



SAÉ 1.02
Création d'une base de données

~ PROJET TRANS ~

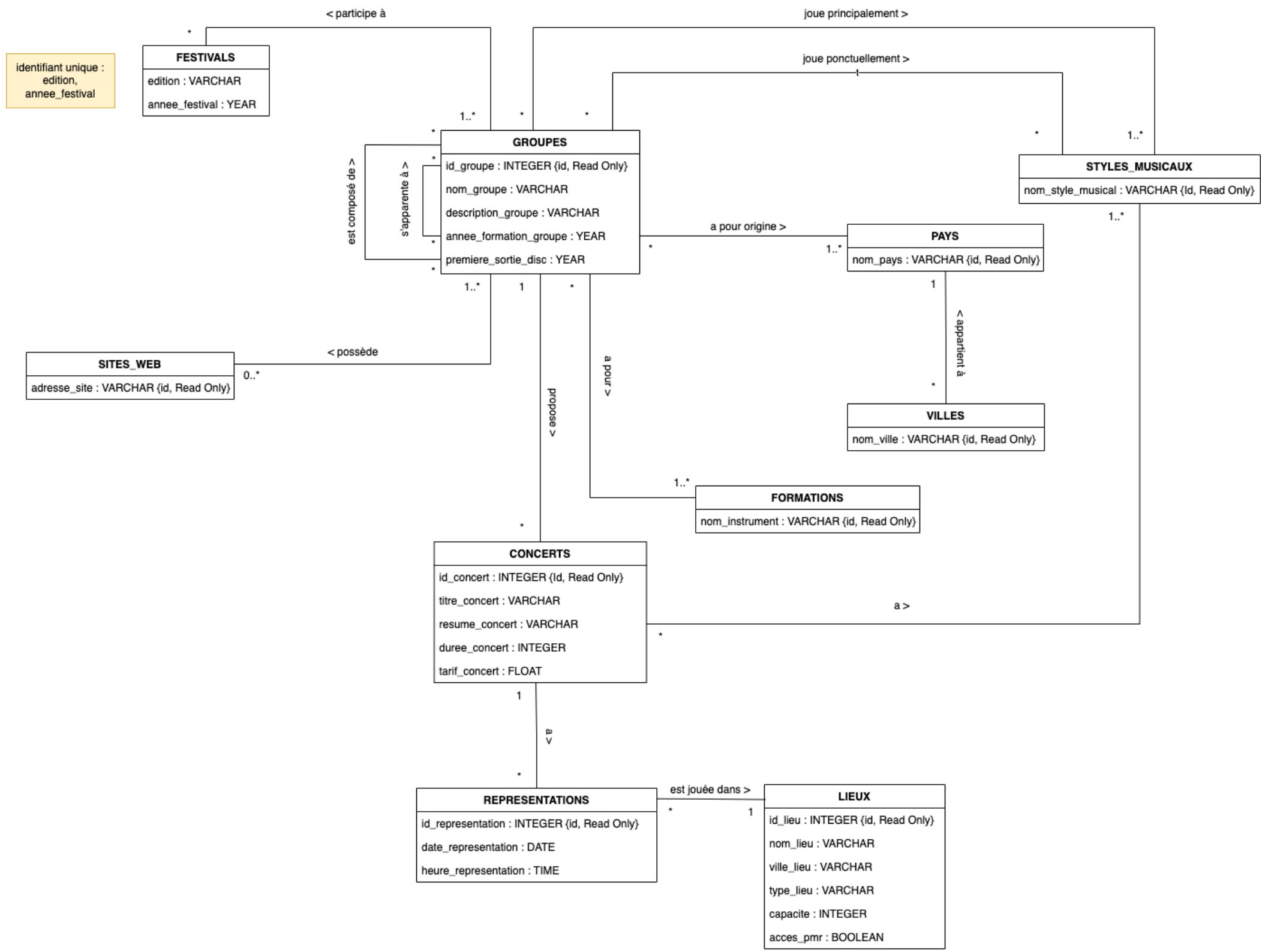
Partie 1

LES
TRANS
NOUVEAU DEPUIS 1979

GROUPE D2

Corentin [REDACTED]

Justine Verger



~ DESCRIPTION DES TABLES ET DE LEURS ATTRIBUTS ~

FESTIVALS

La classe FESTIVALS regroupe les informations relatives aux festivals.

Description des attributs :

- edition : VARCHAR
=> correspond au numéro et au nom de l'édition du festival
- annee_festival : YEAR
=> correspond à l'année durant laquelle a eu lieu le festival

Clé primaire :

La clé primaire est composée des attributs `edition` et `annee_festival`.

Multiplicités :

- La multiplicité allant de FESTIVALS vers GROUPES est de 1..* car un festival peut accueillir un ou plusieurs groupes.

Exemple d'extension de la table FESTIVALS :

edition	annee_festival
1ères Rencontres Trans Musicales de Rennes	1979

GROUPES

La classe GROUPES regroupe les informations relatives aux groupes/artistes.

Description des attributs :

- id_groupe : INTEGER
=> correspond à l'identifiant unique du groupe
- nom_groupe : VARCHAR
=> correspond au nom du groupe
- description_groupe : VARCHAR
=> correspond à un bref résumé décrivant le groupe
- annee_formation_groupe : YEAR
=> correspond à l'année de formation du groupe
- premiere_sortie_disc : YEAR
=> correspond à la date de sortie du premier disque du groupe

Clé primaire :

L'attribut `id_groupe` représente la clé primaire de la classe GROUPES.

Multiplicités :

- La multiplicité allant de GROUPES vers FESTIVALS est de *, car un groupe peut ou peut ne pas participer à un festival.
- La première multiplicité allant de GROUPES vers STYLES_MUSICAUX (joue principalement >) est de 1..*, car un groupe possède forcément au moins un style de musique.
- La seconde multiplicité allant de GROUPES vers STYLES_MUSICAUX (joue ponctuellement >) est de *, car un groupe ne possède pas forcément de style secondaire (mais peut toutefois en posséder plusieurs).
- La multiplicité allant de GROUPES vers PAYS est de 1..*, car un groupe est forcément originaire d'un pays.
- La multiplicité allant de GROUPES vers FORMATIONS est de 1..*, car un groupe a forcément besoin d'un instrument au minimum pour pouvoir exister (étant ici considéré comme instrument toute technique émettant un son (percussions corporelles incluses)).
- La multiplicité allant de GROUPES vers CONCERTS est de *, car un groupe peut ou peut ne pas proposer de concert (si par exemple, il a été enregistré dans la base de donnée lors d'une édition x et qu'il ne participe pas aux suivantes).
- La multiplicité allant de GROUPES vers SITES_WEB est de 0..*, car un groupe peut ne pas avoir de site web, mais peut tout aussi bien en avoir plusieurs.
- La classe GROUPE est en relation avec elle-même pour exprimer le cas d'un groupe par exemple composé de deux groupes/artistes existant indépendamment : sa multiplicité est donc de * dans les deux sens.
- La classe GROUPE est en relation avec elle-même pour exprimer sa parenté éventuelle avec un ou plusieurs autre(s) groupe(s) : sa multiplicité est donc de * dans les deux sens.

Exemple d'extension de la table GROUPES :

id_groupe	nom_groupe	description_groupe	annee_formati on_groupe	premiere_sort ie_disc
00358	Sebastian	Chanteur solo avec de super musiciens	2004	2005

CONCERTS

La classe CONCERTS regroupe les informations relatives aux concerts.

Description des attributs :

- id_concert : INTEGER
=> correspond à l'identifiant unique du concert
- titre_concert : VARCHAR
=> correspond au titre du concert
- resume_concert : VARCHAR
=> correspond à un bref texte résumant le contenu du concert
- duree_concert : INTEGER
=> correspond à la durée du concert (exprimée en minutes)

- `tarif_concert` : FLOAT
=> correspond au prix du concert

Clé primaire :

L'attribut `id_concert` représente la clé primaire de la classe `CONCERT`.

Multiplicités :

- La multiplicité allant de `CONCERTS` vers `GROUPES` est de 1, car un concert ne peut être proposé que par un et seulement un groupe.
- La multiplicité allant de `CONCERTS` vers `STYLE_MUSICAUX` est de 1..*, car un concert peut avoir un ou plusieurs styles musicaux.
- La multiplicité allant de `CONCERTS` vers `REPRÉSENTATIONS` est de *, car un concert peut, comme peut ne pas, avoir de représentations.

Exemple d'extension de la table `CONCERTS` :

<code>titre_concert</code>	<code>resume_concert</code>	<code>duree_concert</code>	<code>tarif_concert</code>
L'Enfance	Vous allez vous retrouver en enfance. Replongez dans vos souvenirs éloignés et voyagez !	1h30	35€

REPRESENTATIONS

La classe `REPRESENTATIONS` regroupe les informations relatives aux représentations (uniques) données par un groupe.

Description des attributs :

- `num_concert` : INTEGER
=> correspond à l'identifiant unique de la représentation
- `date_concert` : DATE
=> correspond à la date de la représentation

Clé primaire :

L'attribut `num_concert` représente la clé primaire de la classe `REPRESENTATIONS`.

Multiplicités :

- La multiplicité allant de `REPRESENTATIONS` vers `CONCERTS` est de 1, car une représentation ne peut être associée qu'à un seul concert.
- La multiplicité allant de `REPRESENTATIONS` vers `LIEUX` est de 1, car une représentation ne peut être jouée que dans un seul lieu.

Exemple d'extension de la table REPRESENTATIONS :

num_concert	date_concert
00279	05/12/2008

LIEUX

La classe LIEUX regroupe les informations relatives aux lieux (salles de concert, bars,...) dans lesquels sont joués les concerts.

Description des attributs :

- id_lieu : INTEGER
=> correspond à l'identifiant unique du lieu dans lequel est joué un concert
- nom_lieu : VARCHAR
=> correspond au nom du lieu où est joué un concert
- ville_lieu : VARCHAR
=> correspond à la ville où se trouve le lieu
- type_lieu : VARCHAR
=> correspond au type de lieu (bar, salle de concert, ...)
- capacite : INTEGER
=> correspond à la capacité maximale d'accueil du lieu
- acces_pmr : BOOLEAN
=> correspond à la présence ou non d'un accès facilité pour les personnes à mobilité réduite (PMR).

Clé primaire :

L'attribut id_lieu représente la clé primaire de la classe LIEUX.

Multiplicités :

- La multiplicité allant de LIEUX vers REPRESENTATION est de *, car un lieu peut accueillir aucune, comme une infinité de représentations différentes.

Exemple d'extension de la table LIEUX :

id_lieu	nom_lieu	ville_lieu	type_lieu	capacite	acces_pmr
00004	Antipode	Rennes	bar	58	True

STYLES_MUSICAUX

La classe STYLES_MUSICAUX contient le nom du ou des styles musicaux.

Description des attributs :

- nom_style_musical : VARCHAR
=> correspond au nom du style musical

Clé primaire :

L'attribut `nom_style_musical` représente la clé primaire de la classe `STYLES_MUSICAUX`.

Multiplicités :

- La première multiplicité allant de `STYLES_MUSICAUX` vers `GROUPES` est de *, car un style musical peut être le style principal d'aucun, comme de plusieurs groupes.
- La première multiplicité allant de `STYLES_MUSICAUX` vers `GROUPES` est de *, car un style musical peut être le style secondaire d'aucun, comme de plusieurs groupes.
- La multiplicité allant de `STYLES_MUSICAUX` vers `CONCERTS` est de *, car un style musical peut être le style d'aucun comme de plusieurs concerts.

Exemple d'extension de la table `STYLES_MUSICAUX` :

<u>nom_style_musical</u>
jazz

FORMATIONS

La classe `FORMATIONS` contient le nom du ou des instruments joués par un groupe.

Description des attributs :

- nom_instrument : VARCHAR
=> correspond au nom d'un instrument

Clé primaire :

L'attribut `nom_instrument` représente la clé primaire de la classe `FORMATIONS`.

Multiplicités :

- La multiplicité allant de `FORMATIONS` vers `GROUPES` est de *, car une même formation peut être celle d'aucun, comme de plusieurs groupes.

Exemple d'extension de la table `FORMATIONS` :

<u>nom_instrument</u>
guitare

PAYS

La classe `PAYS` contient le nom du ou des pays dont est originaire le groupe.

Description des attributs :

- `nom_pays` : VARCHAR
=> correspond au nom du pays dont est originaire le groupe

Clé primaire :

L'attribut `nom_pays` représente la clé primaire de la classe `GROUPES`.

Multiplicités :

- La multiplicité allant de `PAYS` vers `GROUPES` est de *, car un pays peut être le pays d'origine d'aucun, comme de plusieurs groupes.

Exemple d'extension de la table `PAYS` :

<code>nom_pays</code>
France

VILLES

La classe `VILLES` contient le nom de la ou des villes dont est originaire le groupe.

Description des attributs :

- `nom_ville` : VARCHAR
=> correspond au nom de la ville dont est originaire le groupe

Clé primaire :

L'attribut `nom_ville` représente la clé primaire de la classe `VILLES`.

Multiplicités :

- La multiplicité allant de `VILLES` vers `PAYS` est de 1, car une ville appartient forcément à un pays et un seul.

Exemple d'extension de la table `VILLES` :

<code>nom_ville</code>
Nantes

SITES_WEB

La classe `SITES_WEB` contient l'URL du ou des site(s) du groupe.

Description des attributs :

- adresse_site : VARCHAR
=> correspond à l'adresse du site web du groupe

Clé primaire :

L'attribut `adresse_site` représente la clé primaire de la classe `SITE_WEB`.

Multiplicités :

- La multiplicité allant de `SITE_WEB` vers `GROUPES` est de 1..*, car un site web peut être possédé par un ou plusieurs groupes (site web commun, par exemple).

Exemple d'extension de la table `SITES_WEB` :

adresse_site
http://www.20/20-pour-les-verger.univ-rennes1.fr