RACE	RACENOM = CLASSE_RACENOM_RACE	SYS_C0040559
→	LANGUE_RACE	RACENOM = LANGUE_RACENOM_RACE	FK_RACE_LR
→	PERSONNAGE	RACENOM = PERSONNAGENOM_RACE	FK_RACE_PERSONNAGE
→	RACE_RACIAL	RACENOM = RACE_RACIALNOM_RACE	FK_RACE_RACE_RACIAL

### Unique keys

Columns		Name / Description
🔑	NOM	SYS_C0040504

### Used By

		Name
grid	RACE	
→	APPARENCE_RACE	
→	APPARENCEPOSSIBLE_RACE	
→	CLASSE_RACE	
→	LANGUE_RACE	
→	PERSONNAGE	
→	RACE_RACIAL	

## 1.2. Views

### 1.2.1. View: V\_ALLIE

#### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	TAILLE	NUMBER(5, 0)	Nullable
█	ALIGNEMENT	VARCHAR2(255 BYTE)	Nullable
█	DESCRIPTION	CLOB	Nullable
█	DATECREATION	DATE	Nullable
█	NOM_CLASSE	VARCHAR2(255 BYTE)	
█	NOM_RACE	VARCHAR2(255 BYTE)	
█	ID_LIEUX	NUMBER(5, 0)	

#### Triggers

	Name	When	Description
⚡	DELETE_ALLIE_TRIGGER	Instead Of Delete	
	TRIGGER Delete_Allie_trigger INSTEAD OF DELETE ON V_ALLIE FOR EACH ROW BEGIN -- Suppression de l'allie DELETE FROM ALLIE WHERE Id = :OLD.Id;  -- Suppression du pnj DELETE FROM V_PNJ WHERE Id = :OLD.Id; END;		
⚡	NEW_ALLIE_TRIGGER	Instead Of Insert	
	TRIGGER New_Allie_trigger INSTEAD OF INSERT ON V_ALLIE FOR EACH ROW DECLARE New_Id NUMBER; BEGIN -- Nouveau PNJ INSERT INTO V_PNJ(Nom, Taille, Alignement, Description, DateCreation, Nom_Classe, Nom_Race, Id_Lieux) VALUES (:NEW.Nom, :NEW.Taille, :NEW.Alignement, :NEW.Description, :NEW.DateCreation, :NEW.Nom_Classe, :NEW.Nom_Race, :NEW.Id_Lieux);  -- Récupérer l'ID généré SELECT MAX(Id) INTO New_Id FROM V_PNJ;  -- Nouveau ALLIE INSERT INTO ALLIE(Id) VALUES (New_Id); END;		
⚡	UPDATE_ALLIE_TRIGGER	Instead Of Update	

```

TRIGGER Update_Allie_trigger
INSTEAD OF UPDATE ON V_ALLIE
FOR EACH ROW
BEGIN
    -- Mise à jour du personnage
    UPDATE V_PNJ
    SET Nom = :NEW.Nom,
        Taille = :NEW.Taille,
        Alignement = :NEW.Alignement,
        Description = :NEW.Description,
        DateCreation = :NEW.DateCreation,
        Nom_Classe = :NEW.Nom_Classe,
        Nom_Race = :NEW.Nom_Race,
        Id_Lieux = :NEW.Id_Lieux
    WHERE Id = :NEW.Id;
END;

```

## Uses

	Name
📅 V_ALLIE	
📅 ALLIE	
📅 V_PNJ	
⚡ DELETE_ALLIE_TRIGGER	
📅 ALLIE	
📅 V_ALLIE	
📅 V_PNJ	
⚡ NEW_ALLIE_TRIGGER	
📅 ALLIE	
📅 V_ALLIE	
📅 V_PNJ	
⚡ UPDATE_ALLIE_TRIGGER	
📅 V_ALLIE	
📅 V_PNJ	

## Used By

	Name
📅 V_ALLIE	
⚡ DELETE_ALLIE_TRIGGER	
⚡ NEW_ALLIE_TRIGGER	
⚡ UPDATE_ALLIE_TRIGGER	

## Script

```

SELECT P.Id, P.Nom, P.Taille, P.Alignement, P.Description, P.DateCreation, P.Nom_Classe, P.Nom_Race, P.Id_Lieux
FROM V_PNJ P
JOIN ALLIE A ON P.Id = A.Id

```

## 1.2.2. View: V\_MONSTRE

### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	TAILLE	NUMBER(5, 0)	<b>Nullable</b>
█	ALIGNEMENT	VARCHAR2(255 BYTE)	<b>Nullable</b>
█	DESCRIPTION	CLOB	<b>Nullable</b>
█	DATECREATION	DATE	<b>Nullable</b>
█	NOM_CLASSE	VARCHAR2(255 BYTE)	
█	NOM_RACE	VARCHAR2(255 BYTE)	
█	ID_LIEUX	NUMBER(5, 0)	
█	NIVEAU	NUMBER(5, 0)	<b>Nullable</b>
█	POINTDEVIE	NUMBER(5, 0)	<b>Nullable</b>
█	POINTDEMANA	NUMBER(5, 0)	<b>Nullable</b>
█	EXPERIENCE	NUMBER(5, 0)	<b>Nullable</b>
█	GOLD	NUMBER(5, 0)	<b>Nullable</b>
█	ESTBOSS	NUMBER(1, 0)	<b>Nullable</b>

### Triggers

	Name	When	Description
⚡	DELETE_MONSTRE_TRIGGER	Instead Of Delete	
	TRIGGER Delete_Monstre_trigger INSTEAD OF DELETE ON V_MONSTRE FOR EACH ROW BEGIN -- Suppression du Monstre DELETE FROM MONSTRE WHERE Id = :OLD.Id;  -- Suppression du PNJ DELETE FROM V_PNJ WHERE Id = :OLD.Id; END;		
⚡	NEW_MONSTRE_TRIGGER	Instead Of Insert	
	TRIGGER New_Monstre_trigger INSTEAD OF INSERT ON V_MONSTRE FOR EACH ROW DECLARE New_Id NUMBER; BEGIN -- Nouveau PNJ INSERT INTO V_PNJ(Nom, Taille, Alignement, Description, DateCreation, Nom_Classe, Nom_Race, Id_Lieux) VALUES (:NEW.Nom, :NEW.Taille, :NEW.Alignement, :NEW.Description, :NEW.DateCreation, :NEW.Nom_Classe, :NEW.Nom_Race, :NEW.Id_Lieux);  -- Récupérer l'ID généré SELECT MAX(Id) INTO New_Id FROM V_PNJ;  -- Nouveau MONSTRE INSERT INTO MONSTRE(Id, Niveau, PointDeVie, PointDeMane, Experience, Gold, EstBoss) VALUES (New_Id, :NEW.Niveau, :NEW.PointDeVie, :NEW.PointDeMane, :NEW.Experience, :NEW.Gold, :NEW.EstBoss); END;		
⚡	UPDATE_MONSTRE_TRIGGER	Instead Of Update	

```

TRIGGER Update_Monstre_trigger
INSTEAD OF UPDATE ON V_MONSTRE
FOR EACH ROW
BEGIN
    -- Mise à jour du pnj
    UPDATE V_PNJ
    SET Nom = :NEW.Nom,
        Taille = :NEW.Taille,
        Alignement = :NEW.Alignement,
        Description = :NEW.Description,
        DateCreation = :NEW.DateCreation,
        Nom_Classe = :NEW.Nom_Classe,
        Nom_Race = :NEW.Nom_Race,
        Id_Lieux = :NEW.Id_Lieux
    WHERE Id = :NEW.Id;

    -- Mise à jour du monstre
    UPDATE MONSTRE
    SET Niveau = :NEW.Niveau,
        PointDeVie = :NEW.PointDeVie,
        PointDeMana = :NEW.PointDeMana,
        Experience = :NEW.Experience,
        Gold = :NEW.Gold,
        EstBoss = :NEW.EstBoss
    WHERE Id = :NEW.Id;
END;

```

## Uses

	Name
📁 V_MONSTRE	
📅 MONSTRE	
📁 V_PNJ	
⚡ DELETE_MONSTRE_TRIGGER	
📅 MONSTRE	
📁 V_MONSTRE	
📁 V_PNJ	
⚡ NEW_MONSTRE_TRIGGER	
📅 MONSTRE	
📁 V_MONSTRE	
📁 V_PNJ	
⚡ UPDATE_MONSTRE_TRIGGER	
📅 MONSTRE	
📁 V_MONSTRE	
📁 V_PNJ	

## Used By

	Name
📁 V_MONSTRE	
⚡ DELETE_MONSTRE_TRIGGER	
⚡ NEW_MONSTRE_TRIGGER	
⚡ UPDATE_MONSTRE_TRIGGER	

## Script

```

SELECT P.Id, P.Nom, P.Taille, P.Alignement, P.Description, P.DateCreation, P.Nom_Classe, P.Nom_Race, P.Id_Lieux, M.Niveau,
M.PointDeVie, M.PointDeMana, M.Experience, M.Gold, M.EstBoss
FROM V_PNJ P
JOIN MONSTRE M ON P.Id = M.Id

```

### 1.2.3. View: V\_PJ

#### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	TAILLE	NUMBER(5, 0)	<b>Nullable</b>
█	ALIGNEMENT	VARCHAR2(255 BYTE)	<b>Nullable</b>
█	DESCRIPTION	CLOB	<b>Nullable</b>
█	DATECREATION	DATE	<b>Nullable</b>
█	NOM_CLASSE	VARCHAR2(255 BYTE)	
█	NOM_RACE	VARCHAR2(255 BYTE)	
█	ID_LIEUX	NUMBER(5, 0)	
█	NIVEAU	NUMBER(5, 0)	<b>Nullable</b>
█	POINTDEVIE	NUMBER(5, 0)	<b>Nullable</b>
█	POINTDEMANA	NUMBER(5, 0)	<b>Nullable</b>
█	EXPERIENCE	NUMBER(5, 0)	<b>Nullable</b>
█	GOLD	NUMBER(5, 0)	<b>Nullable</b>

#### Triggers

	Name	When	Description
⚡	DELETE_PJ_TRIGGER	Instead Of Delete	
	TRIGGER Delete_PJ_trigger INSTEAD OF DELETE ON V_PJ FOR EACH ROW BEGIN -- Suppression du PJ DELETE FROM PJ WHERE Id = :OLD.Id;  -- Suppression du personnage DELETE FROM PERSONNAGE WHERE Id = :OLD.Id; END;		
⚡	NEW_PJ_TRIGGER	Instead Of Insert	
	TRIGGER New_PJ_trigger INSTEAD OF INSERT ON V_PJ FOR EACH ROW DECLARE New_Id NUMBER; BEGIN -- Nouveau PERSONNAGE INSERT INTO PERSONNAGE(Nom, Taille, Alignement, Description, DateCreation, Nom_Classe, Nom_Race, Id_Lieux) VALUES (:NEW.Nom, :NEW.Taille, :NEW.Alignement, :NEW.Description, :NEW.DateCreation, :NEW.Nom_Classe, :NEW.Nom_Race, :NEW.Id_Lieux);  -- Récupérer l'ID généré SELECT MAX(Id) INTO New_Id FROM PERSONNAGE;  -- Nouveau PJ INSERT INTO PJ(Id, Niveau, PointDeVie, PointDeMana, Experience, Gold) VALUES (New_Id, :NEW.Niveau, :NEW.PointDeVie, :NEW.PointDeMana, :NEW.Experience, :NEW.Gold); END;		
⚡	UPDATE_PJ_TRIGGER	Instead Of Update	

```

TRIGGER Update_PJ_trigger
INSTEAD OF UPDATE ON V_PJ
FOR EACH ROW
BEGIN
    -- Mise à jour du personnage
    UPDATE PERSONNAGE
    SET Nom = :NEW.Nom,
        Taille = :NEW.Taille,
        Alignement = :NEW.Alignement,
        Description = :NEW.Description,
        DateCreation = :NEW.DateCreation,
        Nom_Classe = :NEW.Nom_Classe,
        Nom_Race = :NEW.Nom_Race,
        Id_Lieux = :NEW.Id_Lieux
    WHERE Id = :NEW.Id;

    -- Mise à jour du PJ
    UPDATE PJ
    SET Niveau = :NEW.Niveau,
        PointDeVie = :NEW.PointDeVie,
        PointDeMana = :NEW.PointDeMana,
        Experience = :NEW.Experience,
        Gold = :NEW.Gold
    WHERE Id = :NEW.Id;
END;

```

## Uses

Name
V_PJ
PERSONNAGE
PJ
DELETE_PJ_TRIGGER
PERSONNAGE
PJ
V_PJ
NEW_PJ_TRIGGER
PERSONNAGE
PJ
V_PJ
UPDATE_PJ_TRIGGER
PERSONNAGE
PJ
V_PJ

## Used By

Name
V_PJ
DELETE_PJ_TRIGGER
NEW_PJ_TRIGGER
UPDATE_PJ_TRIGGER

## Script

```

SELECT P.Id, P.Nom, P.Taille, P.Alignement, P.Description, P.DateCreation, P.Nom_Classe, P.Nom_Race, P.Id_Lieux, PJ.Niveau,
PJ.PointDeVie, PJ.PointDeMana, PJ.Experience, PJ.Gold
FROM PERSONNAGE P
JOIN PJ ON P.Id = PJ.Id

```

## 1.2.4. View: V\_PNJ

### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	TAILLE	NUMBER(5, 0)	<b>Nullable</b>
█	ALIGNEMENT	VARCHAR2(255 BYTE)	<b>Nullable</b>
█	DESCRIPTION	CLOB	<b>Nullable</b>
█	DATECREATION	DATE	<b>Nullable</b>
█	NOM_CLASSE	VARCHAR2(255 BYTE)	
█	NOM_RACE	VARCHAR2(255 BYTE)	
█	ID_LIEUX	NUMBER(5, 0)	

### Triggers

	Name	When	Description
⚡	DELETE_PNJ_TRIGGER	Instead Of Delete	
	TRIGGER Delete_PNJ_trigger INSTEAD OF DELETE ON V_PNJ FOR EACH ROW BEGIN -- Suppression du PNJ DELETE FROM PNJ WHERE Id = :OLD.Id;  -- Suppression du personnage DELETE FROM PERSONNAGE WHERE Id = :OLD.Id; END;		
⚡	NEW_PNJ_TRIGGER	Instead Of Insert	
	TRIGGER New_PNJ_trigger INSTEAD OF INSERT ON V_PNJ FOR EACH ROW DECLARE New_Id NUMBER; BEGIN -- Nouveau PERSONNAGE INSERT INTO PERSONNAGE (Nom, Taille, Alignement, Description, DateCreation, Nom_Classe, Nom_Race, Id_Lieux) VALUES (:NEW.Nom, :NEW.Taille, :NEW.Alignement, :NEW.Description, :NEW.DateCreation, :NEW.Nom_Classe, :NEW.Nom_Race, :NEW.Id_Lieux);  -- Récupérer l'ID généré SELECT MAX(Id) INTO New_Id FROM PERSONNAGE;  -- Nouveau PNJ INSERT INTO PNJ(Id) VALUES (New_Id); END;		
⚡	UPDATE_PNJ_TRIGGER	Instead Of Update	

```

TRIGGER Update_PNJ_trigger
  INSTEAD OF UPDATE ON V_PNJ
  FOR EACH ROW
  BEGIN
    -- Mise à jour du personnage
    UPDATE PERSONNAGE
    SET Nom = :NEW.Nom,
        Taille = :NEW.Taille,
        Alignement = :NEW.Alignement,
        Description = :NEW.Description,
        DateCreation = :NEW.DateCreation,
        Nom_Classe = :NEW.Nom_Classe,
        Nom_Race = :NEW.Nom_Race,
        Id_Lieux = :NEW.Id_Lieux
    WHERE Id = :NEW.Id;
  END;

```

## Uses

	Name
📅 V_PNJ	
👤 PERSONNAGE	
👤 PNJ	
⚡ DELETE_PNJ_TRIGGER	
👤 PERSONNAGE	
👤 PNJ	
📅 V_PNJ	
⚡ NEW_PNJ_TRIGGER	
👤 PERSONNAGE	
👤 PNJ	
📅 V_PNJ	
⚡ UPDATE_PNJ_TRIGGER	
👤 PERSONNAGE	
👤 V_PNJ	

## Used By

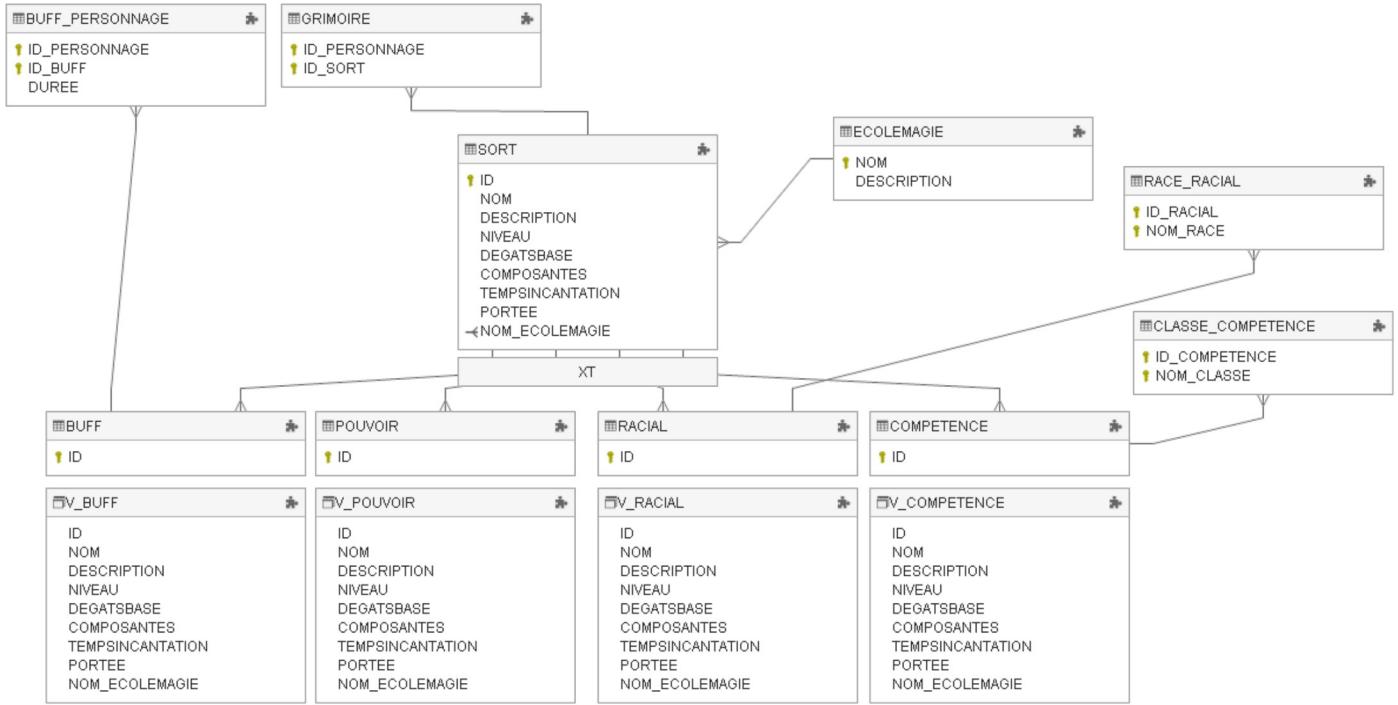
	Name
📅 V_PNJ	
📅 V_ALLIE	
📅 V_MONSTRE	
⚡ DELETE_ALLIE_TRIGGER	
⚡ DELETE_MONSTRE_TRIGGER	
⚡ DELETE_PNJ_TRIGGER	
⚡ NEW_ALLIE_TRIGGER	
⚡ NEW_MONSTRE_TRIGGER	
⚡ NEW_PNJ_TRIGGER	
⚡ UPDATE_ALLIE_TRIGGER	
⚡ UPDATE_MONSTRE_TRIGGER	
⚡ UPDATE_PNJ_TRIGGER	

## Script

```
SELECT P.Id, P.Nom, P.Taille, P.Alignement, P.Description, P.DateCreation, P.Nom_Classe, P.Nom_Race, P.Id_Lieux  
FROM PERSONNAGE P  
JOIN PNJ ON P.Id = PNJ.Id
```



## 2. HERITAGE SORT



La table **SORT** est la mère.

Elle possède quatre filles :

- **BUFF**
- **POUVOIR**
- **RACIAL**
- **COMPETENCE**

La gestion de l'héritage **SORT** est la même que pour la famille **PERSONNAGE**.

## 2.1. Tables

### 2.1.1. Table: BUFF

#### Columns

	Name	Data type	Description / Attributes
ID		NUMBER	<b>References:</b> SORT

#### Links to

	Table	Join	Title / Name / Description
SORT		<b>BUFFID</b> = SORTID	FK_SORT_BUFF

#### Linked from

	Table	Join	Title / Name / Description
BUFF_PERSONNAGE		<b>BUFFID</b> = BUFF_PERSONNAGEID_BUFF	FK_BUFF_BUFF_PERSONNAGE

#### Unique keys

	Columns	Name / Description
ID		SYS_C0040581

#### Uses

	Name
BUFF	
SORT	

#### Used By

	Name
BUFF	
V_BUFF	
DELETE_BUFF_TRIGGER	
NEW_BUFF_TRIGGER	
BUFF_PERSONNAGE	

## 2.1.2. Table: BUFF\_PERSONNAGE

### Columns

		Name	Data type	Description / Attributes
■	🔑	ID_PERSONNAGE	NUMBER(5, 0)	<b>References:</b> PERSONNAGE
■	🔑	ID_BUFF	NUMBER(5, 0)	<b>References:</b> BUFF
■		DUREE	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 60

### Links to

Table		Join	Title / Name / Description
→	BUFF	<b>BUFF_PERSONNAGEID_BUFF</b> = BUFFID	FK_BUFF_BUFF_PERSONNAGE
→	PERSONNAGE	<b>BUFF_PERSONNAGEID_PERSONNAGE</b> = PERSONNAGEID	FK_PERSONNAGE_BUFF_PERSONNAGE

### Unique keys

Columns		Name / Description
🔑	ID_PERSONNAGE, ID_BUFF	SYS_C0040663

### Uses

Name	
█	BUFF_PERSONNAGE
→	BUFF
→	PERSONNAGE

## 2.1.3. Table: CLASSE\_COMPETENCE

### Columns

		Name	Data type	Description / Attributes
		ID_COMPETENCE	NUMBER(5, 0)	References: COMPETENCE
		NOM_CLASSE	VARCHAR2(255 BYTE)	References: CLASSE

### Links to

Table		Join	Title / Name / Description
	CLASSE	CLASSE_COMPETENCE NOM_CLASSE = CLASSENOM	FK_CLASSE_CC
	COMPETENCE	CLASSE_COMPETENCE ID_COMPETEN CE = COMPETENCEID	FK_COMPETENCE_CC

### Unique keys

Columns		Name / Description
	ID_COMPETENCE, NOM_CLASSE	SYS_C0040628

### Uses

		Name
		CLASSE_COMPETENCE
		CLASSE
		COMPETENCE

## 2.1.4. Table: COMPETENCE

### Columns

	Name	Data type	Description / Attributes
	ID	NUMBER	References: SORT

### Links to

	Table	Join	Title / Name / Description
	SORT	COMPETENCEID = SORTID	FK_SORT_COMPETENCE

### Linked from

	Table	Join	Title / Name / Description
	CLASSE_COMPETENCE	COMPETENCEID = CLASSE_COMPETENCEID_COMPETENCE	FK_COMPETENCE_CC

### Unique keys

	Columns	Name / Description
	ID	SYS_C0040587

### Uses

	Name
	COMPETENCE
	SORT

### Used By

	Name
	COMPETENCE
	V_COMPETENCE
	DELETE_COMPETENCE_TRIGGER
	NEW_COMPETENCE_TRIGGER
	CLASSE_COMPETENCE

## 2.1.5. Table: ECOLEMAGIE

### Columns

Name		Data type	Description / Attributes
	NOM	VARCHAR2(255 BYTE)	
	DESCRIPTION	CLOB	<b>Nullable</b>

### Linked from

Table		Join	Title / Name / Description
	SORT	ECOLEMAGIENOM = SORTNOM_ECOLEMAGIE	FK_ECOLEMAGIE_SORT

### Unique keys

Columns		Name / Description
	NOM	SYS_C0040502

### Used By

Name	
	ECOLEMAGIE
	SORT

## 2.1.6. Table: GRIMOIRE

### Columns

		Name	Data type	Description / Attributes
目	🔑	ID_PERSONNAGE	NUMBER(5, 0)	References: PERSONNAGE
目	🔑	ID_SORT	NUMBER(5, 0)	References: SORT

### Links to

Table		Join	Title / Name / Description
→	PERSONNAGE	GRIMOIREID_PERSONNAGE = PERSONNAGEID	FK_PERSONNAGE_GRIMOIRE
→	SORT	GRIMOIREID_SORT = SORTID	FK_SORT_GRIMOIRE

### Unique keys

Columns		Name / Description
🔑	ID_PERSONNAGE, ID_SORT	SYS_C0040643

### Uses

		Name
grid	GRIMOIRE	
→	PERSONNAGE	
→	SORT	

## 2.1.7. Table: POUVOIR

### Columns

	Name	Data type	Description / Attributes
	ID	NUMBER	<b>References:</b> SORT

### Links to

	Table	Join	Title / Name / Description
	SORT	POUVOIRID = SORTID	FK_SORT_POUVOIR

### Unique keys

	Columns	Name / Description
	ID	SYS_C0040583

### Uses

	Name
	POUVOIR
	SORT

### Used By

	Name
	POUVOIR
	V_POUVOIR
	DELETE_POUVOIR_TRIGGER
	NEW_POUVOIR_TRIGGER

## 2.1.8. Table: RACE\_RACIAL

### Columns

		Name	Data type	Description / Attributes
目	🔑	ID_RACIAL	NUMBER(5, 0)	References: RACIAL
目	🔑	NOM_RACE	VARCHAR2(255 BYTE)	References: RACE

### Links to

Table		Join	Title / Name / Description
→	RACE	RACE_RACIAL NOM_RACE = RACENOM	FK_RACE_RACE_RACIAL
→	RACIAL	RACE_RACIAL ID_RACIAL = RACIALID	FK_RACIAL_RACE_RACIAL

### Unique keys

Columns		Name / Description
🔑	ID_RACIAL, NOM_RACE	SYS_C0040631

### Uses

		Name
grid	RACE_RACIAL	
→	RACE	
→	RACIAL	

## 2.1.9. Table: RACIAL

### Columns

	Name	Data type	Description / Attributes
	ID	NUMBER	References: SORT

### Links to

	Table	Join	Title / Name / Description
	SORT	RACIALID = SORTID	FK_SORT_RACIAL

### Linked from

	Table	Join	Title / Name / Description
	RACE_RACIAL	RACIALID = RACE_RACIALID_RACIAL	FK_RACIAL_RACE_RACIAL

### Unique keys

	Columns	Name / Description
	ID	SYS_C0040585

### Uses

	Name
	RACIAL
	SORT

### Used By

	Name
	RACIAL
	V_RACIAL
	DELETE_RACIAL_TRIGGER
	NEW_RACIAL_TRIGGER
	RACE_RACIAL

## 2.1.10. Table: SORT

### Columns

		Name	Data type	Description / Attributes
█	█	ID	NUMBER	
█		NOM	VARCHAR2(255 BYTE)	
█		DESCRIPTION	CLOB	<b>Nullable</b>
█		NIVEAU	NUMBER(5, 0)	<b>Nullable</b> Default: 1
█		DEGATSBASE	NUMBER(5, 0)	<b>Nullable</b> Default: 100
█		COMPOSANTES	VARCHAR2(255 BYTE)	<b>Nullable</b>
█		TEMPSINCANTATION	NUMBER(5, 0)	<b>Nullable</b> Default: 0
█		PORTEE	NUMBER(5, 0)	<b>Nullable</b> Default: 5
█		NOM_ECOLEMAGIE	VARCHAR2(255 BYTE)	<b>References:</b> ECOLEMAGIE

### Links to

	Table	Join	Title / Name / Description
→	ECOLEMAGIE	SORTNOM_ECOLEMAGIE = ECOLEMAGIENOM	FK_ECOLEMAGIE_SORT

### Linked from

	Table	Join	Title / Name / Description
←	APPARENCE_SORT	SORTID = APPARENCE_SORTID_SORT	FK_SORT_AS
←	APPARENCEPOSSIBLE_SORT	SORTID = APPARENCEPOSSIBLE_SORTID_SORT	FK_SORTAPS
←	BUFF	SORTID = BUFFID	FK_SORT_BUFF
←	COMPETENCE	SORTID = COMPETENCEID	FK_SORT_COMPETENCE
←	GRIMOIRE	SORTID = GRIMOIREID_SORT	FK_SORT_GRIMOIRE
←	POUVOIR	SORTID = POUVOIRID	FK_SORT_POUVOIR
←	RACIAL	SORTID = RACIALID	FK_SORT_RACIAL

### Unique keys

	Columns	Name / Description
█	ID	SYS_C0040579

## Triggers

	Name	When	Description
⚡	NEW_SORT_TRIGGER	Before Insert	
<pre>TRIGGER New_Sort_trigger BEFORE INSERT ON SORT FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := SORT_seq.NEXTVAL; END;</pre>			

## Uses

	Name
grid	SORT
→	ECOLEMAGIE
⚡	NEW_SORT_TRIGGER
grid	SORT

## Used By

	Name
grid	SORT
file	V_BUFF
file	V_COMPETENCE
file	V_POUVOIR
file	V_RACIAL
⚡	DELETE_BUFF_TRIGGER
⚡	DELETE_COMPETENCE_TRIGGER
⚡	DELETE_POUVOIR_TRIGGER
⚡	DELETE_RACIAL_TRIGGER
⚡	NEW_BUFF_TRIGGER
⚡	NEW_COMPETENCE_TRIGGER
⚡	NEW_POUVOIR_TRIGGER
⚡	NEW_RACIAL_TRIGGER
⚡	NEW_SORT_TRIGGER
⚡	UPDATE_BUFF_TRIGGER
⚡	UPDATE_COMPETENCE_TRIGGER
⚡	UPDATE_POUVOIR_TRIGGER
⚡	UPDATE_RACIAL_TRIGGER
←	APPARENCE_SORT
←	APPARENCEPOSSIBLE_SORT
←	BUFF
←	COMPETENCE
←	GRIMOIRE

Name

→ POUVOIR

→ RACIAL

## 2.2. Views

### 2.2.1. View: V\_BUFF

#### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	DESCRIPTION	CLOB	<b>Nullable</b>
█	NIVEAU	NUMBER(5, 0)	<b>Nullable</b>
█	DEGATSBASE	NUMBER(5, 0)	<b>Nullable</b>
█	COMPOSANTES	VARCHAR2(255 BYTE)	<b>Nullable</b>
█	TEMPSINCANTATION	NUMBER(5, 0)	<b>Nullable</b>
█	PORTEE	NUMBER(5, 0)	<b>Nullable</b>
█	NOM_ECOLEMAGIE	VARCHAR2(255 BYTE)	

#### Triggers

	Name	When	Description
⚡	DELETE_BUFF_TRIGGER	Instead Of Delete	
	TRIGGER Delete_Buff_trigger INSTEAD OF DELETE ON V_BUFF FOR EACH ROW BEGIN -- Suppression du Buff DELETE FROM BUFF WHERE Id = :OLD.Id;  -- Suppression du Sort DELETE FROM SORT WHERE Id = :OLD.Id; END;		
⚡	NEW_BUFF_TRIGGER	Instead Of Insert	
	TRIGGER New_Buff_trigger INSTEAD OF INSERT ON V_BUFF FOR EACH ROW DECLARE New_Id NUMBER; BEGIN -- Nouveau SORT INSERT INTO SORT(Id, Nom, Description, Niveau, DegatsBase, Composantes, TempsIncantation, Portee, Nom_EcoleMagie) VALUES (:NEW.Id, :NEW.Nom, :NEW.Description, :NEW.Niveau, :NEW.DegatsBase, :NEW.Composantes, :NEW.TempsIncantation, :NEW.Portee, :NEW.Nom_EcoleMagie);  -- Récupérer l'ID généré SELECT MAX(Id) INTO New_Id FROM SORT;  -- Nouveau BUFF INSERT INTO BUFF(Id) VALUES (New_Id); END;		
⚡	UPDATE_BUFF_TRIGGER	Instead Of Update	

```

TRIGGER Update_Buff_trigger
INSTEAD OF UPDATE ON V_BUFF
FOR EACH ROW
BEGIN
    -- Mise à jour du SORT
    UPDATE SORT
    SET Nom = :NEW.Nom,
        Description = :NEW.Description,
        Niveau = :NEW.Niveau,
        DegatsBase = :NEW.DegatsBase,
        Composantes = :NEW.Composantes,
        TempsIncantation = :NEW.TempsIncantation,
        Portee = :NEW.Portee,
        Nom_EcoleMagie = :NEW.Nom_EcoleMagie
    WHERE Id = :NEW.Id;
END;

```

## Uses

Name
V_BUFF
BUFF
SORT
DELETE_BUFF_TRIGGER
BUFF
SORT
V_BUFF
NEW_BUFF_TRIGGER
BUFF
SORT
V_BUFF
UPDATE_BUFF_TRIGGER
SORT
V_BUFF

## Used By

Name
V_BUFF
DELETE_BUFF_TRIGGER
NEW_BUFF_TRIGGER
UPDATE_BUFF_TRIGGER

## Script

```

SELECT S.Id, S.Nom, S.Description, S.Niveau, S.DegatsBase, S.Composantes, S.TempsIncantation, S.Portee, S.Nom_EcoleMagie
FROM SORT S
JOIN BUFF B ON S.Id = B.Id

```

## 2.2.2. View: V\_COMPETENCE

### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	DESCRIPTION	CLOB	<b>Nullable</b>
█	NIVEAU	NUMBER(5, 0)	<b>Nullable</b>
█	DEGATSBASE	NUMBER(5, 0)	<b>Nullable</b>
█	COMPOSANTES	VARCHAR2(255 BYTE)	<b>Nullable</b>
█	TEMPSINCANTATION	NUMBER(5, 0)	<b>Nullable</b>
█	PORTEE	NUMBER(5, 0)	<b>Nullable</b>
█	NOM_ECOLEMAGIE	VARCHAR2(255 BYTE)	

### Triggers

	Name	When	Description
⚡	DELETE_COMPETENCE_TRIGGER	Instead Of Delete	
	TRIGGER Delete_Compotence_trigger INSTEAD OF DELETE ON V_COMPETENCE FOR EACH ROW BEGIN -- Suppression du COMPETENCE DELETE FROM COMPETENCE WHERE Id = :OLD.Id;  -- Suppression du Sort DELETE FROM SORT WHERE Id = :OLD.Id; END;		
⚡	NEW_COMPETENCE_TRIGGER	Instead Of Insert	
	TRIGGER New_Compotence_trigger INSTEAD OF INSERT ON V_COMPETENCE FOR EACH ROW DECLARE New_Id NUMBER; BEGIN -- Nouveau SORT INSERT INTO SORT(Id, Nom, Description, Niveau, DegatsBase, Composantes, TempsIncantation, Portee, Nom_EcoleMagie) VALUES (:NEW.Id, :NEW.Nom, :NEW.Description, :NEW.Niveau, :NEW.DegatsBase, :NEW.Composantes, :NEW.TempsIncantation, :NEW.Portee, :NEW.Nom_EcoleMagie);  -- Récupérer l'ID généré SELECT MAX(Id) INTO New_Id FROM SORT;  -- Nouveau COMPETENCE INSERT INTO COMPETENCE (Id) VALUES (New_Id); END;		
⚡	UPDATE_COMPETENCE_TRIGGER	Instead Of Update	

```

TRIGGER Update_Compentence_trigger
INSTEAD OF UPDATE ON V_COMPENCE
FOR EACH ROW
BEGIN
    -- Mise à jour du SORT
    UPDATE SORT
    SET Nom = :NEW.Nom,
        Description = :NEW.Description,
        Niveau = :NEW.Niveau,
        DegatsBase = :NEW.DegatsBase,
        Composantes = :NEW.Composantes,
        TempsIncantation = :NEW.TempsIncantation,
        Portee = :NEW.Portee,
        Nom_EcoleMagie = :NEW.Nom_EcoleMagie
    WHERE Id = :NEW.Id;
END;

```

## Uses

Name
V_COMPENCE
COMPENCE
SORT
DELETE_COMPENCE_TRIGGER
COMPENCE
SORT
V_COMPENCE
NEW_COMPENCE_TRIGGER
COMPENCE
SORT
V_COMPENCE
UPDATE_COMPENCE_TRIGGER
SORT
V_COMPENCE

## Used By

Name
V_COMPENCE
DELETE_COMPENCE_TRIGGER
NEW_COMPENCE_TRIGGER
UPDATE_COMPENCE_TRIGGER

## Script

```

SELECT S.Id, S.Nom, S.Description, S.Niveau, S.DegatsBase, S.Composantes, S.TempsIncantation, S.Portee, S.Nom_EcoleMagie
FROM SORT S
JOIN COMPENCE C ON S.Id = C.Id

```

## 2.2.3. View: V\_CONSUMMABLE

### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	DESCRIPTION	CLOB	<b>Nullable</b>
█	EFFET	VARCHAR2(255 BYTE)	<b>Nullable</b>
█	ID_TYPECONSOMMABLE	NUMBER(5, 0)	<b>Nullable</b>

### Triggers

	Name	When	Description
⚡	DELETE_CONSUMMABLE_TRIGGER	Instead Of Delete	<pre>TRIGGER Delete_Consummable_trigger   INSTEAD OF DELETE ON V_CONSUMMABLE   FOR EACH ROW   BEGIN     -- Suppression du consommable     DELETE FROM CONSOMMABLE     WHERE Id = :OLD.Id;      -- Suppression ddu divers     DELETE FROM V_DIVERS     WHERE Id = :OLD.Id;   END;</pre>
⚡	NEW_CONSUMMABLE_TRIGGER	Instead Of Insert	<pre>TRIGGER New_Consummable_trigger   INSTEAD OF INSERT ON V_CONSUMMABLE   FOR EACH ROW   DECLARE     New_Id NUMBER;   BEGIN     -- Nouvel Divers     INSERT INTO V_DIVERS(Nom, Description)     VALUES (:NEW.Nom, :NEW.Description);      -- Récupérer l'ID généré     SELECT MAX(Id) INTO New_Id FROM V_DIVERS;      -- Nouvelle Consommable     INSERT INTO CONSOMMABLE(Id, Effet, Id_Typeconsommable)     VALUES (New_Id, :NEW.Effet, :NEW.Id_Typeconsommable);   END;</pre>
⚡	UPDATE_CONSUMMABLE_TRIGGER	Instead Of Update	<pre>TRIGGER Update_Consummable_trigger   INSTEAD OF UPDATE ON V_CONSUMMABLE   FOR EACH ROW   BEGIN     -- Mise à jour du divers     UPDATE V_DIVERS     SET Nom = :NEW.Nom,         Description = :NEW.Description     WHERE Id = :NEW.Id;      -- Mise à jour du consommable     UPDATE CONSOMMABLE     SET Effet = :NEW.Effet,         Id_Typeconsommable = :NEW.Id_Typeconsommable     WHERE Id = :NEW.Id;   END;</pre>

## Uses

Name
CONSOMMABLE
V_DIVERS
DELETE_CONSOMMABLE_TRIGGER
CONSOMMABLE
V_CONSOMMABLE
V_DIVERS
NEW_CONSOMMABLE_TRIGGER
CONSOMMABLE
V_CONSOMMABLE
V_DIVERS
UPDATE_CONSOMMABLE_TRIGGER
CONSOMMABLE
V_CONSOMMABLE
V_DIVERS

## Used By

Name
V_CONSOMMABLE
DELETE_CONSOMMABLE_TRIGGER
NEW_CONSOMMABLE_TRIGGER
UPDATE_CONSOMMABLE_TRIGGER

## Script

```
SELECT I.Id, I.Nom, I.Description, C.Effet, C.Id_Typeconsommable
  FROM V_DIVERS I
  JOIN CONSOMMABLE C ON I.Id = C.Id
```

## 2.2.4. View: V\_POUVOIR

### Columns

	Name	Data type	Description / Attributes
■	ID	NUMBER	
■	NOM	VARCHAR2(255 BYTE)	
■	DESCRIPTION	CLOB	<b>Nullable</b>
■	NIVEAU	NUMBER(5, 0)	<b>Nullable</b>
■	DEGATSBASE	NUMBER(5, 0)	<b>Nullable</b>
■	COMPOSANTES	VARCHAR2(255 BYTE)	<b>Nullable</b>
■	TEMPSINCANTATION	NUMBER(5, 0)	<b>Nullable</b>
■	PORTEE	NUMBER(5, 0)	<b>Nullable</b>
■	NOM_ECOLEMAGIE	VARCHAR2(255 BYTE)	

### Triggers

	Name	When	Description
⚡	DELETE_POUVOIR_TRIGGER	Instead Of Delete	
	TRIGGER Delete_Pouvoir_trigger INSTEAD OF DELETE ON V_POUVOIR FOR EACH ROW BEGIN -- Suppression du POUVOIR DELETE FROM POUVOIR WHERE Id = :OLD.Id;  -- Suppression du Sort DELETE FROM SORT WHERE Id = :OLD.Id; END;		
⚡	NEW_POUVOIR_TRIGGER	Instead Of Insert	
	TRIGGER New_Pouvoir_trigger INSTEAD OF INSERT ON V_POUVOIR FOR EACH ROW DECLARE New_Id NUMBER; BEGIN -- Nouveau SORT INSERT INTO SORT(Id, Nom, Description, Niveau, DegatsBase, Composantes, TempsIncantation, Portee, Nom_EcoleMagie) VALUES (:NEW.Id, :NEW.Nom, :NEW.Description, :NEW.Niveau, :NEW.DegatsBase, :NEW.Composantes, :NEW.TempsIncantation, :NEW.Portee, :NEW.Nom_EcoleMagie);  -- Récupérer l'ID généré SELECT MAX(Id) INTO New_Id FROM SORT;  -- Nouveau POUVOIR INSERT INTO POUVOIR(Id) VALUES (New_Id); END;		
⚡	UPDATE_POUVOIR_TRIGGER	Instead Of Update	

```

TRIGGER Update_Pouvoir_trigger
INSTEAD OF UPDATE ON V_POUVOIR
FOR EACH ROW
BEGIN
    -- Mise à jour du SORT
    UPDATE SORT
    SET Nom = :NEW.Nom,
        Description = :NEW.Description,
        Niveau = :NEW.Niveau,
        DegatsBase = :NEW.DegatsBase,
        Composantes = :NEW.Composantes,
        TempsIncantation = :NEW.TempsIncantation,
        Portee = :NEW.Portee,
        Nom_EcoleMagie = :NEW.Nom_EcoleMagie
    WHERE Id = :NEW.Id;
END;

```

## Uses

Name
V_POUVOIR
POUVOIR
SORT
DELETE_POUVOIR_TRIGGER
POUVOIR
SORT
V_POUVOIR
NEW_POUVOIR_TRIGGER
POUVOIR
SORT
V_POUVOIR
UPDATE_POUVOIR_TRIGGER
SORT
V_POUVOIR

## Used By

Name
V_POUVOIR
DELETE_POUVOIR_TRIGGER
NEW_POUVOIR_TRIGGER
UPDATE_POUVOIR_TRIGGER

## Script

```

SELECT S.Id, S.Nom, S.Description, S.Niveau, S.DegatsBase, S.Composantes, S.TempsIncantation, S.Portee, S.Nom_EcoleMagie
FROM SORT S
JOIN POUVOIR P ON S.Id = P.Id

```

## 2.2.5. View: V\_RACIAL

### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	DESCRIPTION	CLOB	<b>Nullable</b>
█	NIVEAU	NUMBER(5, 0)	<b>Nullable</b>
█	DEGATSBASE	NUMBER(5, 0)	<b>Nullable</b>
█	COMPOSANTES	VARCHAR2(255 BYTE)	<b>Nullable</b>
█	TEMPSINCANTATION	NUMBER(5, 0)	<b>Nullable</b>
█	PORTEE	NUMBER(5, 0)	<b>Nullable</b>
█	NOM_ECOLEMAGIE	VARCHAR2(255 BYTE)	

### Triggers

	Name	When	Description
⚡	DELETE_RACIAL_TRIGGER	Instead Of Delete	
	TRIGGER Delete_Racial_trigger INSTEAD OF DELETE ON V_RACIAL FOR EACH ROW BEGIN -- Suppression du RACIAL DELETE FROM RACIAL WHERE Id = :OLD.Id;  -- Suppression du Sort DELETE FROM SORT WHERE Id = :OLD.Id; END;		
⚡	NEW_RACIAL_TRIGGER	Instead Of Insert	
	TRIGGER New_Racial_trigger INSTEAD OF INSERT ON V_RACIAL FOR EACH ROW DECLARE New_Id NUMBER; BEGIN -- Nouveau SORT INSERT INTO SORT(Id, Nom, Description, Niveau, DegatsBase, Composantes, TempsIncantation, Portee, Nom_EcoleMagie) VALUES (:NEW.Id, :NEW.Nom, :NEW.Description, :NEW.Niveau, :NEW.DegatsBase, :NEW.Composantes, :NEW.TempsIncantation, :NEW.Portee, :NEW.Nom_EcoleMagie);  -- Récupérer l'ID généré SELECT MAX(Id) INTO New_Id FROM SORT;  -- Nouveau RACIAL INSERT INTO RACIAL(Id) VALUES (New_Id); END;		
⚡	UPDATE_RACIAL_TRIGGER	Instead Of Update	

```

TRIGGER Update_Racial_trigger
INSTEAD OF UPDATE ON V_RACIAL
FOR EACH ROW
BEGIN
    -- Mise à jour du SORT
    UPDATE SORT
    SET Nom = :NEW.Nom,
        Description = :NEW.Description,
        Niveau = :NEW.Niveau,
        DegatsBase = :NEW.DegatsBase,
        Composantes = :NEW.Composantes,
        TempsIncantation = :NEW.TempsIncantation,
        Portee = :NEW.Portee,
        Nom_EcoleMagie = :NEW.Nom_EcoleMagie
    WHERE Id = :NEW.Id;
END;

```

## Uses

Name
V_RACIAL
RACIAL
SORT
DELETE_RACIAL_TRIGGER
RACIAL
SORT
V_RACIAL
NEW_RACIAL_TRIGGER
RACIAL
SORT
V_RACIAL
UPDATE_RACIAL_TRIGGER
SORT
V_RACIAL

## Used By

Name
V_RACIAL
DELETE_RACIAL_TRIGGER
NEW_RACIAL_TRIGGER
UPDATE_RACIAL_TRIGGER

## Script

```

SELECT S.Id, S.Nom, S.Description, S.Niveau, S.DegatsBase, S.Composantes, S.TempsIncantation, S.Portee, S.Nom_EcoleMagie
FROM SORT S
JOIN RACIAL R ON S.Id = R.Id

```

## 2.2.6. View: V\_RESSOURCE

### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	DESCRIPTION	CLOB	<b>Nullable</b>
█	ID_TYPERESSOURCE	NUMBER(5, 0)	<b>Nullable</b>

### Triggers

	Name	When	Description
⚡	DELETE_RESSOURCE_TRIGGER	Instead Of Delete	<pre>TRIGGER Delete_Ressource_trigger   INSTEAD OF DELETE ON V_RESSOURCE   FOR EACH ROW   BEGIN     -- Suppression du consommable     DELETE FROM RESSOURCE     WHERE Id = :OLD.Id;      -- Suppression ddu divers     DELETE FROM V_DIVERS     WHERE Id = :OLD.Id;   END;</pre>
⚡	NEW_RESSOURCE_TRIGGER	Instead Of Insert	<pre>TRIGGER New_Ressource_trigger   INSTEAD OF INSERT ON V_RESSOURCE   FOR EACH ROW   DECLARE     New_Id NUMBER;   BEGIN     -- Nouvel Divers     INSERT INTO V_DIVERS(Nom, Description)     VALUES (:NEW.Nom, :NEW.Description);      -- Récupérer l'ID généré     SELECT MAX(Id) INTO New_Id FROM V_DIVERS;      -- Nouvelle Ressource     INSERT INTO RESSOURCE(Id, Id_Typeressource)     VALUES (New_Id, :NEW.Id_Typeressource);   END;</pre>
⚡	UPDATE_RESSOURCE_TRIGGER	Instead Of Update	<pre>TRIGGER Update_Ressource_trigger   INSTEAD OF UPDATE ON V_RESSOURCE   FOR EACH ROW   BEGIN     -- Mise à jour du divers     UPDATE V_DIVERS     SET Nom = :NEW.Nom,         Description = :NEW.Description     WHERE Id = :NEW.Id;      -- Mise à jour de la ressource     UPDATE RESSOURCE     SET Id_Typeressource = :NEW.Id_Typeressource     WHERE Id = :NEW.Id;   END;</pre>

### Uses

	Name
█	V_RESSOURCE
█	RESSOURCE

## Name

V\_DIVERS

DELETE\_RESSOURCE\_TRIGGER

RESSOURCE

V\_DIVERS

V\_RESOURCE

NEW\_RESSOURCE\_TRIGGER

RESSOURCE

V\_DIVERS

V\_RESOURCE

UPDATE\_RESSOURCE\_TRIGGER

RESSOURCE

V\_DIVERS

V\_RESOURCE

## Used By

## Name

V\_RESOURCE

DELETE\_RESSOURCE\_TRIGGER

NEW\_RESSOURCE\_TRIGGER

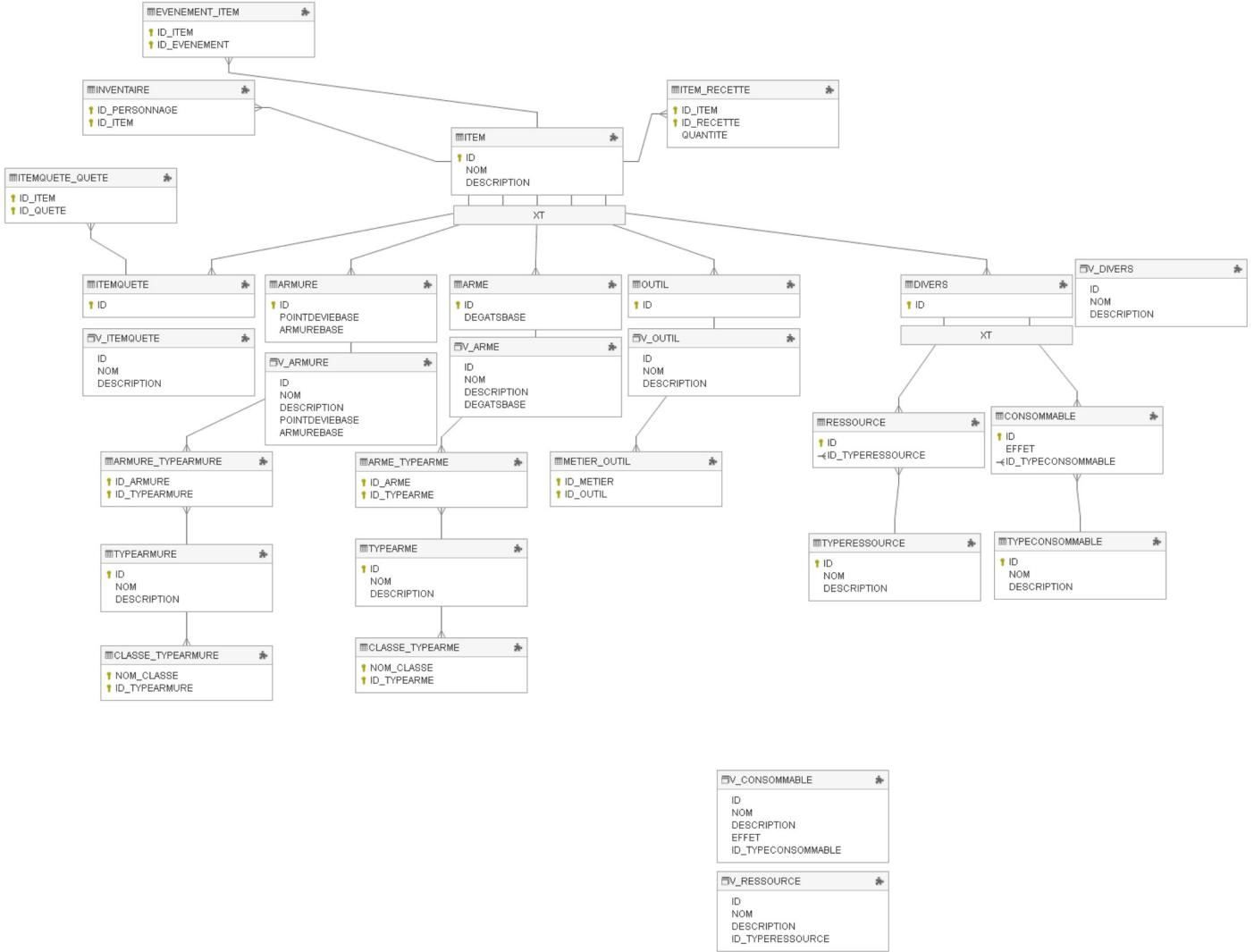
UPDATE\_RESSOURCE\_TRIGGER

## Script

```
SELECT I.Id, I.Nom, I.Description, R.Id_Typeressource
  FROM V_DIVERS I
  JOIN RESSOURCE R ON I.Id = R.Id
```



### 3. HERITAGE ITEM



La table ITEM est la grand-mère.

Elle possède cinq filles :

- ITEMQUETE
- ARMURE
- ARME
- OUTIL
- DIVERS

DIVERS est elle-même la table mère de deux filles :

- CONSUMMABLE
- RESSOURCE

La gestion de l'héritage ITEM est la même que pour la famille PERSONNAGE.

## 3.1. Tables

### 3.1.1. Table: ARME

#### Columns

		Name	Data type	Description / Attributes
目	🔑 ID	NUMBER		References: ITEM
目	DEGATSBASE	NUMBER(5, 0)		Nullable Default: 100

#### Links to

		Table	Join	Title / Name / Description
→	ITEM	ARMEID = ITEMID		FK_ARME_ITEM

#### Linked from

		Table	Join	Title / Name / Description
←	ARME_TYPEARME	ARMEID = ARME_TYPEARMEID_ARME		FK_ARME_AT

#### Unique keys

		Columns	Name / Description
🔑	ID	SYS_C0040521	

#### Uses

		Name
grid	ARME	
→	ITEM	

#### Used By

		Name
grid	ARME	
file	V_ARME	
flash	DELETE_ARME_TRIGGER	
flash	NEW_ARME_TRIGGER	
flash	UPDATE_ARME_TRIGGER	
←	ARME_TYPEARME	

### 3.1.2. Table: ARME\_TYPEARME

#### Columns

		Name	Data type	Description / Attributes
目	🔑	ID_ARME	NUMBER(5, 0)	References: ARME
目	🔑	ID_TYPEARME	NUMBER(5, 0)	References: TYPEARME

#### Links to

Table		Join	Title / Name / Description
→	ARME	ARME_TYPEARMEID_ARME = ARMEID	FK_ARME_AT
→	TYPEARME	ARME_TYPEARMEID_TYPEARME = TYPEARMEID	FK_TYPEARME_AT

#### Unique keys

Columns		Name / Description
🔑	ID_ARME, ID_TYPEARME	SYS_C0040607

#### Uses

		Name
grid	ARME_TYPEARME	
→	ARME	
→	TYPEARME	

### 3.1.3. Table: ARMURE

#### Columns

		Name	Data type	Description / Attributes
		ID	NUMBER	<b>References:</b> ITEM
		POINTDEVIEBASE	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 100
		ARMUREBASE	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 100

#### Links to

Table		Join	Title / Name / Description
	ITEM	<b>ARMUREID = ITEMID</b>	FK_ARMURE_ITEM

#### Linked from

Table		Join	Title / Name / Description
	ARMURE_TYPEARMURE	<b>ARMUREID =</b> ARMURE_TYPEARMUREID_ARMURE	FK_ARMURE_AT

#### Unique keys

Columns		Name / Description
	ID	SYS_C0040519

#### Uses

		Name
	ARMURE	
	ITEM	

#### Used By

		Name
	ARMURE	
	V_ARMURE	
	DELETE_ARMURE_TRIGGER	
	NEW_ARMURE_TRIGGER	
	UPDATE_ARMURE_TRIGGER	
	ARMURE_TYPEARMURE	

### 3.1.4. Table: ARMURE\_TYPEARMURE

#### Columns

		Name	Data type	Description / Attributes
目	🔑	ID_ARMURE	NUMBER(5, 0)	References: ARMURE
目	🔑	ID_TYPEARMURE	NUMBER(5, 0)	References: TYPEARMURE

#### Links to

Table		Join	Title / Name / Description
→	ARMURE	ARMURE_TYPEARMUREID_ARMURE = ARMUREID	FK_ARMURE_AT
→	TYPEARMURE	ARMURE_TYPEARMUREID_TYPEARMURE = TYPEARMUREID	FK_TYPEARMURE_AT

#### Unique keys

Columns		Name / Description
🔑	ID_ARMURE, ID_TYPEARMURE	SYS_C0040610

#### Uses

		Name
grid	ARMURE_TYPEARMURE	
→	ARMURE	
→	TYPEARMURE	

### 3.1.5. Table: CLASSE\_TYPEARME

#### Columns

		Name	Data type	Description / Attributes
		NOM_CLASSE	VARCHAR2(255 BYTE)	References: CLASSE
		ID_TYPEARME	NUMBER(5, 0)	References: TYPEARME

#### Links to

Table		Join	Title / Name / Description
→	CLASSE	CLASSE_TYPEARME NOM_CLASSE = CLASSENOM	FK_CLASSE_CLASSE_TYPEARME
→	TYPEARME	CLASSE_TYPEARME ID_TYPEARME = TYPEARMEID	FK_TYPEARME_CLASSE_TYPEARME

#### Unique keys

Columns		Name / Description
	NOM_CLASSE, ID_TYPEARME	SYS_C0040622

#### Uses

Name	
	CLASSE_TYPEARME
→	CLASSE
→	TYPEARME

### 3.1.6. Table: CLASSE\_TYPEARMURE

#### Columns

		Name	Data type	Description / Attributes
目	🔑	NOM_CLASSE	VARCHAR2(255 BYTE)	References: CLASSE
目	🔑	ID_TYPEARMURE	NUMBER(5, 0)	References: TYPEARMURE

#### Links to

Table		Join	Title / Name / Description
→	CLASSE	CLASSE_TYPEARMURE NOM_CLASSE = CLASSENOM	FK_CLASSE_CT
→	TYPEARMURE	CLASSE_TYPEARMURE ID_TYPEARMURE = TYPEARMUREID	FK_TYPEARMURE_CT

#### Unique keys

Columns		Name / Description
🔑	NOM_CLASSE, ID_TYPEARMURE	SYS_C0040625

#### Uses

		Name
grid	CLASSE_TYPEARMURE	
→	CLASSE	
→	TYPEARMURE	

### 3.1.7. Table: CONSOMMABLE

#### Columns

		Name	Data type	Description / Attributes
█	█	ID	NUMBER	<b>References:</b> DIVERS
█		EFFET	VARCHAR2(255 BYTE)	<b>Nullable</b>
█		ID_TYPECONSOMMABLE	NUMBER(5, 0)	<b>Nullable</b> <b>References:</b> TYPECONSOMMABLE

#### Links to

Table		Join	Title / Name / Description
→	DIVERS	<b>CONSOMMABLEID</b> = DIVERSID	FK_CONSUMMABLE_DIVERS
→	TYPECONSOMMABLE	<b>CONSOMMABLEID_TYPECONSOMMABLE</b> = TYPECONSOMMABLEID	FK_CONSUMMABLE_TYPECONS

#### Unique keys

Columns		Name / Description
█	ID	SYS_C0040529

#### Uses

		Name
█	CONSOMMABLE	
→	DIVERS	
→	TYPECONSOMMABLE	

#### Used By

		Name
█	CONSOMMABLE	
⊕	V_CONSUMMABLE	
⚡	DELETE_CONSUMMABLE_TRIGGER	
⚡	NEW_CONSUMMABLE_TRIGGER	
⚡	UPDATE_CONSUMMABLE_TRIGGER	

### 3.1.8. Table: DIVERS

#### Columns

	Name	Data type	Description / Attributes
	ID	NUMBER	References: ITEM

#### Links to

	Table	Join	Title / Name / Description
	ITEM	DIVERSID = ITEMID	FK_DIVERS_ITEM

#### Linked from

	Table	Join	Title / Name / Description
	CONSOMMABLE	DIVERSID = CONSOMMABLEID	FK_CONSUMMABLE_DIVERS
	RESSOURCE	DIVERSID = RESSOURCEID	FK_RESSOURCE_DIVERS

#### Unique keys

	Columns	Name / Description
	ID	SYS_C0040527

#### Uses

	Name
	DIVERS
	ITEM

#### Used By

	Name
	DIVERS
	V_DIVERS
	DELETE_DIVERS_TRIGGER
	NEW_DIVERS_TRIGGER
	CONSOMMABLE
	RESSOURCE

### 3.1.9. Table: EVENEMENT\_ITEM

#### Columns

		Name	Data type	Description / Attributes
目	🔑	ID_ITEM	NUMBER(5, 0)	References: ITEM
目	🔑	ID_EVENEMENT	NUMBER(5, 0)	References: EVENEMENT

#### Links to

Table		Join	Title / Name / Description
→	EVENEMENT	EVENEMENT_ITEMID_EVENEMENT = EVENEMENTID	FK_EVENEMENT_EVENEMENT_ITEM
→	ITEM	EVENEMENT_ITEMID_ITEM = ITEMID	FK_ITEM_EVENEMENT_ITEM

#### Unique keys

Columns		Name / Description
🔑	ID_ITEM, ID_EVENEMENT	SYS_C0040652

#### Uses

		Name
█	EVENEMENT_ITEM	
→	EVENEMENT	
→	ITEM	

### 3.1.10. Table: INVENTAIRE

#### Columns

		Name	Data type	Description / Attributes
目	🔑	ID_PERSONNAGE	NUMBER(5, 0)	References: PERSONNAGE
目	🔑	ID_ITEM	NUMBER(5, 0)	References: ITEM

#### Links to

Table		Join	Title / Name / Description
→	ITEM	INVENTAIREID_ITEM = ITEMID	FK_ITEM_INVENTAIRE
→	PERSONNAGE	INVENTAIREID_PERSONNAGE = PERSONNAGEID	FK_PERSONNAGE_INVENTAIRE

#### Unique keys

Columns		Name / Description
🔑	ID_PERSONNAGE, ID_ITEM	SYS_C0040646

#### Uses

		Name
grid	INVENTAIRE	
→	ITEM	
→	PERSONNAGE	

### 3.1.11. Table: ITEM

#### Columns

		Name	Data type	Description / Attributes
█	█	ID	NUMBER	
█		NOM	VARCHAR2(255 BYTE)	
█		DESCRIPTION	CLOB	Nullable

#### Linked from

Table		Join	Title / Name / Description
←	APPARENCE_ITEM	ITEMID = APPARENCE_ITEMID_ITEM	FK_ITEM_AI
←	APPARENCEPOSSIBLE_ITEM	ITEMID = APPARENCEPOSSIBLE_ITEMID_ITEM	FK_ITEM_API
←	ARME	ITEMID = ARMEID	FK_ARME_ITEM
←	ARMURE	ITEMID = ARMUREID	FK_ARMURE_ITEM
←	DIVERS	ITEMID = DIVERSID	FK_DIVERS_ITEM
←	EVENEMENT_ITEM	ITEMID = EVENEMENT_ITEMID_ITEM	FK_ITEM_EVENEMENT_ITEM
←	INVENTAIRE	ITEMID = INVENTAIREID_ITEM	FK_ITEM_INVENTAIRE
←	ITEM_RECETTE	ITEMID = ITEM_RECETTEID_ITEM	FK_ITEM_ITEM_RECETTE
←	ITEMQUETE	ITEMID = ITEMQUETEID	FK_ITEMQUETE_ITEM
←	OUTIL	ITEMID = OUTILID	FK_OUTIL_ITEM
←	RECETTE	ITEMID = RECETTEIDITEM	FK_RECETTE_ITEM

#### Unique keys

Columns		Name / Description
█	ID	SYS_C0040518

#### Triggers

Name		When	Description
⚡	NEW_ITEM_TRIGGER	Before Insert	
<pre>TRIGGER New_Item_trigger BEFORE INSERT ON ITEM FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := ITEM_seq.NEXTVAL; END;</pre>			

#### Uses

Name	
⚡	NEW_ITEM_TRIGGER
█	ITEM

#### Used By

Name	
█	ITEM

Name
V_ARME
V_ARMURE
V_DIVERS
V_ITEMQUETE
V_OUTIL
⚡ DELETE_ARME_TRIGGER
⚡ DELETE_ARMURE_TRIGGER
⚡ DELETE_DIVERS_TRIGGER
⚡ DELETE_ITEMQUETE_TRIGGER
⚡ DELETE_OUTIL_TRIGGER
⚡ NEW_ARME_TRIGGER
⚡ NEW_ARMURE_TRIGGER
⚡ NEW_DIVERS_TRIGGER
⚡ NEW_ITEM_TRIGGER
⚡ NEW_ITEMQUETE_TRIGGER
⚡ NEW_OUTIL_TRIGGER
⚡ UPDATE_ARME_TRIGGER
⚡ UPDATE_ARMURE_TRIGGER
⚡ UPDATE_DIVERS_TRIGGER
⚡ UPDATE_ITEMQUETE_TRIGGER
⚡ UPDATE_OUTIL_TRIGGER
→ APPARENCE_ITEM
→ APPARENCEPOSSIBLE_ITEM
→ ARME
→ ARMURE
→ DIVERS
→ EVENEMENT_ITEM
→ INVENTAIRE
→ ITEM_RECETTE
→ ITEMQUETE
→ OUTIL
→ RECETTE

### 3.1.12. Table: ITEM\_RECETTE

#### Columns

		Name	Data type	Description / Attributes
█	█	ID_ITEM	NUMBER(5, 0)	<b>References:</b> ITEM
█	█	ID_RECETTE	NUMBER(5, 0)	<b>References:</b> RECETTE
█		QUANTITE	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 1

#### Links to

Table		Join	Title / Name / Description
→	ITEM	ITEM_RECETTE.ID_ITEM = ITEM.ID	FK_ITEM_ITEM_RECETTE
→	RECETTE	ITEM_RECETTE.ID_RECETTE = RECETTE.ID	FK_RECETTE_ITEM_RECETTE

#### Unique keys

Columns		Name / Description
█	ID_ITEM, ID_RECETTE	SYS_C0040619

#### Uses

		Name
█	ITEM_RECETTE	
→	ITEM	
→	RECETTE	

### 3.1.13. Table: ITEMQUETE

#### Columns

	Name	Data type	Description / Attributes
	ID	NUMBER	References: ITEM

#### Links to

	Table	Join	Title / Name / Description
	ITEM	ITEMQUETEID = ITEMID	FK_ITEMQUETE_ITEM

#### Linked from

	Table	Join	Title / Name / Description
	ITEMQUETE_QUETE	ITEMQUETEID = ITEMQUETE_QUETEID_ITEM	FK_ITEMQUETE_ITEMQUETE_QUETE

#### Unique keys

	Columns	Name / Description
	ID	SYS_C0040523

#### Uses

	Name
	ITEMQUETE
	ITEM

#### Used By

	Name
	ITEMQUETE
	V_ITEMQUETE
	DELETE_ITEMQUETE_TRIGGER
	NEW_ITEMQUETE_TRIGGER
	ITEMQUETE_QUETE

### 3.1.14. Table: ITEMQUETE\_QUETE

#### Columns

		Name	Data type	Description / Attributes
		ID_ITEM	NUMBER(5, 0)	References: ITEMQUETE
		ID_QUETE	NUMBER(5, 0)	References: QUETE

#### Links to

Table		Join	Title / Name / Description
→	ITEMQUETE	ITEMQUETE_QUETEID_ITEM = ITEMQUETEID	FK_ITEMQUETE_ITEMQUETE_QUETE
→	QUETE	ITEMQUETE_QUETEID_QUETE = QUETEID	FK_QUETE_ITEMQUETE_QUETE

#### Unique keys

Columns		Name / Description
	ID_ITEM, ID_QUETE	SYS_C0040675

#### Uses

		Name
	ITEMQUETE_QUETE	
→	ITEMQUETE	
→	QUETE	

### 3.1.15. Table: METIER\_OUTIL

#### Columns

		Name	Data type	Description / Attributes
目	键	ID_METIER	NUMBER(5, 0)	References: METIER
目	键	ID_OUTIL	NUMBER(5, 0)	References: OUTIL

#### Links to

Table		Join	Title / Name / Description
→	METIER	METIER_OUTILID_METIER = METIERID	FK_METIER_MO
→	OUTIL	METIER_OUTILID_OUTIL = OUTILID	FK_OUTIL_MO

#### Unique keys

Columns		Name / Description
键	ID_METIER, ID_OUTIL	SYS_C0040616

#### Uses

		Name
表	METIER_OUTIL	
→	METIER	
→	OUTIL	

### 3.1.16. Table: OUTIL

#### Columns

	Name	Data type	Description / Attributes
	ID	NUMBER	References: ITEM

#### Links to

	Table	Join	Title / Name / Description
	ITEM	OUTILID = ITEMID	FK_OUTIL_ITEM

#### Linked from

	Table	Join	Title / Name / Description
	METIER_OUTIL	OUTILID = METIER_OUTILID_OUTIL	FK_OUTIL_MO

#### Unique keys

	Columns	Name / Description
	ID	SYS_C0040525

#### Uses

	Name
	OUTIL
	ITEM

#### Used By

	Name
	OUTIL
	V_OUTIL
	DELETE_OUTIL_TRIGGER
	NEW_OUTIL_TRIGGER
	METIER_OUTIL

### 3.1.17. Table: RESSOURCE

#### Columns

		Name	Data type	Description / Attributes
		ID	NUMBER	<b>References:</b> DIVERS
		ID_TYPERESSOURCE	NUMBER(5, 0)	<b>Nullable</b> <b>References:</b> TYPERESSOURCE

#### Links to

Table		Join	Title / Name / Description
→	DIVERS	RESSOURCEID = DIVERSID	FK_RESSOURCE_DIVERS
→	TYPERESSOURCE	RESSOURCEID_TYPERESSOURCE = TYPERESSOURCEID	FK_RESSOURCE_TYPERES

#### Unique keys

Columns		Name / Description
	ID	SYS_C0040532

#### Uses

		Name
		RESSOURCE
→	DIVERS	
→	TYPERESSOURCE	

#### Used By

		Name
		RESSOURCE
		V_RESSOURCE
		DELETE_RESSOURCE_TRIGGER
		NEW_RESSOURCE_TRIGGER
		UPDATE_RESSOURCE_TRIGGER

### 3.1.18. Table: TYPEARME

#### Columns

		Name	Data type	Description / Attributes
█	█	ID	NUMBER	
█		NOM	VARCHAR2(255 BYTE)	
█		DESCRIPTION	CLOB	Nullable

#### Linked from

Table		Join	Title / Name / Description
→	ARME_TYPEARME	<b>TYPEARMEID</b> = ARME_TYPEARMEID_TYPEARME	FK_TYPEARME_AT
→	CLASSE_TYPEARME	<b>TYPEARMEID</b> = CLASSE_TYPEARMEID_TYPEARME	FK_TYPEARME_CLASSE_TYPEARME

#### Unique keys

Columns		Name / Description
█	ID	SYS_C0040538

#### Triggers

Name		When	Description
⚡	NEW_TYPEARME_TRIGGER	Before Insert	
<pre>TRIGGER New_Typearme_trigger BEFORE INSERT ON TYPEARME FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := TYPEARME_seq.NEXTVAL; END;</pre>			

#### Uses

Name	
⚡	NEW_TYPEARME_TRIGGER
█	TYPEARME

#### Used By

Name	
█	TYPEARME
⚡	NEW_TYPEARME_TRIGGER
→	ARME_TYPEARME
→	CLASSE_TYPEARME

### 3.1.19. Table: TYPEARMURE

#### Columns

		Name	Data type	Description / Attributes
█	█	ID	NUMBER	
█		NOM	VARCHAR2(255 BYTE)	
█		DESCRIPTION	CLOB	Nullable

#### Linked from

Table		Join	Title / Name / Description
→	ARMURE_TYPEARMURE	<b>TYPEARMUREID</b> = ARMURE_TYPEARMUREID_TYPEARMURE	FK_TYPEARMURE_AT
→	CLASSE_TYPEARMURE	<b>TYPEARMUREID</b> = CLASSE_TYPEARMUREID_TYPEARMURE	FK_TYPEARMURE_CT

#### Unique keys

Columns		Name / Description
█	ID	SYS_C0040536

#### Triggers

Name		When	Description
⚡	NEW_TYPEARMURE_TRIGGER	Before Insert	
<pre>TRIGGER New_Typearmure_trigger BEFORE INSERT ON TYPEARMURE FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := TYPEARMURE_seq.NEXTVAL; END;</pre>			

#### Uses

		Name
⚡	NEW_TYPEARMURE_TRIGGER	
█	TYPEARMURE	

#### Used By

		Name
█	TYPEARMURE	
⚡	NEW_TYPEARMURE_TRIGGER	
→	ARMURE_TYPEARMURE	
→	CLASSE_TYPEARMURE	

### 3.1.20. Table: TYPECONSUMMABLE

#### Columns

		Name	Data type	Description / Attributes
█	█	ID	NUMBER	
█		NOM	VARCHAR2(255 BYTE)	
█		DESCRIPTION	CLOB	Nullable

#### Linked from

Table		Join	Title / Name / Description
→	CONSOMMABLE	TYPECONSUMMABLEID = CONSOMMABLEID_TYPECONSUMMABLE	FK_CONSUMMABLE_TYPECONS

#### Unique keys

Columns		Name / Description
█	ID	SYS_C0040516

#### Triggers

Name		When	Description
⚡	NEW_TYPECONSUMMABLE_TRIGGER	Before Insert	
<pre>TRIGGER New_Typeconsummable_trigger BEFORE INSERT ON TYPECONSUMMABLE FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := TYPECONSUMMABLE_seq.NEXTVAL; END;</pre>			

#### Uses

		Name
⚡	NEW_TYPECONSUMMABLE_TRIGGER	
█	TYPECONSUMMABLE	

#### Used By

		Name
█	TYPECONSUMMABLE	
⚡	NEW_TYPECONSUMMABLE_TRIGGER	
→	CONSOMMABLE	

### 3.1.21. Table: TYPERESSOURCE

#### Columns

		Name	Data type	Description / Attributes
		ID	NUMBER	
		NOM	VARCHAR2(255 BYTE)	
		DESCRIPTION	CLOB	Nullable

#### Linked from

Table		Join	Title / Name / Description
	METIER_TYPERESSOURCE	TYPERESSOURCEID = METIER_TYPERESSOURCEID_TYPERESOURCE	FK_RESSOURCE_METIER_RESSOURCE
	RESSOURCE	TYPERESSOURCEID = RESSOURCEID_TYPERESSOURCE	FK_RESSOURCE_TYPERES

#### Unique keys

Columns		Name / Description
	ID	SYS_C0040514

#### Triggers

Name		When	Description
	NEW_TYPERESSOURCE_TRIGGER	Before Insert	
<pre>TRIGGER New_Typeressource_trigger   BEFORE INSERT ON TYPERESSOURCE   FOR EACH ROW   BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := TYPERESSOURCE_seq.NEXTVAL;   END;</pre>			

#### Uses

Name	
	NEW_TYPERESSOURCE_TRIGGER
	TYPERESSOURCE

#### Used By

Name	
	TYPERESSOURCE
	NEW_TYPERESSOURCE_TRIGGER
	METIER_TYPERESSOURCE
	RESSOURCE

## 3.2. Views

### 3.2.1. View: V\_ARME

#### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	DESCRIPTION	CLOB	Nullable
█	DEGATSBASE	NUMBER(5, 0)	Nullable

#### Triggers

	Name	When	Description
⚡	DELETE_ARME_TRIGGER	Instead Of Delete	<pre>TRIGGER Delete_Arme_trigger INSTEAD OF DELETE ON V_ARME FOR EACH ROW BEGIN     -- Suppression de l'armure     DELETE FROM ARME     WHERE Id = :OLD.Id;      -- Suppression de l'item     DELETE FROM ITEM     WHERE Id = :OLD.Id; END;</pre>
⚡	NEW_ARME_TRIGGER	Instead Of Insert	<pre>TRIGGER New_Arme_trigger INSTEAD OF INSERT ON V_ARME FOR EACH ROW DECLARE     New_Id NUMBER; BEGIN     -- Nouvel item     INSERT INTO ITEM(Nom, Description)     VALUES (:NEW.Nom, :NEW.Description);      -- Récupérer l'ID généré     SELECT MAX(Id) INTO New_Id FROM ITEM;      -- Nouvelle arme     INSERT INTO ARME(Id, DegatsBase)     VALUES (New_Id, :NEW.DegatsBase); END;</pre>
⚡	UPDATE_ARME_TRIGGER	Instead Of Update	<pre>TRIGGER Update_Arme_trigger INSTEAD OF UPDATE ON V_ARME FOR EACH ROW BEGIN     -- Mise à jour de l'item     UPDATE ITEM     SET Nom = :NEW.Nom,         Description = :NEW.Description     WHERE Id = :NEW.Id;      -- Mise à jour de l'armure     UPDATE ARME     SET DegatsBase = :NEW.DegatsBase     WHERE Id = :NEW.Id; END;</pre>

## Uses

Name
V_ARME
ARME
ITEM
DELETE_ARME_TRIGGER
ARME
ITEM
V_ARME
NEW_ARME_TRIGGER
ARME
ITEM
V_ARME
UPDATE_ARME_TRIGGER
ARME
ITEM
V_ARME

## Used By

Name
V_ARME
DELETE_ARME_TRIGGER
NEW_ARME_TRIGGER
UPDATE_ARME_TRIGGER

## Script

```
SELECT I.Id, I.Nom, I.Description, A.DegatsBase
  FROM ITEM I
  JOIN ARME A ON I.Id = A.Id
```

### 3.2.2. View: V\_ARMURE

#### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	DESCRIPTION	CLOB	<b>Nullable</b>
█	POINTDEVIEBASE	NUMBER(5, 0)	<b>Nullable</b>
█	ARMUREBASE	NUMBER(5, 0)	<b>Nullable</b>

#### Triggers

	Name	When	Description
⚡	DELETE_ARMURE_TRIGGER	Instead Of Delete	<pre>TRIGGER Delete_Armure_trigger   INSTEAD OF DELETE ON V_ARMURE   FOR EACH ROW   BEGIN     -- Suppression de l'armure     DELETE FROM ARMURE     WHERE Id = :OLD.Id;      -- Suppression de l'item     DELETE FROM ITEM     WHERE Id = :OLD.Id;   END;</pre>
⚡	NEW_ARMURE_TRIGGER	Instead Of Insert	<pre>TRIGGER New_Armure_trigger   INSTEAD OF INSERT ON V_ARMURE   FOR EACH ROW   DECLARE     New_Id NUMBER;   BEGIN     -- Nouvel item     INSERT INTO ITEM(Nom, Description)     VALUES (:NEW.Nom, :NEW.Description);      -- Récupérer l'ID généré     SELECT MAX(Id) INTO New_Id FROM ITEM;      -- Nouvelle armure     INSERT INTO ARMURE(Id, PointDeVieBase, ArmureBase)     VALUES (New_Id, :NEW.PointDeVieBase, :NEW.ArmureBase);   END;</pre>
⚡	UPDATE_ARMURE_TRIGGER	Instead Of Update	<pre>TRIGGER Update_Armure_trigger   INSTEAD OF UPDATE ON V_ARMURE   FOR EACH ROW   BEGIN     -- Mise à jour de l'item     UPDATE ITEM     SET Nom = :NEW.Nom,         Description = :NEW.Description     WHERE Id = :NEW.Id;      -- Mise à jour de l'armure     UPDATE ARMURE     SET PointDeVieBase = :NEW.PointDeVieBase,         ArmureBase = :NEW.ArmureBase     WHERE Id = :NEW.Id;   END;</pre>

#### Uses

	Name
█	V_ARMURE

Name
ARMURE
ITEM
<b>DELETE_ARMURE_TRIGGER</b>
ARMURE
ITEM
V_ARMURE
<b>NEW_ARMURE_TRIGGER</b>
ARMURE
ITEM
V_ARMURE
<b>UPDATE_ARMURE_TRIGGER</b>
ARMURE
ITEM
V_ARMURE

### Used By

Name
V_ARMURE
DELETE_ARMURE_TRIGGER
NEW_ARMURE_TRIGGER
UPDATE_ARMURE_TRIGGER

### Script

```
SELECT I.Id, I.Nom, I.Description, A.PointDeVieBase, A.ArmureBase
  FROM ITEM I
  JOIN ARMURE A ON I.Id = A.Id
```

### 3.2.3. View: V\_CONSUMMABLE

#### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	DESCRIPTION	CLOB	<b>Nullable</b>
█	EFFET	VARCHAR2(255 BYTE)	<b>Nullable</b>
█	ID_TYPECONSOMMABLE	NUMBER(5, 0)	<b>Nullable</b>

#### Triggers

	Name	When	Description
⚡	DELETE_CONSUMMABLE_TRIGGER	Instead Of Delete	<pre>TRIGGER Delete_Consummable_trigger   INSTEAD OF DELETE ON V_CONSUMMABLE   FOR EACH ROW   BEGIN     -- Suppression du consommable     DELETE FROM CONSOMMABLE     WHERE Id = :OLD.Id;      -- Suppression ddu divers     DELETE FROM V_DIVERS     WHERE Id = :OLD.Id;   END;</pre>
⚡	NEW_CONSUMMABLE_TRIGGER	Instead Of Insert	<pre>TRIGGER New_Consummable_trigger   INSTEAD OF INSERT ON V_CONSUMMABLE   FOR EACH ROW   DECLARE     New_Id NUMBER;   BEGIN     -- Nouvel Divers     INSERT INTO V_DIVERS(Nom, Description)     VALUES (:NEW.Nom, :NEW.Description);      -- Récupérer l'ID généré     SELECT MAX(Id) INTO New_Id FROM V_DIVERS;      -- Nouvelle Consommable     INSERT INTO CONSOMMABLE(Id, Effet, Id_Typeconsommable)     VALUES (New_Id, :NEW.Effet, :NEW.Id_Typeconsommable);   END;</pre>
⚡	UPDATE_CONSUMMABLE_TRIGGER	Instead Of Update	<pre>TRIGGER Update_Consummable_trigger   INSTEAD OF UPDATE ON V_CONSUMMABLE   FOR EACH ROW   BEGIN     -- Mise à jour du divers     UPDATE V_DIVERS     SET Nom = :NEW.Nom,         Description = :NEW.Description     WHERE Id = :NEW.Id;      -- Mise à jour du consommable     UPDATE CONSOMMABLE     SET Effet = :NEW.Effet,         Id_Typeconsommable = :NEW.Id_Typeconsommable     WHERE Id = :NEW.Id;   END;</pre>

## Uses

Name
CONSOMMABLE
V_DIVERS
DELETE_CONSOMMABLE_TRIGGER
CONSOMMABLE
V_CONSOMMABLE
V_DIVERS
NEW_CONSOMMABLE_TRIGGER
CONSOMMABLE
V_CONSOMMABLE
V_DIVERS
UPDATE_CONSOMMABLE_TRIGGER
CONSOMMABLE
V_CONSOMMABLE
V_DIVERS

## Used By

Name
V_CONSOMMABLE
DELETE_CONSOMMABLE_TRIGGER
NEW_CONSOMMABLE_TRIGGER
UPDATE_CONSOMMABLE_TRIGGER

## Script

```
SELECT I.Id, I.Nom, I.Description, C.Effet, C.Id_Typeconsommable
  FROM V_DIVERS I
  JOIN CONSOMMABLE C ON I.Id = C.Id
```

### 3.2.4. View: V\_DIVERS

#### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	DESCRIPTION	CLOB	Nullable

#### Triggers

	Name	When	Description
⚡	DELETE_DIVERS_TRIGGER	Instead Of Delete	<pre>TRIGGER Delete_Divers_trigger   INSTEAD OF DELETE ON V_DIVERS   FOR EACH ROW   BEGIN     -- Suppression de l'armure     DELETE FROM DIVERS     WHERE Id = :OLD.Id;      -- Suppression de l'item     DELETE FROM ITEM     WHERE Id = :OLD.Id;   END;</pre>
⚡	NEW_DIVERS_TRIGGER	Instead Of Insert	<pre>TRIGGER New_Divers_trigger   INSTEAD OF INSERT ON V_DIVERS   FOR EACH ROW   DECLARE     New_Id NUMBER;   BEGIN     -- Nouvel item     INSERT INTO ITEM(Nom, Description)     VALUES (:NEW.Nom, :NEW.Description);      -- Récupérer l'ID généré     SELECT MAX(id) INTO New_Id FROM ITEM;      -- Nouvelle divers     INSERT INTO DIVERS(Id)     VALUES (New_Id);   END;</pre>
⚡	UPDATE_DIVERS_TRIGGER	Instead Of Update	<pre>TRIGGER Update_DIVERS_trigger   INSTEAD OF UPDATE ON V_DIVERS   FOR EACH ROW   BEGIN     -- Mise à jour de l'item     UPDATE ITEM     SET Nom = :NEW.Nom,         Description = :NEW.Description     WHERE Id = :NEW.Id;   END;</pre>

#### Uses

	Name
█	V_DIVERS
█	DIVERS
█	ITEM
⚡	DELETE_DIVERS_TRIGGER
█	DIVERS

Name	
ITEM	
V_DIVERS	
NEW_DIVERS_TRIGGER	
DIVERS	
ITEM	
V_DIVERS	
UPDATE_DIVERS_TRIGGER	
ITEM	
V_DIVERS	

## Used By

Name	
V_DIVERS	
V_CONSUMMABLE	
V_RESSOURCE	
DELETE_CONSUMMABLE_TRIGGER	
DELETE_DIVERS_TRIGGER	
DELETE_RESSOURCE_TRIGGER	
NEW_CONSUMMABLE_TRIGGER	
NEW_DIVERS_TRIGGER	
NEW_RESSOURCE_TRIGGER	
UPDATE_CONSUMMABLE_TRIGGER	
UPDATE_DIVERS_TRIGGER	
UPDATE_RESSOURCE_TRIGGER	

## Script

```
SELECT I.Id, I.Nom, I.Description
  FROM ITEM I
  JOIN DIVERS D ON I.Id = D.Id
```

### 3.2.5. View: V\_ITEMQUETE

#### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	DESCRIPTION	CLOB	Nullable

#### Triggers

	Name	When	Description
⚡	DELETE_ITEMQUETE_TRIGGER	Instead Of Delete	<pre>TRIGGER Delete_Itemquete_trigger INSTEAD OF DELETE ON V_ITEMQUETE FOR EACH ROW BEGIN     -- Suppression de l'itemquete     DELETE FROM ITEMQUETE     WHERE Id = :OLD.Id;      -- Suppression de l'item     DELETE FROM ITEM     WHERE Id = :OLD.Id; END;</pre>
⚡	NEW_ITEMQUETE_TRIGGER	Instead Of Insert	<pre>TRIGGER New_Itemquete_trigger INSTEAD OF INSERT ON V_ITEMQUETE FOR EACH ROW DECLARE     New_Id NUMBER; BEGIN     -- Nouvel item     INSERT INTO ITEM(Nom, Description)     VALUES (:NEW.Nom, :NEW.Description);      -- Récupérer l'ID généré     SELECT MAX(id) INTO New_Id FROM ITEM;      -- Nouvelle itemquete     INSERT INTO ITEMQUETE(Id)     VALUES (New_Id); END;</pre>
⚡	UPDATE_ITEMQUETE_TRIGGER	Instead Of Update	<pre>TRIGGER Update_Itemquete_trigger INSTEAD OF UPDATE ON V_ITEMQUETE FOR EACH ROW BEGIN     -- Mise à jour de l'item     UPDATE ITEM     SET Nom = :NEW.Nom,         Description = :NEW.Description     WHERE Id = :NEW.Id; END;</pre>

#### Uses

	Name
█	V_ITEMQUETE
█	ITEM
█	ITEMQUETE
⚡	DELETE_ITEMQUETE_TRIGGER
█	ITEM

## Name

ITEMQUETE

V\_ITEMQUETE

NEW\_ITEMQUETE\_TRIGGER

ITEM

ITEMQUETE

V\_ITEMQUETE

UPDATE\_ITEMQUETE\_TRIGGER

ITEM

V\_ITEMQUETE

## Used By

## Name

V\_ITEMQUETE

DELETE\_ITEMQUETE\_TRIGGER

NEW\_ITEMQUETE\_TRIGGER

UPDATE\_ITEMQUETE\_TRIGGER

## Script

```
SELECT I.Id, I.Nom, I.Description
  FROM ITEM I
  JOIN ITEMQUETE IQ ON I.Id = IQ.Id
```

### 3.2.6. View: V\_OUTIL

#### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	DESCRIPTION	CLOB	Nullable

#### Triggers

	Name	When	Description
⚡	DELETE_OUTIL_TRIGGER	Instead Of Delete	<pre>TRIGGER Delete_Outil_trigger   INSTEAD OF DELETE ON V_OUTIL   FOR EACH ROW   BEGIN     -- Suppression de l'armure     DELETE FROM OUTIL     WHERE Id = :OLD.Id;      -- Suppression de l'item     DELETE FROM ITEM     WHERE Id = :OLD.Id;   END;</pre>
⚡	NEW_OUTIL_TRIGGER	Instead Of Insert	<pre>TRIGGER New_Outil_trigger   INSTEAD OF INSERT ON V_OUTIL   FOR EACH ROW   DECLARE     New_Id NUMBER;   BEGIN     -- Nouvel item     INSERT INTO ITEM(Nom, Description)     VALUES (:NEW.Nom, :NEW.Description);      -- Récupérer l'ID généré     SELECT MAX(id) INTO New_Id FROM ITEM;      -- Nouvelle outil     INSERT INTO OUTIL(Id)     VALUES (New_Id);   END;</pre>
⚡	UPDATE_OUTIL_TRIGGER	Instead Of Update	<pre>TRIGGER Update_Outil_trigger   INSTEAD OF UPDATE ON V_OUTIL   FOR EACH ROW   BEGIN     -- Mise à jour de l'item     UPDATE ITEM     SET Nom = :NEW.Nom,         Description = :NEW.Description     WHERE Id = :NEW.Id;   END;</pre>

#### Uses

	Name
█	V_OUTIL
█	ITEM
█	OUTIL
⚡	DELETE_OUTIL_TRIGGER
█	ITEM

	Name
OUTIL	
V_OUTIL	
NEW_OUTIL_TRIGGER	
ITEM	
OUTIL	
V_OUTIL	
UPDATE_OUTIL_TRIGGER	
ITEM	
V_OUTIL	

### Used By

	Name
V_OUTIL	
DELETE_OUTIL_TRIGGER	
NEW_OUTIL_TRIGGER	
UPDATE_OUTIL_TRIGGER	

### Script

```
SELECT I.Id, I.Nom, I.Description
  FROM ITEM I
  JOIN OUTIL O ON I.Id = O.Id
```

### 3.2.7. View: V\_RESSOURCE

#### Columns

	Name	Data type	Description / Attributes
█	ID	NUMBER	
█	NOM	VARCHAR2(255 BYTE)	
█	DESCRIPTION	CLOB	<b>Nullable</b>
█	ID_TYPERESSOURCE	NUMBER(5, 0)	<b>Nullable</b>

#### Triggers

	Name	When	Description
⚡	DELETE_RESSOURCE_TRIGGER	Instead Of Delete	<pre>TRIGGER Delete_Ressource_trigger   INSTEAD OF DELETE ON V_RESSOURCE   FOR EACH ROW   BEGIN     -- Suppression du consommable     DELETE FROM RESSOURCE     WHERE Id = :OLD.Id;      -- Suppression ddu divers     DELETE FROM V_DIVERS     WHERE Id = :OLD.Id;   END;</pre>
⚡	NEW_RESSOURCE_TRIGGER	Instead Of Insert	<pre>TRIGGER New_Ressource_trigger   INSTEAD OF INSERT ON V_RESSOURCE   FOR EACH ROW   DECLARE     New_Id NUMBER;   BEGIN     -- Nouvel Divers     INSERT INTO V_DIVERS(Nom, Description)     VALUES (:NEW.Nom, :NEW.Description);      -- Récupérer l'ID généré     SELECT MAX(Id) INTO New_Id FROM V_DIVERS;      -- Nouvelle Ressource     INSERT INTO RESSOURCE(Id, Id_Typeressource)     VALUES (New_Id, :NEW.Id_Typeressource);   END;</pre>
⚡	UPDATE_RESSOURCE_TRIGGER	Instead Of Update	<pre>TRIGGER Update_Ressource_trigger   INSTEAD OF UPDATE ON V_RESSOURCE   FOR EACH ROW   BEGIN     -- Mise à jour du divers     UPDATE V_DIVERS     SET Nom = :NEW.Nom,         Description = :NEW.Description     WHERE Id = :NEW.Id;      -- Mise à jour de la ressource     UPDATE RESSOURCE     SET Id_Typeressource = :NEW.Id_Typeressource     WHERE Id = :NEW.Id;   END;</pre>

#### Uses

	Name
█	V_RESSOURCE
█	RESSOURCE

## Name

V\_DIVERS

DELETE\_RESSOURCE\_TRIGGER

RESSOURCE

V\_DIVERS

V\_RESOURCE

NEW\_RESSOURCE\_TRIGGER

RESSOURCE

V\_DIVERS

V\_RESOURCE

UPDATE\_RESSOURCE\_TRIGGER

RESSOURCE

V\_DIVERS

V\_RESOURCE

## Used By

## Name

V\_RESOURCE

DELETE\_RESSOURCE\_TRIGGER

NEW\_RESSOURCE\_TRIGGER

UPDATE\_RESSOURCE\_TRIGGER

## Script

```
SELECT I.Id, I.Nom, I.Description, R.Id_Typeressource
  FROM V_DIVERS I
  JOIN RESSOURCE R ON I.Id = R.Id
```



## 4. Other

### 4.1. Tables

#### 4.1.1. Table: APPARENCE

##### Columns

		Name	Data type	Description / Attributes
	ID	NUMBER		
	NOM	VARCHAR2(255 BYTE)		
	DESCRIPTION	CLOB	<b>Nullable</b>	
	IMAGE	VARCHAR2(255 BYTE)	<b>Nullable</b>	

##### Linked from

Table		Join	Title / Name / Description
→	APPARENCE_CLASSE	<b>APPARENCEID</b> = APPARENCE_CLASSEID_APPARENCE	FK_APPARENCE_AC
→	APPARENCE_EVENEMENT	<b>APPARENCEID</b> = APPARENCE_EVENEMENTID_APPARENCE	FK_APPARENCE_AE
→	APPARENCE_FACTION	<b>APPARENCEID</b> = APPARENCE_FACTIONID_APPARENCE	FK_APPARENCE_AF
→	APPARENCE_HAUTFAIT	<b>APPARENCEID</b> = APPARENCE_HAUTFAITID_APPARENCE	FK_APPARENCE_AH
→	APPARENCE_ITEM	<b>APPARENCEID</b> = APPARENCE_ITEMID_APPARENCE	FK_APPARENCE_AI
→	APPARENCE_LIEUX	<b>APPARENCEID</b> = APPARENCE_LIEUXID_APPARENCE	FK_APPARENCE_AL
→	APPARENCE_METIER	<b>APPARENCEID</b> = APPARENCE_METIERID_APPARENCE	FK_APPARENCE_AM
→	APPARENCE_PERSONNAGE	<b>APPARENCEID</b> = APPARENCE_PERSONNAGEID_APPARENCE	FK_APPARENCE_AP
→	APPARENCE_QUETE	<b>APPARENCEID</b> = APPARENCE_QUETEID_APPARENCE	FK_APPARENCE_AQ
→	APPARENCE_RACE	<b>APPARENCEID</b> = APPARENCE_RACEID_APPARENCE	FK_APPARENCE_AR
→	APPARENCE_SORT	<b>APPARENCEID</b> = APPARENCE_SORTID_APPARENCE	FK_APPARENCE_AS
→	APPARENCE_TITRE	<b>APPARENCEID</b> = APPARENCE_TITREID_APPARENCE	FK_APPARENCE_AT
→	APPARENCEPOSSIBLE_CLASSE	<b>APPARENCEID</b> = APPARENCEPOSSIBLE_CLASSEID_APPARENCE	FK_APPARENCE_AP_C
→	APPARENCEPOSSIBLE_EVENEMENT	<b>APPARENCEID</b> = APPARENCEPOSSIBLE_EVENEMENTID_APPARENCE	FK_APPARENCE_APE

Table		Join	Title / Name / Description
→ APPARENCEPOSSIBLE_FACTION	<b>APPARENCEID = APPARENCEPOSSIBLE_FACTIONID_APPARENCE</b>		FK_APPARENCE_APF
→ APPARENCEPOSSIBLE_HAUTFAIT	<b>APPARENCEID = APPARENCEPOSSIBLE_HAUTFAITID_APPARENCE</b>		FK_APPARENCEAPH
→ APPARENCEPOSSIBLE_ITEM	<b>APPARENCEID = APPARENCEPOSSIBLE_ITEMID_APPARENCE</b>		FK_APPARENCEAPI
→ APPARENCEPOSSIBLE_LIEUX	<b>APPARENCEID = APPARENCEPOSSIBLE_LIEUXID_APPARENCE</b>		FK_APPARENCEAPL
→ APPARENCEPOSSIBLE_METIER	<b>APPARENCEID = APPARENCEPOSSIBLE_METIERID_APPARENCE</b>		FK_APPARENCEAPM
→ APPARENCEPOSSIBLE_PERSONNAGE	<b>APPARENCEID = APPARENCEPOSSIBLE_PERSONNAGEID_APPARENCE</b>		FK_APPARENCEAPP
→ APPARENCEPOSSIBLE_QUETE	<b>APPARENCEID = APPARENCEPOSSIBLE_QUETEID_APPARENCE</b>		FK_APPARENCEAPQ
→ APPARENCEPOSSIBLE_RACE	<b>APPARENCEID = APPARENCEPOSSIBLE_RACEID_APPARENCE</b>		FK_APPARENCEAPR
→ APPARENCEPOSSIBLE_SORT	<b>APPARENCEID = APPARENCEPOSSIBLE_SORTID_APPARENCE</b>		FK_APPARENCEAPS
→ APPARENCEPOSSIBLE_TITRE	<b>APPARENCEID = APPARENCEPOSSIBLE_TITREID_APPARENCE</b>		FK_APPARENCEAPT

## Unique keys

Columns		Name / Description
🔑 ID	SYS_C0040556	

## Triggers

Name				When	Description
⚡ NEW_APPARENCE_TRIGGER		Before Insert			
<pre>TRIGGER New_Apparence_trigger BEFORE INSERT ON APPARENCE FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := APPARENCE_seq.NEXTVAL; END;</pre>					

## Uses

Name	
⚡ NEW_APPARENCE_TRIGGER	
grid APPARENCE	

## Used By

Name
APPARENCE
⚡ NEW_APPARENCE_TRIGGER
→ APPARENCE_CLASSE
→ APPARENCE_EVENEMENT
→ APPARENCE_FACTION
→ APPARENCE_HAUTFAIT
→ APPARENCE_ITEM
→ APPARENCE_LIEUX
→ APPARENCE_METIER
→ APPARENCE_PERSONNAGE
→ APPARENCE_QUETE
→ APPARENCE_RACE
→ APPARENCE_SORT
→ APPARENCE_TITRE
→ APPARENCEPOSSIBLE_CLASSE
→ APPARENCEPOSSIBLE_EVENEMENT
→ APPARENCEPOSSIBLE_FACTION
→ APPARENCEPOSSIBLE_HAUTFAIT
→ APPARENCEPOSSIBLE_ITEM
→ APPARENCEPOSSIBLE_LIEUX
→ APPARENCEPOSSIBLE_METIER
→ APPARENCEPOSSIBLE_PERSONNAGE
→ APPARENCEPOSSIBLE_QUETE
→ APPARENCEPOSSIBLE_RACE
→ APPARENCEPOSSIBLE_SORT
→ APPARENCEPOSSIBLE_TITRE

## 4.1.2. Table: APPARENCE\_CLASSE

### Columns

Name		Data type	Description / Attributes
目	🔑 NOM_CLASSE	VARCHAR2(255 BYTE)	References: APPARENCEPOSSIBLE_CLASSE, CLASSE
目	🔑 ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE, APPARENCEPOSSIBLE_CLASSE

### Links to

Table		Join	Title / Name / Description
→	APPARENCE	<b>APPARENCE_CLASSE</b> ID_APPARENCE = APPARENCEID	FK_APPARENCE_AC
→	APPARENCEPOSSIBLE_CLASSE	<b>APPARENCE_CLASSE</b> NOM_CLASSE = APPARENCEPOSSIBLE_CLASSENOM_CLASSE, <b>APPARENCE_CLASSE</b> ID_APPARENCE = APPARENCEPOSSIBLE_CLASSEID_APPARENCE	FK_GOODAC
→	CLASSE	<b>APPARENCE_CLASSE</b> NOM_CLASSE = CLASSENOM	FK_CLASSE_AC

### Unique keys

Columns		Name / Description
🔑	NOM_CLASSE, ID_APPARENCE	SYS_C0040716

### Uses

Name	
grid	APPARENCE_CLASSE
→	APPARENCE
→	APPARENCEPOSSIBLE_CLASSE
→	CLASSE

### 4.1.3. Table: APPARENCE\_EVENEMENT

#### Columns

		Name	Data type	Description / Attributes
		ID_EVENEMENT	NUMBER(5, 0)	<b>References:</b> APPARENCEPOSSIBLE_EVENEMENT, EVENEMENT
		ID_APPARENCE	NUMBER(5, 0)	<b>References:</b> APPARENCE, APPARENCEPOSSIBLE_EVENEMENT

#### Links to

Table		Join	Title / Name / Description
→	APPARENCE	<b>APPARENCE_EVENEMENTID_APPARENCE</b> = APPARENCEID	FK_APPARENCE_AE
→	APPARENCEPOSSIBLE_EVENEMENT	<b>APPARENCE_EVENEMENTID_EVENEMENT</b> = APPARENCEPOSSIBLE_EVENEMENTID_EVENEMENT, <b>APPARENCE_EVENEMENTID_APPARENCE</b> = APPARENCEPOSSIBLE_EVENEMENTID_APPARENCE	FK_GOODAE
→	EVENEMENT	<b>APPARENCE_EVENEMENTID_EVENEMENT</b> = EVENEMENTID	FK_EVENEMENT_AE

#### Unique keys

Columns		Name / Description
	ID_EVENEMENT, ID_APPARENCE	SYS_C0040744

#### Uses

		Name
		APPARENCE_EVENEMENT
	→	APPARENCE
	→	APPARENCEPOSSIBLE_EVENEMENT
	→	EVENEMENT

#### 4.1.4. Table: APPARENCE\_FACTION

##### Columns

		Name	Data type	Description / Attributes
		ID_FACTION	NUMBER(5, 0)	<b>References:</b> APPARENCEPOSSIBLE_FACTION, FACTION
		ID_APPARENCE	NUMBER(5, 0)	<b>References:</b> APPARENCE, APPARENCEPOSSIBLE_FACTION

##### Links to

Table		Join	Title / Name / Description
→	APPARENCE	<b>APPARENCE_FACTION</b> ID_APPARENCE = APPARENCEID	FK_APPARENCE_AF
→	APPARENCEPOSSIBLE_FACTION	<b>APPARENCE_FACTION</b> ID_FACTION = APPARENCEPOSSIBLE_FACTIONID_FACTION, <b>APPARENCE_FACTION</b> ID_APPARENCE = APPARENCEPOSSIBLE_FACTIONID_APPARENCE	FK_GOOD_AF
→	FACTION	<b>APPARENCE_FACTION</b> ID_FACTION = FACTIONID	FK_FACTION_AF

##### Unique keys

Columns		Name / Description
	ID_FACTION, ID_APPARENCE	SYS_C0040681

##### Uses

Name	
	APPARENCE_FACTION
→	APPARENCE
→	APPARENCEPOSSIBLE_FACTION
→	FACTION

## 4.1.5. Table: APPARENCE\_HAUTFAIT

### Columns

		Name	Data type	Description / Attributes
		ID_HAUTFAIT	NUMBER(5, 0)	<b>References:</b> APPARENCEPOSSIBLE_HAUTFAIT, HAUTFAIT
		ID_APPARENCE	NUMBER(5, 0)	<b>References:</b> APPARENCE, APPARENCEPOSSIBLE_HAUTFAIT

### Links to

Table		Join	Title / Name / Description
→	APPARENCE	<b>APPARENCE_HAUTFAITID_APPARENCE</b> = APPARENCEID	FK_APPARENCE_AH
→	APPARENCEPOSSIBLE_HAUTFAIT	<b>APPARENCE_HAUTFAITID_HAUTFAIT</b> = APPARENCEPOSSIBLE_HAUTFAITID_HAUTFAIT, <b>APPARENCE_HAUTFAITID_APPARENCE</b> = APPARENCEPOSSIBLE_HAUTFAITID_APPARENCE	FK_GOOD_AH
→	HAUTFAIT	<b>APPARENCE_HAUTFAITID_HAUTFAIT</b> = HAUTFAITID	FK_HAUTFAIT_AH

### Unique keys

Columns		Name / Description
	ID_HAUTFAIT, ID_APPARENCE	SYS_C0040688

### Uses

		Name
		APPARENCE_HAUTFAIT
	→	APPARENCE
	→	APPARENCEPOSSIBLE_HAUTFAIT
	→	HAUTFAIT

## 4.1.6. Table: APPARENCE\_ITEM

### Columns

		Name	Data type	Description / Attributes
		ID_ITEM	NUMBER(5, 0)	References: APPARENCEPOSSIBLE_ITEM, ITEM
		ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE, APPARENCEPOSSIBLE_ITEM

### Links to

Table		Join	Title / Name / Description
→	APPARENCE	<b>APPARENCE_ITEMID_APPARENCE</b> = APPARENCEID	FK_APPARENCE_AI
→	APPARENCEPOSSIBLE_ITEM	<b>APPARENCE_ITEMID_ITEM</b> = APPARENCEPOSSIBLE_ITEMID_ITEM, <b>APPARENCE_ITEMID_APPARENCE</b> = APPARENCEPOSSIBLE_ITEMID_APPARENCE	FK_GOODAI
→	ITEM	<b>APPARENCE_ITEMID_ITEM</b> = ITEMID	FK_ITEM_AI

### Unique keys

Columns		Name / Description
	ID_ITEM, ID_APPARENCE	SYS_C0040737

### Uses

		Name
		APPARENCE_ITEM
	→	APPARENCE
	→	APPARENCEPOSSIBLE_ITEM
	→	ITEM

## 4.1.7. Table: APPARENCE\_LIEUX

### Columns

		Name	Data type	Description / Attributes
		ID_LIEUX	NUMBER(5, 0)	<b>References:</b> APPARENCEPOSSIBLE_LIEUX, LIEUX
		ID_APPARENCE	NUMBER(5, 0)	<b>References:</b> APPARENCE, APPARENCEPOSSIBLE_LIEUX

### Links to

Table		Join	Title / Name / Description
→	APPARENCE	<b>APPARENCE_LIEUXID_APPARENCE</b> = APPARENCEID	FK_APPARENCE_AL
→	APPARENCEPOSSIBLE_LIEUX	<b>APPARENCE_LIEUXID_LIEUX</b> = APPARENCEPOSSIBLE_LIEUXID_LIEUX, <b>APPARENCE_LIEUXID_APPARENCE</b> = APPARENCEPOSSIBLE_LIEUXID_APPARENCE	FK_GOODAL
→	LIEUX	<b>APPARENCE_LIEUXID_LIEUX</b> = LIEUXID	FK_LIEUX_AL

### Unique keys

Columns		Name / Description
	ID_LIEUX, ID_APPARENCE	SYS_C0040758

### Uses

		Name
		APPARENCE_LIEUX
	→	APPARENCE
	→	APPARENCEPOSSIBLE_LIEUX
	→	LIEUX

## 4.1.8. Table: APPARENCE\_METIER

### Columns

		Name	Data type	Description / Attributes
		ID_METIER	NUMBER(5, 0)	<b>References:</b> APPARENCEPOSSIBLE_METIER, METIER
		ID_APPARENCE	NUMBER(5, 0)	<b>References:</b> APPARENCE, APPARENCEPOSSIBLE_METIER

### Links to

Table		Join	Title / Name / Description
→	APPARENCE	<b>APPARENCE_METIERID_APPARENCE</b> = APPARENCEID	FK_APPARENCE_AM
→	APPARENCEPOSSIBLE_METIER	<b>APPARENCE_METIERID_METIER</b> = APPARENCEPOSSIBLE_METIERID_METIER, <b>APPARENCE_METIERID_APPARENCE</b> = APPARENCEPOSSIBLE_METIERID_APPARENCE	FK_GOODAM
→	METIER	<b>APPARENCE_METIERID_METIER</b> = METIERID	FK_METIER_AM

### Unique keys

Columns		Name / Description
	ID_METIER, ID_APPARENCE	SYS_C0040730

### Uses

Name	
	APPARENCE_METIER
→	APPARENCE
→	APPARENCEPOSSIBLE_METIER
→	METIER

## 4.1.9. Table: APPARENCE\_PERSONNAGE

### Columns

		Name	Data type	Description / Attributes
		ID_PERSONNAGE	NUMBER(5, 0)	<b>References:</b> APPARENCEPOSSIBLE_PERSONNAGE, PERSONNAGE
		ID_APPARENCE	NUMBER(5, 0)	<b>References:</b> APPARENCE, APPARENCEPOSSIBLE_PERSONNAGE

### Links to

Table		Join	Title / Name / Description
→	APPARENCE	<b>APPARENCE_PERSONNAGE</b> ID_APPARENCE = APPARENCEID	FK_APPARENCE_AP
→	APPARENCEPOSSIBLE_PERSONNAGE	<b>APPARENCE_PERSONNAGE</b> ID_PERSONNAGE = APPARENCEPOSSIBLE_PERSONNAGEID_PERSONNAGE, <b>APPARENCE_PERSONNAGE</b> ID_APPARENCE = APPARENCEPOSSIBLE_PERSONNAGEID_APPARENCE	FK_GOODAP
→	PERSONNAGE	<b>APPARENCE_PERSONNAGE</b> ID_PERSONNAGE = PERSONNAGEID	FK_PERSONNAGE_AP

### Unique keys

Columns		Name / Description
	ID_PERSONNAGE, ID_APPARENCE	SYS_C0040702

### Uses

		Name
		APPARENCE_PERSONNAGE
	→	APPARENCE
	→	APPARENCEPOSSIBLE_PERSONNAGE
	→	PERSONNAGE

#### 4.1.10. Table: APPARENCE\_QUETE

##### Columns

		Name	Data type	Description / Attributes
		ID_QUETE	NUMBER(5, 0)	<b>References:</b> APPARENCEPOSSIBLE_QUETE, QUETE
		ID_APPARENCE	NUMBER(5, 0)	<b>References:</b> APPARENCE, APPARENCEPOSSIBLE_QUETE

##### Links to

Table		Join	Title / Name / Description
→	APPARENCE	<b>APPARENCE_QUETEID_APPARENCE</b> = APPARENCEID	FK_APPARENCE_AQ
→	APPARENCEPOSSIBLE_QUETE	<b>APPARENCE_QUETEID_QUETE</b> = APPARENCEPOSSIBLE_QUETEID_QUETE, <b>APPARENCE_QUETEID_APPARENCE</b> = APPARENCEPOSSIBLE_QUETEID_APPARENCE	FK_GOODAQ
→	QUETE	<b>APPARENCE_QUETEID_QUETE</b> = QUETEID	FK_QUETE_AQ

##### Unique keys

Columns		Name / Description
	ID_QUETE, ID_APPARENCE	SYS_C0040751

##### Uses

		Name
		APPARENCE_QUETE
	→	APPARENCE
	→	APPARENCEPOSSIBLE_QUETE
	→	QUETE

#### 4.1.11. Table: APPARENCE\_RACE

##### Columns

Name		Data type	Description / Attributes
目	🔑 NOM_RACE	VARCHAR2(255 BYTE)	References: APPARENCEPOSSIBLE_RACE, RACE
目	🔑 ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE, APPARENCEPOSSIBLE_RACE

##### Links to

Table		Join	Title / Name / Description
→	APPARENCE	<b>APPARENCE_RACEID_APPARENCE = APPARENCEID</b>	FK_APPARENCE_AR
→	APPARENCEPOSSIBLE_RACE	<b>APPARENCE_RACENOM_RACE = APPARENCEPOSSIBLE_RACENOM_RACE,</b> <b>APPARENCE_RACEID_APPARENCE = APPARENCEPOSSIBLE_RACEID_APPARENCE</b>	FK_GOODAR
→	RACE	<b>APPARENCE_RACENOM_RACE = RACENOM</b>	FK_RACE_AR

##### Unique keys

Columns		Name / Description
🔑	NOM_RACE, ID_APPARENCE	SYS_C0040709

##### Uses

Name	
grid	APPARENCE_RACE
→	APPARENCE
→	APPARENCEPOSSIBLE_RACE
→	RACE

## 4.1.12. Table: APPARENCE\_SORT

### Columns

		Name	Data type	Description / Attributes
目	键	ID_SORT	NUMBER(5, 0)	References: APPARENCEPOSSIBLE_SORT, SORT
目	键	ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE, APPARENCEPOSSIBLE_SORT

### Links to

Table		Join	Title / Name / Description
→	APPARENCE	<b>APPARENCE_SORTID_APPARENCE = APPARENCEID</b>	FK_APPARENCE_AS
→	APPARENCEPOSSIBLE_SORT	<b>APPARENCE_SORTID_SORT = APPARENCEPOSSIBLE_SORTID_SORT</b> , <b>APPARENCE_SORTID_APPARENCE = APPARENCEPOSSIBLE_SORTID_APPARENCE</b>	FK_GOODAS
→	SORT	<b>APPARENCE_SORTID_SORT = SORTID</b>	FK_SORT_AS

### Unique keys

Columns		Name / Description
键	ID_SORT, ID_APPARENCE	SYS_C0040723

### Uses

		Name
表	APPARENCE_SORT	
→	APPARENCE	
→	APPARENCEPOSSIBLE_SORT	
→	SORT	

#### 4.1.13. Table: APPARENCE\_TITRE

##### Columns

		Name	Data type	Description / Attributes
目	🔑	ID_TITRE	NUMBER(5, 0)	References: APPARENCEPOSSIBLE_TITRE, TITRE
目	🔑	ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE, APPARENCEPOSSIBLE_TITRE

##### Links to

Table		Join	Title / Name / Description
→	APPARENCE	<b>APPARENCE_TITREID_APPARENCE</b> = APPARENCEID	FK_APPARENCE_AT
→	APPARENCEPOSSIBLE_TITRE	<b>APPARENCE_TITREID_TITRE</b> = APPARENCEPOSSIBLE_TITREID_TITRE, <b>APPARENCE_TITREID_APPARENCE</b> = APPARENCEPOSSIBLE_TITREID_APPARENCE	FK_GOODAT
→	TITRE	<b>APPARENCE_TITREID_TITRE</b> = TITREID	FK_TITRE_AT

##### Unique keys

Columns		Name / Description
🔑	ID_TITRE, ID_APPARENCE	SYS_C0040695

##### Uses

		Name
█	APPARENCE_TITRE	
→	APPARENCE	
→	APPARENCEPOSSIBLE_TITRE	
→	TITRE	

#### 4.1.14. Table: APPARENCEPOSSIBLE\_CLASSE

##### Columns

Name		Data type	Description / Attributes
目	🔑 NOM_CLASSE	VARCHAR2(255 BYTE)	References: CLASSE
目	🔑 ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE

##### Links to

Table		Join	Title / Name / Description
→	APPARENCE	APPARENCEPOSSIBLE_CLASSEID_APPARENCE = APPARENCEID	FK_APPARENCE_APP
→	CLASSE	APPARENCEPOSSIBLE_CLASSENOM_CLASSE = CLASSENOM	FK_CLASSE_APP

##### Linked from

Table		Join	Title / Name / Description
←	APPARENCE_CLASSE	APPARENCEPOSSIBLE_CLASSENOM_CLASSE = APPARENCE_CLASSENOM_CLASSE, APPARENCEPOSSIBLE_CLASSEID_APPARENCE = APPARENCE_CLASSEID_APPARENCE	FK_GOODAC

##### Unique keys

Columns		Name / Description
🔑	NOM_CLASSE, ID_APPARENCE	SYS_C0040713

##### Uses

Name	
█	APPARENCEPOSSIBLE_CLASSE
→	APPARENCE
→	CLASSE

##### Used By

Name	
█	APPARENCEPOSSIBLE_CLASSE
←	APPARENCE_CLASSE

## 4.1.15. Table: APPARENCEPOSSIBLE\_EVENEMENT

### Columns

		Name	Data type	Description / Attributes
目	🔑	ID_EVENEMENT	NUMBER(5, 0)	References: EVENEMENT
目	🔑	ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE

### Links to

Table		Join	Title / Name / Description
→	APPARENCE	APPARENCEPOSSIBLE_EVENEMENTID_APPARENCE = APPARENCEID	FK_APPARENCE_APE
→	EVENEMENT	APPARENCEPOSSIBLE_EVENEMENTID_EVENEMENT = EVENEMENTID	FK_EVENEMENT_APE

### Linked from

Table		Join	Title / Name / Description
←	APPARENCE_EVENEMENT	APPARENCEPOSSIBLE_EVENEMENTID_EVENEMENT = APPARENCE_EVENEMENTID_EVENEMENT, APPARENCEPOSSIBLE_EVENEMENTID_APPARENCE = APPARENCE_EVENEMENTID_APPARENCE	FK_GOODAE

### Unique keys

Columns		Name / Description
🔑	ID_EVENEMENT, ID_APPARENCE	SYS_C0040741

### Uses

		Name
█	APPARENCEPOSSIBLE_EVENEMENT	
→	APPARENCE	
→	EVENEMENT	

### Used By

		Name
█	APPARENCEPOSSIBLE_EVENEMENT	
←	APPARENCE_EVENEMENT	

#### 4.1.16. Table: APPARENCEPOSSIBLE\_FACTION

##### Columns

		Name	Data type	Description / Attributes
		ID_FACTION	NUMBER(5, 0)	References: FACTION
		ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE

##### Links to

Table		Join	Title / Name / Description
→	APPARENCE	APPARENCEPOSSIBLE_FACTIONID_APPARENCE = APPARENCEID	FK_APPARENCE_APF
→	FACTION	APPARENCEPOSSIBLE_FACTIONID_FACTION = FACTIONID	FK_FACTION_APF

##### Linked from

Table		Join	Title / Name / Description
←	APPARENCE_FACTION	APPARENCEPOSSIBLE_FACTIONID_FACTION = APPARENCE_FACTIONID_FACTION, APPARENCEPOSSIBLE_FACTIONID_APPARENCE = APPARENCE_FACTIONID_APPARENCE	FK_GOOD_AF

##### Unique keys

Columns		Name / Description
	ID_FACTION, ID_APPARENCE	SYS_C0040678

##### Uses

		Name
		APPARENCEPOSSIBLE_FACTION
→	APPARENCE	
→	FACTION	

##### Used By

		Name
		APPARENCEPOSSIBLE_FACTION
←	APPARENCE_FACTION	

## 4.1.17. Table: APPARENCEPOSSIBLE\_HAUTFAIT

### Columns

		Name	Data type	Description / Attributes
		ID_HAUTFAIT	NUMBER(5, 0)	References: HAUTFAIT
		ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE

### Links to

Table		Join	Title / Name / Description
→	APPARENCE	APPARENCEPOSSIBLE_HAUTFAITID_APPARENCE = APPARENCEID	FK_APPARENCE_APH
→	HAUTFAIT	APPARENCEPOSSIBLE_HAUTFAITID_HAUTFAIT = HAUTFAITID	FK_HAUTFAIT_APH

### Linked from

Table		Join	Title / Name / Description
←	APPARENCE_HAUTFAIT	APPARENCEPOSSIBLE_HAUTFAITID_HAUTFAIT = APPARENCE_HAUTFAITID_HAUTFAIT, APPARENCEPOSSIBLE_HAUTFAITID_APPARENCE = APPARENCE_HAUTFAITID_APPARENCE	FK_GOOD_AH

### Unique keys

Columns		Name / Description
	ID_HAUTFAIT, ID_APPARENCE	SYS_C0040685

### Uses

Name	
	APPARENCEPOSSIBLE_HAUTFAIT
→	APPARENCE
→	HAUTFAIT

### Used By

Name	
	APPARENCEPOSSIBLE_HAUTFAIT
←	APPARENCE_HAUTFAIT

## 4.1.18. Table: APPARENCEPOSSIBLE\_ITEM

### Columns

		Name	Data type	Description / Attributes
█	█	ID_ITEM	NUMBER(5, 0)	References: ITEM
█	█	ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE

### Links to

Table		Join	Title / Name / Description
→	APPARENCE	APPARENCEPOSSIBLE_ITEMID_APPARENCE = APPARENCEID	FK_APPARENCE_API
→	ITEM	APPARENCEPOSSIBLE_ITEMID_ITEM = ITEMID	FK_ITEM_API

### Linked from

Table		Join	Title / Name / Description
←	APPARENCE_ITEM	APPARENCEPOSSIBLE_ITEMID_ITEM = APPARENCE_ITEMID_ITEM, APPARENCEPOSSIBLE_ITEMID_APPARENCE = APPARENCE_ITEMID_APPARENCE	FK_GOODAI

### Unique keys

Columns		Name / Description
█	ID_ITEM, ID_APPARENCE	SYS_C0040734

### Uses

		Name
█	APPARENCEPOSSIBLE_ITEM	
→	APPARENCE	
→	ITEM	

### Used By

		Name
█	APPARENCEPOSSIBLE_ITEM	
←	APPARENCE_ITEM	

#### 4.1.19. Table: APPARENCEPOSSIBLE\_LIEUX

##### Columns

		Name	Data type	Description / Attributes
目	键	ID_LIEUX	NUMBER(5, 0)	References: LIEUX
目	键	ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE

##### Links to

Table		Join	Title / Name / Description
→	APPARENCE	APPARENCEPOSSIBLE_LIEUXID_APPARENCE = APPARENCEID	FK_APPARENCE_APL
→	LIEUX	APPARENCEPOSSIBLE_LIEUXID_LIEUX = LIEUXID	FK_LIEUX_APL

##### Linked from

Table		Join	Title / Name / Description
←	APPARENCE_LIEUX	APPARENCEPOSSIBLE_LIEUXID_LIEUX = APPARENCE_LIEUXID_LIEUX, APPARENCEPOSSIBLE_LIEUXID_APPARENCE = APPARENCE_LIEUXID_APPARENCE	FK_GOODAL

##### Unique keys

Columns		Name / Description
键	ID_LIEUX, ID_APPARENCE	SYS_C0040755

##### Uses

		Name
表格	APPARENCEPOSSIBLE_LIEUX	
→	APPARENCE	
→	LIEUX	

##### Used By

		Name
表格	APPARENCEPOSSIBLE_LIEUX	
←	APPARENCE_LIEUX	

## 4.1.20. Table: APPARENCEPOSSIBLE\_METIER

### Columns

		Name	Data type	Description / Attributes
目	🔑	ID_METIER	NUMBER(5, 0)	References: METIER
目	🔑	ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE

### Links to

Table		Join	Title / Name / Description
→	APPARENCE	APPARENCEPOSSIBLE_METIERID_APP ARENCE = APPARENCEID	FK_APPARENCE_APM
→	METIER	APPARENCEPOSSIBLE_METIERID_MET IER = METIERID	FK_METIER_APM

### Linked from

Table		Join	Title / Name / Description
←	APPARENCE_METIER	APPARENCEPOSSIBLE_METIERID_MET IER = APPARENCE_METIERID_METIER, APPARENCEPOSSIBLE_METIERID_APP ARENCE = APPARENCE_METIERID_APPARENCE	FK_GOODAM

### Unique keys

Columns		Name / Description
🔑	ID_METIER, ID_APPARENCE	SYS_C0040727

### Uses

		Name
█	APPARENCEPOSSIBLE_METIER	
→	APPARENCE	
→	METIER	

### Used By

		Name
█	APPARENCEPOSSIBLE_METIER	
←	APPARENCE_METIER	

## 4.1.21. Table: APPARENCEPOSSIBLE\_PERSONNAGE

### Columns

		Name	Data type	Description / Attributes
		ID_PERSONNAGE	NUMBER(5, 0)	References: PERSONNAGE
		ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE

### Links to

Table		Join	Title / Name / Description
→	APPARENCE	APPARENCEPOSSIBLE_PERSONNAGE  D_APPARENCE = APPARENCEID	FK_APPARENCE_APP
→	PERSONNAGE	APPARENCEPOSSIBLE_PERSONNAGE  D_PERSONNAGE = PERSONNAGEID	FK_PERSONNAGE_APP

### Linked from

Table		Join	Title / Name / Description
←	APPARENCE_PERSONNAGE	APPARENCEPOSSIBLE_PERSONNAGE  D_PERSONNAGE = APPARENCE_PERSONNAGEID_PERSONNAGE, APPARENCEPOSSIBLE_PERSONNAGE  D_APPARENCE = APPARENCE_PERSONNAGEID_APPARENCE	FK_GOODAP

### Unique keys

Columns		Name / Description
	ID_PERSONNAGE, ID_APPARENCE	SYS_C0040699

### Uses

Name	
	APPARENCEPOSSIBLE_PERSONNAGE
→	APPARENCE
→	PERSONNAGE

### Used By

Name	
	APPARENCEPOSSIBLE_PERSONNAGE
←	APPARENCE_PERSONNAGE

## 4.1.22. Table: APPARENCEPOSSIBLE\_QUETE

### Columns

		Name	Data type	Description / Attributes
目	键	ID_QUETE	NUMBER(5, 0)	References: QUETE
目	键	ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE

### Links to

Table		Join	Title / Name / Description
→	APPARENCE	APPARENCEPOSSIBLE_QUETEID_APP ARENCE = APPARENCEID	FK_APPARENCE_APQ
→	QUETE	APPARENCEPOSSIBLE_QUETEID_QUE TE = QUETEID	FK_QUETE_APQ

### Linked from

Table		Join	Title / Name / Description
←	APPARENCE_QUETE	APPARENCEPOSSIBLE_QUETEID_QUE TE = APPARENCE_QUETEID_QUETE, APPARENCEPOSSIBLE_QUETEID_APP ARENCE = APPARENCE_QUETEID_APPARENCE	FK_GOODAQ

### Unique keys

Columns		Name / Description
键	ID_QUETE, ID_APPARENCE	SYS_C0040748

### Uses

		Name
表格	APPARENCEPOSSIBLE_QUETE	
→	APPARENCE	
→	QUETE	

### Used By

		Name
表格	APPARENCEPOSSIBLE_QUETE	
←	APPARENCE_QUETE	

## 4.1.23. Table: APPARENCEPOSSIBLE\_RACE

### Columns

Name		Data type	Description / Attributes
目	🔑 NOM_RACE	VARCHAR2(255 BYTE)	References: RACE
目	🔑 ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE

### Links to

Table		Join	Title / Name / Description
→	APPARENCE	APPARENCEPOSSIBLE_RACEID_APPARENCE = APPARENCEID	FK_APPARENCE_APR
→	RACE	APPARENCEPOSSIBLE_RACENOM_RACE = RACENOM	FK_RACE_APR

### Linked from

Table		Join	Title / Name / Description
←	APPARENCE_RACE	APPARENCEPOSSIBLE_RACENOM_RACE = APPARENCE_RACENOM_RACE, APPARENCEPOSSIBLE_RACEID_APPARENCE = APPARENCE_RACEID_APPARENCE	FK_GOODAR

### Unique keys

Columns		Name / Description
🔑	NOM_RACE, ID_APPARENCE	SYS_C0040706

### Uses

Name	
█ APPARENCEPOSSIBLE_RACE	
→ APPARENCE	
→ RACE	

### Used By

Name	
█ APPARENCEPOSSIBLE_RACE	
← APPARENCE_RACE	

#### 4.1.24. Table: APPARENCEPOSSIBLE\_SORT

##### Columns

		Name	Data type	Description / Attributes
█	█	ID_SORT	NUMBER(5, 0)	References: SORT
█	█	ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE

##### Links to

Table		Join	Title / Name / Description
→	APPARENCE	APPARENCEPOSSIBLE_SORTID_APPARENCE = APPARENCEID	FK_APPARENCEAPS
→	SORT	APPARENCEPOSSIBLE_SORTID_SORT = SORTID	FK_SORTAPS

##### Linked from

Table		Join	Title / Name / Description
←	APPARENCE_SORT	APPARENCEPOSSIBLE_SORTID_SORT = APPARENCE_SORTID_SORT, APPARENCEPOSSIBLE_SORTID_APPARENCE = APPARENCE_SORTID_APPARENCE	FK_GOODAS

##### Unique keys

Columns		Name / Description
█	ID_SORT, ID_APPARENCE	SYS_C0040720

##### Uses

		Name
█	APPARENCEPOSSIBLE_SORT	
→	APPARENCE	
→	SORT	

##### Used By

		Name
█	APPARENCEPOSSIBLE_SORT	
←	APPARENCE_SORT	

## 4.1.25. Table: APPARENCEPOSSIBLE\_TITRE

### Columns

		Name	Data type	Description / Attributes
		ID_TITRE	NUMBER(5, 0)	References: TITRE
		ID_APPARENCE	NUMBER(5, 0)	References: APPARENCE

### Links to

		Table	Join	Title / Name / Description
→	APPARENCE	APPARENCEPOSSIBLE_TITREID_APPARENCE = APPARENCEID	FK_APPARENCE_APT	
→	TITRE	APPARENCEPOSSIBLE_TITREID_TITRE = TITREID	FK_TITRE_APT	

### Linked from

		Table	Join	Title / Name / Description
←	APPARENCE_TITRE	APPARENCEPOSSIBLE_TITREID_TITRE = APPARENCE_TITREID_TITRE, APPARENCEPOSSIBLE_TITREID_APPARENCE = APPARENCE_TITREID_APPARENCE	FK_GOODAT	

### Unique keys

		Columns	Name / Description
		ID_TITRE, ID_APPARENCE	SYS_C0040692

### Uses

		Name
		APPARENCEPOSSIBLE_TITRE
→	APPARENCE	
→	TITRE	

### Used By

		Name
		APPARENCEPOSSIBLE_TITRE
←	APPARENCE_TITRE	

## 4.1.26. Table: CALENDRIER\_EVENEMENT

### Columns

		Name	Data type	Description / Attributes
		DATEEVENT	DATE	<b>Default:</b> SYSDATE
		ID	NUMBER(5, 0)	<b>References:</b> EVENEMENT

### Links to

Table		Join	Title / Name / Description
	EVENEMENT	CALENDRIER_EVENEMENTID = EVENEMENTID	FK_EVENEMENT_CE

### Unique keys

Columns		Name / Description
	DATEEVENT, ID	SYS_C0040655

### Uses

		Name
	CALENDRIER_EVENEMENT	
	EVENEMENT	

## 4.1.27. Table: EVENEMENT

### Columns

		Name	Data type	Description / Attributes
■	🔑	ID	NUMBER	
■		NOM	VARCHAR2(255 BYTE)	
■		DESCRIPTION	CLOB	<b>Nullable</b>
■		DUREE	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 1

### Linked from

Table		Join	Title / Name / Description
→	APPARENCE_EVENEMENT	<b>EVENEMENTID</b> = APPARENCE_EVENEMENTID_EVENEMENT	FK_EVENEMENT_AE
→	APPARENCEPOSSIBLE_EVENEMENT	<b>EVENEMENTID</b> = APPARENCEPOSSIBLE_EVENEMENTID_EVENEMENT	FK_EVENEMENT_APE
→	CALENDRIER_EVENEMENT	<b>EVENEMENTID</b> = CALENDRIER_EVENEMENTID	FK_EVENEMENT_CE
→	EVENEMENT_ITEM	<b>EVENEMENTID</b> = EVENEMENT_ITEMID_EVENEMENT	FK_EVENEMENT_EVENEMENT_ITEM
→	EVENEMENT_MONSTRE	<b>EVENEMENTID</b> = EVENEMENT_MONSTREID_EVENEMENT	FK_EVENEMENT_EVENEMENT_MONSTRE
→	EVENEMENT_QUETE	<b>EVENEMENTID</b> = EVENEMENT_QUETEID_EVENEMENT	FK_EVENEMENT_EVENEMENT_QUETE

### Unique keys

Columns		Name / Description
🔑	ID	SYS_C0040554

### Triggers

Name		When	Description
⚡	NEW_EVENEMENT_TRIGGER	Before Insert	
<pre>TRIGGER New_Evenement_trigger BEFORE INSERT ON EVENEMENT FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := EVENEMENT_seq.NEXTVAL; END;</pre>			

### Uses

Name	
⚡	NEW_EVENEMENT_TRIGGER
⌚	EVENEMENT

## Used By

Name
EVENEMENT
⚡ NEW_EVENEMENT_TRIGGER
→ APPARENCE_EVENEMENT
→ APPARENCEPOSSIBLE_EVENEMENT
→ CALENDRIER_EVENEMENT
→ EVENEMENT_ITEM
→ EVENEMENT_MONSTRE
→ EVENEMENT_QUETE

## 4.1.28. Table: EVENEMENT\_MONSTRE

### Columns

		Name	Data type	Description / Attributes
		ID_MONSTRE	NUMBER(5, 0)	<b>References:</b> MONSTRE
		ID_EVENEMENT	NUMBER(5, 0)	<b>References:</b> EVENEMENT

### Links to

Table		Join	Title / Name / Description
→	EVENEMENT	<b>EVENEMENT_MONSTREID_EVENEMENT</b> = EVENEMENTID	FK_EVENEMENT_EVENEMENT_MONSTRE
→	MONSTRE	<b>EVENEMENT_MONSTREID_MONSTRE</b> = MONSTREID	FK_MONSTRE_EVENEMENT_MONSTRE

### Unique keys

Columns		Name / Description
	ID_MONSTRE, ID_EVENEMENT	SYS_C0040649

### Uses

		Name
		EVENEMENT_MONSTRE
→	EVENEMENT	
→	MONSTRE	

#### 4.1.29. Table: EVENEMENT\_QUETE

##### Columns

		Name	Data type	Description / Attributes
目	🔑	ID_QUETE	NUMBER(5, 0)	References: QUETE
目	🔑	ID_EVENEMENT	NUMBER(5, 0)	References: EVENEMENT

##### Links to

Table		Join	Title / Name / Description
→	EVENEMENT	EVENEMENT_QUETEID_EVENEMENT = EVENEMENTID	FK_EVENEMENT_EVENEMENT_QUETE
→	QUETE	EVENEMENT_QUETEID_QUETE = QUETEID	FK_QUETE_EVENEMENT_QUETE

##### Unique keys

Columns		Name / Description
🔑	ID_QUETE, ID_EVENEMENT	SYS_C0040660

##### Uses

		Name
grid	EVENEMENT_QUETE	
→	EVENEMENT	
→	QUETE	

## 4.1.30. Table: FACTION

### Columns

		Name	Data type	Description / Attributes
█	█	ID	NUMBER	
█		NOM	VARCHAR2(255 BYTE)	
█		DESCRIPTION	CLOB	Nullable

### Linked from

Table		Join	Title / Name / Description
→ APPARENCE_FACTION		FACTIONID = APPARENCE_FACTIONID_FACTION	FK_FACTION_AF
→ APPARENCEPOSSIBLE_FACTION		FACTIONID = APPARENCEPOSSIBLE_FACTIONID_FACTION	FK_FACTION_APF
→ FACTION_PERSONNAGE		FACTIONID = FACTION_PERSONNAGEID_FACTION	FK_FACTION_FP

### Unique keys

Columns		Name / Description
█	ID	SYS_C0040506

### Triggers

Name		When	Description
⚡	NEW_FACTION_TRIGGER	Before Insert	
<pre>TRIGGER New_Faction_trigger BEFORE INSERT ON FACTION FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := FACTION_seq.NEXTVAL; END;</pre>			

### Uses

Name	
⚡	NEW_FACTION_TRIGGER
█	FACTION

### Used By

Name	
█	FACTION
⚡	NEW_FACTION_TRIGGER
→	APPARENCE_FACTION
→	APPARENCEPOSSIBLE_FACTION
→	FACTION_PERSONNAGE

#### 4.1.31. Table: FACTION\_PERSONNAGE

##### Columns

		Name	Data type	Description / Attributes
■	🔑	ID_PERSONNAGE	NUMBER(5, 0)	<b>References:</b> PERSONNAGE
■	🔑	ID_FACTION	NUMBER(5, 0)	<b>References:</b> FACTION
■		GRADE	VARCHAR2(255 BYTE)	<b>Nullable</b> <b>Default:</b> 'Étranger'
■		REPUTATION	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 0

##### Links to

Table		Join	Title / Name / Description
→	FACTION	FACTION_PERSONNAGEID_FACTION = FACTIONID	FK_FACTION_FP
→	PERSONNAGE	FACTION_PERSONNAGEID_PERSONNAGE = PERSONNAGEID	FK_PERSONNAGE_FP

##### Unique keys

Columns		Name / Description
🔑	ID_PERSONNAGE, ID_FACTION	SYS_C0040589

##### Uses

		Name
█	FACTION_PERSONNAGE	
→	FACTION	
→	PERSONNAGE	

## 4.1.32. Table: HAUTFAIT

### Columns

		Name	Data type	Description / Attributes
■	🔑	ID	NUMBER	
■		NOM	VARCHAR2(255 BYTE)	
■		DESCRIPTION	CLOB	Nullable
■		POINTHF	NUMBER(5, 0)	Nullable Default: 0

### Linked from

Table		Join	Title / Name / Description
→	APPARENCE_HAUTFAIT	HAUTFAITID = APPARENCE_HAUTFAITID_HAUTFAIT	FK_HAUTFAIT_AH
→	APPARENCEPOSSIBLE_HAUTFAIT	HAUTFAITID = APPARENCEPOSSIBLE_HAUTFAITID_HAUTFAIT	FK_HAUTFAITAPH
→	HAUTFAIT_PERSONNAGE	HAUTFAITID = HAUTFAIT_PERSONNAGEID_HAUTFAIT	FK_HAUTFAIT_HP

### Unique keys

Columns		Name / Description
🔑	ID	SYS_C0040508

### Triggers

Name		When	Description
⚡	NEW_HAUTFAIT_TRIGGER	Before Insert	
<pre>TRIGGER New_HautFait_trigger BEFORE INSERT ON HAUTFAIT FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := HAUTFAIT_seq.NEXTVAL; END;</pre>			

### Uses

		Name
⚡	NEW_HAUTFAIT_TRIGGER	
█	HAUTFAIT	

### Used By

		Name
█	HAUTFAIT	
⚡	NEW_HAUTFAIT_TRIGGER	
→	APPARENCE_HAUTFAIT	
→	APPARENCEPOSSIBLE_HAUTFAIT	
→	HAUTFAIT_PERSONNAGE	

#### 4.1.33. Table: HAUTFAIT\_PERSONNAGE

##### Columns

		Name	Data type	Description / Attributes
目	🔑	ID_PERSONNAGE	NUMBER(5, 0)	References: PERSONNAGE
目	🔑	ID_HAUTFAIT	NUMBER(5, 0)	References: HAUTFAIT

##### Links to

Table		Join	Title / Name / Description
→	HAUTFAIT	HAUTFAIT_PERSONNAGEID_HAUTFAIT = HAUTFAITID	FK_HAUTFAIT_HP
→	PERSONNAGE	HAUTFAIT_PERSONNAGEID_PERSONNAGE = PERSONNAGEID	FK_PERSONNAGE_HP

##### Unique keys

Columns		Name / Description
🔑	ID_PERSONNAGE, ID_HAUTFAIT	SYS_C0040592

##### Uses

		Name
grid	HAUTFAIT_PERSONNAGE	
→	HAUTFAIT	
→	PERSONNAGE	

#### 4.1.34. Table: LANGUE

##### Columns

		Name	Data type	Description / Attributes
		ID	NUMBER	
		NOM	VARCHAR2(255 BYTE)	
		DESCRIPTION	CLOB	Nullable

##### Linked from

Table		Join	Title / Name / Description
	LANGUE_PERSONNAGE	LANGUEID = LANGUE_PERSONNAGEID_LANGUE	FK_LANGUE_LP
	LANGUE_RACE	LANGUEID = LANGUE_RACEID_LANGUE	FK_LANGUE_LR

##### Unique keys

Columns		Name / Description
	ID	SYS_C0040512

##### Triggers

Name		When	Description
	NEW_LANGUE_TRIGGER	Before Insert	
<pre>TRIGGER New_Langue_trigger BEFORE INSERT ON LANGUE FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := LANGUE_seq.NEXTVAL; END;</pre>			

##### Uses

Name	
	NEW_LANGUE_TRIGGER
	LANGUE

##### Used By

Name	
	LANGUE
	NEW_LANGUE_TRIGGER
	LANGUE_PERSONNAGE
	LANGUE_RACE

#### 4.1.35. Table: LANGUE\_PERSONNAGE

##### Columns

		Name	Data type	Description / Attributes
目	🔑	ID_PERSONNAGE	NUMBER(5, 0)	References: PERSONNAGE
目	🔑	ID_LANGUE	NUMBER(5, 0)	References: LANGUE

##### Links to

Table		Join	Title / Name / Description
→	LANGUE	LANGUE_PERSONNAGEID_LANGUE = LANGUEID	FK_LANGUE_LP
→	PERSONNAGE	LANGUE_PERSONNAGEID_PERSONNAGE = PERSONNAGEID	FK_PERSONNAGE_LP

##### Unique keys

Columns		Name / Description
🔑	ID_PERSONNAGE, ID_LANGUE	SYS_C0040598

##### Uses

		Name
grid	LANGUE_PERSONNAGE	
→	LANGUE	
→	PERSONNAGE	

#### 4.1.36. Table: LANGUE\_RACE

##### Columns

		Name	Data type	Description / Attributes
目	🔑	NOM_RACE	VARCHAR2(255 BYTE)	References: RACE
目	🔑	ID_LANGUE	NUMBER(5, 0)	References: LANGUE

##### Links to

Table		Join	Title / Name / Description
→	LANGUE	LANGUE_RACEID_LANGUE = LANGUEID	FK_LANGUE_LR
→	RACE	LANGUE_RACENOM_RACE = RACENOM	FK_RACE_LR

##### Unique keys

Columns		Name / Description
🔑	NOM_RACE, ID_LANGUE	SYS_C0040601

##### Uses

		Name
grid	LANGUE_RACE	
→	LANGUE	
→	RACE	

## 4.1.37. Table: LIEUX

### Columns

		Name	Data type	Description / Attributes
█	█	ID	NUMBER	
█		NOM	VARCHAR2(255 BYTE)	
█		DESCRIPTION	CLOB	<b>Nullable</b>
█		ID_TYPELIEUX	NUMBER(5, 0)	<b>Nullable</b> <b>References:</b> TYPELIEUX

### Links to

Table		Join	Title / Name / Description
→	TYPELIEUX	LIEUXID_TYPELIEUX = TYPELIEUXID	FK_TYPELIEUX_LIEUX

### Linked from

Table		Join	Title / Name / Description
←	APPARENCE_LIEUX	LIEUXID = APPARENCE_LIEUXID_LIEUX	FK_LIEUX_AL
←	APPARENCEPOSSIBLE_LIEUX	LIEUXID = APPARENCEPOSSIBLE_LIEUXID_LIEUX	FK_LIEUX_APL
←	PERSONNAGE	LIEUXID = PERSONNAGEID_LIEUX	FK_LIEUX_PERSONNAGE

### Unique keys

Columns		Name / Description
█	ID	SYS_C0040547

### Triggers

Name		When	Description
⚡	NEW_LIEUX_TRIGGER	Before Insert	
<pre>TRIGGER New_Lieux_trigger BEFORE INSERT ON LIEUX FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := LIEUX_seq.NEXTVAL; END;</pre>			

### Uses

		Name
█	LIEUX	
→	TYPELIEUX	
⚡	NEW_LIEUX_TRIGGER	
▀	LIEUX	

### Used By

		Name
█	LIEUX	

Name

⚡ NEW\_LIEUX\_TRIGGER

→ APPARENCE\_LIEUX

→ APPARENCEPOSSIBLE\_LIEUX

→ PERSONNAGE

## 4.1.38. Table: METIER

### Columns

		Name	Data type	Description / Attributes
		ID	NUMBER	
		NOM	VARCHAR2(255 BYTE)	
		DESCRIPTION	CLOB	Nullable

### Linked from

Table		Join	Title / Name / Description
	APPARENCE_METIER	<b>METIERID = APPARENCE_METIERID_METIER</b>	FK_METIER_AM
	APPARENCEPOSSIBLE_METIER	<b>METIERID = APPARENCEPOSSIBLE_METIERID_METIER</b>	FK_METIER_APM
	METIER_OUTIL	<b>METIERID = METIER_OUTILID_METIER</b>	FK_METIER_MO
	METIER_PERSONNAGE	<b>METIERID = METIER_PERSONNAGEID_METIER</b>	FK_METIER_MP
	METIER_RECETTE	<b>METIERID = METIER_RECETTEID_METIER</b>	FK_METIER_MR
	METIER_TYPERESSOURCE	<b>METIERID = METIER_TYPERESSOURCEID_METIER</b>	FK_METIER_METIER_RESSOURCE

### Unique keys

Columns		Name / Description
	ID	SYS_C0040540

### Triggers

Name When Description		
	NEW_METIER_TRIGGER	Before Insert
<pre>TRIGGER New_Metier_trigger BEFORE INSERT ON METIER FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := METIER_seq.NEXTVAL; END;</pre>		

### Uses

Name	
	NEW_METIER_TRIGGER
	METIER

### Used By

Name	
	METIER
	NEW_METIER_TRIGGER
	APPARENCE_METIER

Name
→ APPARENCEPOSSIBLE_METIER
→ METIER_OUTIL
→ METIER_PERSONNAGE
→ METIER_RECETTE
→ METIER_TYPERESSOURCE

#### 4.1.39. Table: METIER\_PERSONNAGE

##### Columns

		Name	Data type	Description / Attributes
■	🔑	ID_PERSONNAGE	NUMBER(5, 0)	References: PERSONNAGE
■	🔑	ID_METIER	NUMBER(5, 0)	References: METIER
■		NIVEAU	NUMBER(5, 0)	Nullable Default: 1

##### Links to

Table		Join	Title / Name / Description
→	METIER	METIER_PERSONNAGEID_METIER = METIERID	FK_METIER_MP
→	PERSONNAGE	METIER_PERSONNAGEID_PERSONNAGE = PERSONNAGEID	FK_PERSONNAGE_MP

##### Unique keys

Columns		Name / Description
🔑	ID_PERSONNAGE, ID_METIER	SYS_C0040637

##### Uses

Name	
█	METIER_PERSONNAGE
→	METIER
→	PERSONNAGE

#### 4.1.40. Table: METIER\_RECETTE

##### Columns

		Name	Data type	Description / Attributes
		ID_METIER	NUMBER(5, 0)	<b>References:</b> METIER
		ID_RECETTE	NUMBER(5, 0)	<b>References:</b> RECETTE
		NIVEAUREQUIS	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 1

##### Links to

Table		Join	Title / Name / Description
	METIER	<b>METIER_RECETTE</b> ID_METIER = METIERID	FK_METIER_MR
	RECETTE	<b>METIER_RECETTE</b> ID_RECETTE = RECETTEID	FK_RECETTE_MR

##### Unique keys

Columns		Name / Description
	ID_METIER, ID_RECETTE	SYS_C0040613

##### Uses

Name	
	METIER_RECETTE
	METIER
	RECETTE

#### 4.1.41. Table: METIER\_TYPERESSOURCE

##### Columns

		Name	Data type	Description / Attributes
目	键	ID_TYPERESSOURCE	NUMBER(5, 0)	References: TYPERESSOURCE
目	键	ID_METIER	NUMBER(5, 0)	References: METIER

##### Links to

Table		Join	Title / Name / Description
→	METIER	METIER_TYPERESSOURCEID_METIER = METIERID	FK_METIER_METIER_RESSOURCE
→	TYPERESSOURCE	METIER_TYPERESSOURCEID_TYPERESOURCE = TYPERESSOURCEID	FK_RESSOURCE_METIER_RESSOURCE

##### Unique keys

Columns		Name / Description
键	ID_TYPERESSOURCE, ID_METIER	SYS_C0040672

##### Uses

		Name
表格	METIER_TYPERESSOURCE	
→	METIER	
→	TYPERESSOURCE	

#### 4.1.42. Table: MONSTRE\_TYPEMONSTRE

##### Columns

		Name	Data type	Description / Attributes
		ID_MONSTRE	NUMBER(5, 0)	<b>References:</b> MONSTRE
		ID_TYPEMONSTRE	NUMBER(5, 0)	<b>References:</b> TYPEMONSTRE

##### Links to

Table		Join	Title / Name / Description
→	MONSTRE	<b>MONSTRE_TYPEMONSTRE</b> ID_MONSTRE = MONSTREID	FK_MONSTRE_MT
→	TYPEMONSTRE	<b>MONSTRE_TYPEMONSTRE</b> ID_TYPEMONSTRE = TYPEMONSTREID	FK_TYPEMONSTRE_MT

##### Unique keys

Columns		Name / Description
	ID_MONSTRE, ID_TYPEMONSTRE	SYS_C0040604

##### Uses

		Name
		MONSTRE_TYPEMONSTRE
→		MONSTRE
→		TYPEMONSTRE

#### 4.1.43. Table: PERSONNAGE\_QUETE

##### Columns

		Name	Data type	Description / Attributes
■	🔑	ID_PERSONNAGE	NUMBER(5, 0)	<b>References:</b> PERSONNAGE
■	🔑	ID_QUETE	NUMBER(5, 0)	<b>References:</b> QUETE
■		STATUT	VARCHAR2(255 BYTE)	<b>Nullable</b> <b>Default:</b> 'En Cours'
■		DATEDEBUT	DATE	<b>Nullable</b> <b>Default:</b> SYSDATE
■		DATEFIN	DATE	<b>Nullable</b>

##### Links to

Table		Join	Title / Name / Description
→	PERSONNAGE	<b>PERSONNAGE_QUETE</b> ID_PERSONNAGE = PERSONNAGEID	FK_PERSONNAGE_PERSONNAGE_QUETE
→	QUETE	<b>PERSONNAGE_QUETE</b> ID_QUETE = QUETEID	FK_QUETE_PERSONNAGE_QUETE

##### Unique keys

Columns		Name / Description
🔑	ID_PERSONNAGE, ID_QUETE	SYS_C0040666

##### Uses

		Name
grid	PERSONNAGE_QUETE	
→	PERSONNAGE	
→	QUETE	

#### 4.1.44. Table: PERSONNAGE\_QUETEJOURNALIERE

##### Columns

		Name	Data type	Description / Attributes
■	🔑	ID_PERSONNAGE	NUMBER(5, 0)	<b>References:</b> PERSONNAGE
■	🔑	ID_QUETE	NUMBER(5, 0)	<b>References:</b> QUETE
■	🔑	CODE_JOUR	DATE	<b>Default:</b> SYSDATE
■		STATUT	VARCHAR2(255 BYTE)	<b>Nullable</b> <b>Default:</b> 'En Cours'
■		DATEDEBUT	DATE	<b>Nullable</b> <b>Default:</b> SYSDATE
■		DATEFIN	DATE	<b>Nullable</b>

##### Links to

Table		Join	Title / Name / Description
→	PERSONNAGE	PERSONNAGE_QUETEJOURNALIERE D_PERSONNAGE = PERSONNAGEID	FK_PERSONNAGE_PQJ
→	QUETE	PERSONNAGE_QUETEJOURNALIERE D_Quete = QUETEID	FK_QUETEJOURNALIERE_PQJ

##### Unique keys

Columns		Name / Description
🔑	ID_PERSONNAGE, ID_Quete, CODE_JOUR	SYS_C0040669

##### Uses

		Name
█	PERSONNAGE_QUETEJOURNALIERE	
→	PERSONNAGE	
→	QUETE	

#### 4.1.45. Table: PERSONNAGE\_RECETTE

##### Columns

		Name	Data type	Description / Attributes
目	🔑	ID_PERSONNAGE	NUMBER(5, 0)	References: PERSONNAGE
目	🔑	ID_RECETTE	NUMBER(5, 0)	References: RECETTE

##### Links to

Table		Join	Title / Name / Description
→	PERSONNAGE	PERSONNAGE_RECETTE ID_PERSONNAGE = PERSONNAGEID	FK_PERSONNAGE_PR
→	RECETTE	PERSONNAGE_RECETTE ID_RECETTE = RECETTEID	FK_RECETTE_PR

##### Unique keys

Columns		Name / Description
🔑	ID_PERSONNAGE, ID_RECETTE	SYS_C0040640

##### Uses

		Name
grid	PERSONNAGE_RECETTE	
→	PERSONNAGE	
→	RECETTE	

#### 4.1.46. Table: PERSONNAGE\_TITRE

##### Columns

		Name	Data type	Description / Attributes
■	🔑	ID_PERSONNAGE	NUMBER(5, 0)	<b>References:</b> PERSONNAGE
■	🔑	ID_TITRE	NUMBER(5, 0)	<b>References:</b> TITRE
■		ESTACTIF	NUMBER(1, 0)	<b>Nullable</b> <b>Default:</b> 0

##### Links to

Table		Join	Title / Name / Description
→	PERSONNAGE	<b>PERSONNAGE_TITRE</b> ID_PERSONNAG E = PERSONNAGEID	FK_PERSONNAGE_PT
→	TITRE	<b>PERSONNAGE_TITRE</b> ID_TITRE = TITREID	FK_TITRE_PT

##### Unique keys

Columns		Name / Description
🔑	ID_PERSONNAGE, ID_TITRE	SYS_C0040595

##### Uses

Name	
grid	PERSONNAGE_TITRE
→	PERSONNAGE
→	TITRE

#### 4.1.47. Table: QUETE

##### Columns

		Name	Data type	Description / Attributes
		ID	NUMBER	
		TITRE	VARCHAR2(255 BYTE)	
		DESCRIPTION	CLOB	<b>Nullable</b>
		NIVEAURECOMMANDÉ	NUMBER(5, 0)	<b>Nullable</b> <b>Default:</b> 1
		STATUT	VARCHAR2(255 BYTE)	<b>Nullable</b> <b>Default:</b> 'En cours'
		DATEDEBUT	DATE	<b>Nullable</b> <b>Default:</b> SYSDATE
		DATEFIN	DATE	<b>Nullable</b>
		ID_TYPEQUETE	NUMBER(5, 0)	<b>Nullable</b> <b>References:</b> TYPEQUETE

##### Links to

Table		Join	Title / Name / Description
→	TYPEQUETE	QUETEID_TYPEQUETE = TYPEQUETEID	FK_TYPEQUETE_QUETE

##### Linked from

Table		Join	Title / Name / Description
←	APPARENCE_QUETE	QUETEID = APPARENCE_QUETEID_QUETE	FK_QUETE_AQ
←	APPARENCEPOSSIBLE_QUETE	QUETEID = APPARENCEPOSSIBLE_QUETEID_QUETE	FK_QUETE_APQ
←	EVENEMENT_QUETE	QUETEID = EVENEMENT_QUETEID_QUETE	FK_QUETE_EVENEMENT_QUETE
←	ITEMQUETE_QUETE	QUETEID = ITEMQUETE_QUETEID_QUETE	FK_QUETE_ITEMQUETE_QUETE
←	PERSONNAGE_QUETE	QUETEID = PERSONNAGE_QUETEID_QUETE	FK_QUETE_PERSONNAGE_QUETE
←	PERSONNAGE_QUETEJOURNALIERE	QUETEID = PERSONNAGE_QUETEJOURNALIEREID_QUETE	FK_QUETEJOURNALIERE_PQJ
←	SUITEQUETE	QUETEID = SUITEQUETEID_FILS	FK_QUETEFILS_SUITEQUETE
←	SUITEQUETE	QUETEID = SUITEQUETEID_PERE	FK_QUETEPERE_SUITEQUETE

##### Unique keys

Columns		Name / Description
		ID SYS_C0040551

## Triggers

	Name	When	Description
⚡	NEW_QUETE_TRIGGER	Before Insert	
<pre>TRIGGER New_Quete_trigger BEFORE INSERT ON QUETE FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := QUETE_seq.NEXTVAL; END;</pre>			

## Uses

	Name
grid	QUETE
→	TYPEQUETE
⚡	NEW_QUETE_TRIGGER
grid	QUETE

## Used By

	Name
grid	QUETE
⚡	NEW_QUETE_TRIGGER
→	APPARENCE_QUETE
→	APPARENCEPOSSIBLE_QUETE
→	EVENEMENT_QUETE
→	ITEMQUETE_QUETE
→	PERSONNAGE_QUETE
→	PERSONNAGE_QUETEJOURNALIERE
→	SUITEQUETE
→	SUITEQUETE

## 4.1.48. Table: RECETTE

### Columns

		Name	Data type	Description / Attributes
█	🔑	ID	NUMBER	
█		NOM	VARCHAR2(255 BYTE)	
█		DESCRIPTION	CLOB	Nullable
█		IDITEM	NUMBER(5, 0)	Nullable References: ITEM

### Links to

Table		Join	Title / Name / Description
→	ITEM	RECETTEIDITEM = ITEMID	FK_RECETTE_ITEM

### Linked from

Table		Join	Title / Name / Description
←	ITEM_RECETTE	RECETTEID = ITEM_RECETTEID_RECETTE	FK_RECETTE_ITEM_RECETTE
←	METIER_RECETTE	RECETTEID = METIER_RECETTEID_RECETTE	FK_RECETTE_MR
←	PERSONNAGE_RECETTE	RECETTEID = PERSONNAGE_RECETTEID_RECETTE	FK_RECETTE_PR

### Unique keys

Columns		Name / Description
🔑	ID	SYS_C0040542

### Triggers

Name		When	Description
⚡	NEW_RECETTE_TRIGGER	Before Insert	
<pre>TRIGGER New_Recette_trigger BEFORE INSERT ON RECETTE FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := RECETTE_seq.NEXTVAL; END;</pre>			

### Uses

		Name
█	RECETTE	
→	ITEM	
⚡	NEW_RECETTE_TRIGGER	
█	RECETTE	

### Used By

		Name
█	RECETTE	

Name
⚡ NEW_RECETTE_TRIGGER
→ ITEM_RECETTE
→ METIER_RECETTE
→ PERSONNAGE_RECETTE

#### 4.1.49. Table: SUITEQUETE

##### Columns

		Name	Data type	Description / Attributes
目	键	ID_FILS	NUMBER(5, 0)	References: QUETE
目	键	ID_PERE	NUMBER(5, 0)	References: QUETE

##### Links to

Table		Join	Title / Name / Description
→	QUETE	SUITEQUETEID_FILS = QUETEID	FK_QUETEFILS_SUITEQUETE
→	QUETE	SUITEQUETEID_PERE = QUETEID	FK_QUETEPERE_SUITEQUETE

##### Unique keys

Columns		Name / Description
键	ID_FILS, ID_PERE	SYS_C0040657

##### Uses

		Name
表格	SUITEQUETE	
→	QUETE	
→	QUETE	

## 4.1.50. Table: TITRE

### Columns

		Name	Data type	Description / Attributes
█	🔑	ID	NUMBER	
█		NOM	VARCHAR2(255 BYTE)	
█		DESCRIPTION	VARCHAR2(255 BYTE)	<b>Nullable</b>
█		METHODEOPTENTION	VARCHAR2(255 BYTE)	<b>Nullable</b> <b>Default:</b> NULL

### Linked from

Table		Join	Title / Name / Description
→	APPARENCE_TITRE	TITREID = APPARENCE_TITREID_TITRE	FK_TITRE_AT
→	APPARENCEPOSSIBLE_TITRE	TITREID = APPARENCEPOSSIBLE_TITREID_TITRE	FK_TITRE_APT
→	PERSONNAGE_TITRE	TITREID = PERSONNAGE_TITREID_TITRE	FK_TITRE_PT

### Unique keys

Columns		Name / Description
🔑	ID	SYS_C0040510

### Triggers

Name		When	Description
⚡	NEW_TITRE_TRIGGER	Before Insert	
<pre>TRIGGER New_Titre_trigger BEFORE INSERT ON TITRE FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := TITRE_seq.NEXTVAL; END;</pre>			

### Uses

		Name
⚡	NEW_TITRE_TRIGGER	
█	TITRE	

### Used By

		Name
█	TITRE	
⚡	NEW_TITRE_TRIGGER	
→	APPARENCE_TITRE	
→	APPARENCEPOSSIBLE_TITRE	
→	PERSONNAGE_TITRE	

#### 4.1.51. Table: TYPELIEUX

##### Columns

		Name	Data type	Description / Attributes
█	█	ID	NUMBER	
█		NOM	VARCHAR2(255 BYTE)	
█		DESCRIPTION	CLOB	Nullable

##### Linked from

Table		Join	Title / Name / Description
→	LIEUX	TYPELIEUXID = LIEUXID_TYPELIEUX	FK_TYPELIEUX_LIEUX

##### Unique keys

Columns		Name / Description
█	ID	SYS_C0040545

##### Triggers

Name		When	Description
⚡	NEW_TYPELIEUX_TRIGGER	Before Insert	
<pre>TRIGGER New_Typelieux_trigger   BEFORE INSERT ON TYPELIEUX   FOR EACH ROW   BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := TYPELIEUX_seq.NEXTVAL;   END;</pre>			

##### Uses

Name	
⚡	NEW_TYPELIEUX_TRIGGER
█	TYPELIEUX

##### Used By

Name	
█	TYPELIEUX
⚡	NEW_TYPELIEUX_TRIGGER
→	LIEUX

## 4.1.52. Table: TYPEMONSTRE

### Columns

		Name	Data type	Description / Attributes
		ID	NUMBER	
		NOM	VARCHAR2(255 BYTE)	
		DESCRIPTION	CLOB	Nullable

### Linked from

Table		Join	Title / Name / Description
	MONSTRE_TYPEMONSTRE	<b>TYPEMONSTREID =</b> MONSTRE_TYPEMONSTREID_TYPEMONSTRE	FK_TYPEMONSTRE_MT

### Unique keys

Columns		Name / Description
	ID	SYS_C0040501

### Triggers

Name		When	Description
	NEW_TYPEMONSTRE_TRIGGER	Before Insert	
<pre>TRIGGER New_TypeMonstre_trigger BEFORE INSERT ON TYPEMONSTRE FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := TYPEMONSTRE_seq.NEXTVAL; END;</pre>			

### Uses

		Name
	NEW_TYPEMONSTRE_TRIGGER	
	TYPEMONSTRE	

### Used By

		Name
	TYPEMONSTRE	
	NEW_TYPEMONSTRE_TRIGGER	
	MONSTRE_TYPEMONSTRE	

## 4.1.53. Table: TYPEQUETE

### Columns

Name		Data type	Description / Attributes
	ID	NUMBER	
	DESCRIPTION	CLOB	Nullable

### Linked from

Table		Join	Title / Name / Description
	QUETE	TYPEQUETEID = QUETEID_TYPEQUETE	FK_TYPEQUETE_QUETE

### Unique keys

Columns		Name / Description
	ID	SYS_C0040549

### Triggers

Name		When	Description
	NEW_TYPEQUETE_TRIGGER	Before Insert	
<pre>TRIGGER New_Typequete_trigger BEFORE INSERT ON TYPEQUETE FOR EACH ROW BEGIN     -- Utilisation de la séquence pour générer la valeur Id     :NEW.Id := TYPEQUETE_seq.NEXTVAL; END;</pre>			

### Uses

Name	
	NEW_TYPEQUETE_TRIGGER
	TYPEQUETE

### Used By

Name	
	TYPEQUETE
	NEW_TYPEQUETE_TRIGGER
	QUETE

## 4.2. Procedures

### 4.2.1. Procedure: CLEANALL

#### Input/Output

Name	Data type	Description
@ P_FORCE	PL/SQL BOOLEAN	

#### Uses

Name
CLEANALL
SYS.STANDARD
SYS.SYS_STUB_FOR_PURITY_ANALYSIS

#### Script

```
PROCEDURE CleanAll(p_force IN BOOLEAN DEFAULT TRUE) AS
BEGIN
    -- Supprimer les déclencheurs
    FOR tr IN (SELECT trigger_name FROM user_triggers) LOOP
        BEGIN
            EXECUTE IMMEDIATE 'DROP TRIGGER ' || tr.trigger_name;
            DBMS_OUTPUT.PUT_LINE('Déclencheur supprimé : ' || tr.trigger_name);
        EXCEPTION
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('Erreur lors de la suppression du déclencheur : ' || tr.trigger_name || ', erreur : ' || SQLERRM);
        END;
    END LOOP;

    -- Supprimer les vues
    FOR v IN (SELECT view_name FROM user_views) LOOP
        BEGIN
            EXECUTE IMMEDIATE 'DROP VIEW ' || v.view_name;
            DBMS_OUTPUT.PUT_LINE('Vue supprimée : ' || v.view_name);
        EXCEPTION
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('Erreur lors de la suppression de la vue : ' || v.view_name || ', erreur : ' || SQLERRM);
        END;
    END LOOP;

    -- Supprimer les tables avec CASCADE CONSTRAINTS
    FOR t IN (SELECT table_name FROM user_tables) LOOP
        BEGIN
            EXECUTE IMMEDIATE 'DROP TABLE ' || t.table_name || ' CASCADE CONSTRAINTS';
            DBMS_OUTPUT.PUT_LINE('Table supprimée : ' || t.table_name);
        EXCEPTION
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('Erreur lors de la suppression de la table : ' || t.table_name || ', erreur : ' || SQLERRM);
        END;
    END LOOP;

    -- Supprimer les séquences
    FOR s IN (SELECT sequence_name FROM user_sequences) LOOP
        BEGIN
            EXECUTE IMMEDIATE 'DROP SEQUENCE ' || s.sequence_name;
            DBMS_OUTPUT.PUT_LINE('Séquence supprimée : ' || s.sequence_name);
        EXCEPTION
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('Erreur lors de la suppression de la séquence : ' || s.sequence_name || ', erreur : ' || SQLERRM);
        END;
    END LOOP;

    COMMIT;
END;
```

