

SAE « Conception d'une BD pour le festival Nancy Rock Vibration »

Sujet BD

Pour concevoir cette BD, procéder par les étapes suivantes :

1. Établir un dictionnaire des données utilisées dans NRV.net
2. Donner l'ensemble des dépendances fonctionnelles valides entre ces données
3. Déterminer la ou les clés minimales (justifier votre réponse)
4. Vérifier si une relation universelle pour NRV.net vérifie la 3FN (justifier votre réponse).
5. A défaut de la 3FN, décomposer la relation universelle en un ensemble de relations vérifiant la 3FN.
6. Fournir le script SQL pour créer toutes les tables ainsi que quelques enregistrements.

-SOIREE (**idsoiree**, nomSoirée, thematique, dateSoiree, horaireDebut, idLieu)

-SPECTACLE(**idspectacle**, titre, description, video, horaireSpec, dureeSpec, idstyle)

-JOUER(idspectacle, idArtiste)

-APPARTIENT(idSoiree, idspectacle)

-STYLE(**idStyle**, nomStyle)

-LIEU(**idLieu**, nomLieu, adresse, nbPlacesAssises, nbPlacesDebout)

-ARTISTE(**idArtiste**, nomArtiste)

-IMAGESPEC(idspectacle, idImage)

-IMAGELIEU(idLieu, idImage)

-IMAGE(**idImage**, chemin)

-USERS(**idUser**, username, email, password, role)

-PREFERENCE(idUser, idSpectacle)

1 Établir un dictionnaire des données utilisées dans NRV.net :

Nom	Description	Type
idSoiree	id de la soirée	number(3,0)
nomSoirée	titre de la soirée	varchar(50)
thematique	description du thème de la soirée	varchar(50)
dateSoiree	date d'une soirée	date(jour,mois,annee)
horaireDebut	date du début de la soirée	date(heure)
idLieu	id du lieu	number(3,0)
idSpectacle	id d'un spectacle	number(3,0)
titre	titre du spectacle	varchar(50)
description	description du spectacle	varchar(100)
video	lien de la vidéo du spectacle	varchar(100)
horaireSpec	date du début du spectacle	date(heure)
dureeSpec	durée du spectacle	number(3,0) [en minutes]
idArtiste	id de l'artiste	number(3,0)
idStyle	id du style de musique	number(2,0)
nomStyle	nom du style de musique	varchar(50)
idLieu	id du lieu	number(2,0)
nomLieu	nom du lieu	varchar(50)
adresse	adresse du lieu	varchar(100)
nbPlacesAssises	nombre de places assises	number(4,0)
nbPlacesDebout	nombre de places debout	number(4,0)
nomArtiste	nom de l'artiste	varchar(50)
idImage	id de l'image	number(3,0)

idUser	id de l'utilisateur	number(5,0)
username	nom de l'utilisateur	varchar(50)
email	email de l'utilisateur	varchar(100)
password	mot de passe de l'utilisateur	varchar(100)
role	rôle de l'utilisateur (standard, staff ou admin)	varchar(10)

2 Donner l'ensemble des dépendances fonctionnelles valides entre ces données :

Dépendances fonctionnelles :

idSoiree -> (nomSoirée, thematique, dateSoiree, horaireDebut, idLieu)

idSoiree -> (nomLieu, adresse, nbPlacesAssises, nbPlacesDebout)

(dateSoiree, horaireDebut) -> idSoiree

(dateSoiree, horaireDebut) -> (nomSoirée, thematique, idLieu)

(dateSoiree, horaireDebut) -> (nomLieu, adresse, nbPlacesAssises, nbPlacesDebout)

idSpectacle -> (titre, description, video, horaireSpec, dureeSpec, idStyle)

idSpectacle -> nomStyle

idLieu -> (nomLieu, adresse, nbPlacesAssises, nbPlacesDebout)

idStyle -> nomStyle

idArtiste -> nomArtiste

idImage -> chemin

idUser -> (username, email, password, role)

3 Déterminer la ou les clés minimales (justifier votre réponse) :

Les données suivantes n'apparaissent jamais à droite d'une dépendance fonctionnelle :
idSpectacle, idArtiste, idImage et idUser.

Ces attributs sont donc dans toutes les clés.

Ensemble, ils ne forment pas une clé minimale car on ne peut pas avoir accès avec les informations d'une soirée.

Clé minimales :

(idSpectacle, idArtiste, idImage, idUser, idSoiree) :

idSpectacle permet de connaître les informations d'un spectacle ainsi que les informations d'un style

idArtiste permet d'obtenir les informations d'un artiste

idImage permet d'obtenir les informations d'une image

idUser permet d'obtenir les informations d'un utilisateur

idSoiree permet d'obtenir les informations d'une soirée ainsi que les informations d'un lieu

Toutes les informations sont donc obtenues.

(idSpectacle, idArtiste, idImage, idUser, dateSoiree, horaireDebut) :

(dateSoiree, horaireDebut) permet d'obtenir idSoiree et donc les informations concernant une soirée et un lieu

4 Vérifier si une relation universelle pour NRV.net vérifie la 3FN (justifier votre réponse) :

$R = \{\text{idSoiree}, \text{nomSoiree}, \text{thematique}, \text{dateSoiree}, \text{horaireDebut}, \text{idLieu}, \text{idSpectacle}, \text{titre}, \text{description}, \text{video}, \text{horaireSpec}, \text{dureeSpec}, \text{idStyle}, \text{idArtiste}, \text{nomArtiste}, \text{idImage}, \text{chemin}, \text{idUser}, \text{username}, \text{email}, \text{password}, \text{role}, \text{nomLieu}, \text{adresse}, \text{nbPlacesAssises}, \text{nbPlacesDebout}, \text{nomStyle}\}$

Pour qu'une relation soit en 3nf, il faut qu'elle soit en 2nf et qu'aucun attribut non clé soit identifié grâce à un autre attribut non clé.

La relation est en 1nf, pour qu'elle soit en 2nf il faut que tous les attributs dépendent de la clé entière et pas d'une partie de la clé.

La relation ici n'est pas en 2nf car par exemple, idSoiree permet d'identifier nomSoiree alors que idSoiree n'est qu'une partie de la clé (idSoiree, idSpectacle, idArtiste, idImage, idUser).

Le relation ne peut donc pas être en 3nf car elle n'est pas en 2nf.

5 A défaut de la 3FN, décomposer la relation universelle en un ensemble de relations vérifiant la 3FN :

$R = \{\text{idSoiree}, \text{nomSoiree}, \text{thematique}, \text{dateSoiree}, \text{horaireDebut}, \text{idLieu}, \text{idSpectacle}, \text{titre}, \text{description}, \text{video}, \text{horaireSpec}, \text{dureeSpec}, \text{idStyle}, \text{idArtiste}, \text{nomArtiste}, \text{idImage}, \text{chemin}, \text{idUser}, \text{username}, \text{email}, \text{password}, \text{role}, \text{nomLieu}, \text{adresse}, \text{nbPlacesAssises}, \text{nbPlacesDebout}, \text{nomStyle}\}$

On applique la méthode synthétique :

Rappel de la liste des DF :

$\text{idSoiree} \rightarrow (\text{nomSoirée}, \text{thematique}, \text{dateSoiree}, \text{horaireDebut}, \text{idLieu})$

~~$\text{idSoiree} \rightarrow (\text{nomLieu}, \text{adresse}, \text{nbPlacesAssises}, \text{nbPlacesDebout})$~~

$(\text{dateSoiree}, \text{horaireDebut}) \rightarrow \text{idSoiree}$

~~$(\text{dateSoiree}, \text{horaireDebut}) \rightarrow (\text{nomSoirée}, \text{thematique}, \text{idLieu})$~~

~~$(\text{dateSoiree}, \text{horaireDebut}) \rightarrow (\text{nomLieu}, \text{adresse}, \text{nbPlacesAssises}, \text{nbPlacesDebout})$~~

$\text{idSpectacle} \rightarrow (\text{titre}, \text{description}, \text{video}, \text{horaireSpec}, \text{dureeSpec}, \text{idStyle})$

~~$\text{idSpectacle} \rightarrow \text{nomStyle}$~~

$\text{idLieu} \rightarrow (\text{nomLieu}, \text{adresse}, \text{nbPlacesAssises}, \text{nbPlacesDebout})$

$\text{idStyle} \rightarrow \text{nomStyle}$

$\text{idArtiste} \rightarrow \text{nomArtiste}$

$\text{idImage} \rightarrow \text{chemin}$

$\text{idUser} \rightarrow (\text{username}, \text{email}, \text{password}, \text{role})$

$\text{idSoiree} \rightarrow (\text{nomLieu}, \text{adresse}, \text{nbPlacesAssises}, \text{nbPlacesDebout})$ est une relation transitive car $\text{idSoiree} \rightarrow \text{idLieu}$ et $\text{idLieu} \rightarrow (\text{nomLieu}, \text{adresse}, \text{nbPlacesAssises}, \text{nbPlacesDebout})$

c'est la même chose pour $(\text{dateSoiree}, \text{horaireDebut}) \rightarrow (\text{nomSoirée}, \text{thematique}, \text{idLieu})$ et $(\text{dateSoiree}, \text{horaireDebut}) \rightarrow (\text{nomLieu}, \text{adresse}, \text{nbPlacesAssises}, \text{nbPlacesDebout})$

c'est la même chose pour $\text{idSpectacle} \rightarrow \text{nomStyle}$

On a donc les DF :

$\text{idSoiree} \rightarrow (\text{nomSoirée}, \text{thematique}, \text{dateSoiree}, \text{horaireDebut}, \text{idLieu})$

(dateSoiree, horaireDebut) -> idSoiree

idSpectacle -> (titre, description, video, horaireSpec, dureeSpec, idStyle)

idLieu -> (nomLieu, adresse, nbPlacesAssises, nbPlacesDebout)

idStyle -> nomStyle

idArtiste -> nomArtiste

idImage -> chemin

idUser -> (username, email, password, role)

Par décomposition avec les df on peut avoir les tables :

SOIREE(**idSoiree**, nomSoirée, thematique, dateSoiree, horaireDebut, idLieu)
SPECTACLE(**idSpectacle**, titre, description, video, horaireSpec, dureeSpec, idStyle)
LIEU(**idLieu**, nomLieu, adresse, nbPlacesAssises, nbPlacesDebout)
STYLE(**idStyle**, nomStyle)
ARTISTE(**idArtiste**, nomArtiste)
IMAGE(**idImage**, chemin)
USER(**idUser**, username, email, password, role)

Il faut maintenant relier les clés entre elles :

APPARTIENT(**idSoiree**, **idSpectacle**) : permet de savoir à quelle soirée appartient un spectacle

JOUER(**idSpectacle**, **idArtiste**) : permet de savoir quels artistes jouent dans un spectacle

IMAGESPEC(**idImage**, **idSpectacle**) : permet d'associer des images à un spectacle

IMAGELIEU(**idImage**, **idLieu**) : permet d'associer des images à un lieu de spectacle

PREFERENCE(**idUser**, **idSpectacle**) : permet d'identifier les préférences d'un utilisateur si il est identifié

On a donc les tables :

SOIREE(**idSoiree**, nomSoiree, thematique, dateSoiree, horaireDebut, idLieu)
SPECTACLE(**idSpectacle**, titre, description, video, horaireSpec, dureeSpec, idStyle)
LIEU(**idLieu**, nomLieu, adresse, nbPlacesAssises, nbPlacesDebout)
STYLE(**idStyle**, nomStyle)
ARTISTE(**idArtiste**, nomArtiste)
IMAGE(**idImage**, chemin)
USER(**idUser**, username, email, password, role)

APPARTIENT(idSoiree, idSpectacle)
JOUER(idSpectacle, idArtiste)
IMAGESPEC(idImage, idSpectacle)
IMAGELIEU(idImage, idLieu)
PREFERENCE(idUser, idSpectacle)

Toutes ces tables sont en 3nf, la relation respecte donc la 3nf.

6 Fournir le script SQL pour créer toutes les tables ainsi que quelques enregistrements :

```
-- Table LIEU
CREATE TABLE LIEU (
    idLieu INT PRIMARY KEY AUTO_INCREMENT,
    nomLieu VARCHAR(50),
    adresse VARCHAR(100),
    nbPlacesAssises INT,
    nbPlacesDebout INT
);

-- Table STYLE
CREATE TABLE STYLE (
    idStyle INT PRIMARY KEY AUTO_INCREMENT,
    nomStyle VARCHAR(50)
);

-- Table SOIREE
CREATE TABLE SOIREE (
    idSoiree INT PRIMARY KEY AUTO_INCREMENT,
    nomSoiree VARCHAR(50),
    thematique VARCHAR(50),
    dateSoiree DATE,
    horaireDebut TIME,
    idLieu INT,
    FOREIGN KEY (idLieu) REFERENCES LIEU(idLieu) ON DELETE CASCADE
);

-- Table SPECTACLE
CREATE TABLE SPECTACLE (
    idSpectacle INT PRIMARY KEY AUTO_INCREMENT,
    titre VARCHAR(50),
    description TEXT,
    video VARCHAR(100),
    horaireSpec TIME,
```

```

    dureeSpec INT, -- Durée en minutes
    idStyle INT,
    FOREIGN KEY (idStyle) REFERENCES STYLE(idStyle) ON DELETE SET NULL
);

-- Table ARTISTE
CREATE TABLE ARTISTE (
    idArtiste INT PRIMARY KEY AUTO_INCREMENT,
    nomArtiste VARCHAR(50)
);

-- Table IMAGE
CREATE TABLE IMAGE (
    idImage INT PRIMARY KEY AUTO_INCREMENT,
    chemin VARCHAR(255)
);

-- Table JOUER (relation entre SPECTACLE et ARTISTE)
CREATE TABLE JOUER (
    idSpectacle INT,
    idArtiste INT,
    FOREIGN KEY (idSpectacle) REFERENCES SPECTACLE(idSpectacle) ON DELETE
CASCADE,
    FOREIGN KEY (idArtiste) REFERENCES ARTISTE(idArtiste) ON DELETE CASCADE,
    PRIMARY KEY (idSpectacle, idArtiste)
);

-- Table APPARTIENT (relation entre SOIREE et SPECTACLE)
CREATE TABLE APPARTIENT (
    idSoiree INT,
    idSpectacle INT,
    FOREIGN KEY (idSoiree) REFERENCES SOIREE(idSoiree) ON DELETE CASCADE,
    FOREIGN KEY (idSpectacle) REFERENCES SPECTACLE(idSpectacle) ON DELETE
CASCADE,
    PRIMARY KEY (idSoiree, idSpectacle)
);

-- Table IMAGESPEC (relation entre SPECTACLE et IMAGE)
CREATE TABLE IMAGESPEC (
    idSpectacle INT,
    idImage INT,
    FOREIGN KEY (idSpectacle) REFERENCES SPECTACLE(idSpectacle) ON DELETE
CASCADE,
    FOREIGN KEY (idImage) REFERENCES IMAGE(idImage) ON DELETE CASCADE,
    PRIMARY KEY (idSpectacle, idImage)
);

-- Table IMAGELIEU (relation entre LIEU et IMAGE)

```



```
CREATE TABLE IMAGELIEU (  
    idLieu INT,  
    idImage INT,  
    FOREIGN KEY (idLieu) REFERENCES LIEU(idLieu) ON DELETE CASCADE,  
    FOREIGN KEY (idImage) REFERENCES IMAGE(idImage) ON DELETE CASCADE,  
    PRIMARY KEY (idLieu, idImage)  
);
```

-- Table USERS

```
CREATE TABLE USERS (  
    idUser INT PRIMARY KEY AUTO_INCREMENT,  
    username VARCHAR(50) NOT NULL UNIQUE,  
    email VARCHAR(100) NOT NULL UNIQUE,  
    password VARCHAR(100) NOT NULL,  
    role VARCHAR(10)  
);
```

-- Table PREFERENCE (relation entre USERS et SPECTACLE)

```
CREATE TABLE PREFERENCE (  
    idUser INT,  
    idSpectacle INT,  
    FOREIGN KEY (idUser) REFERENCES USERS(idUser) ON DELETE CASCADE,  
    FOREIGN KEY (idSpectacle) REFERENCES SPECTACLE(idSpectacle) ON DELETE  
CASCADE,  
    PRIMARY KEY (idUser, idSpectacle)  
);
```

-- Insertion de données de test

-- Insertion de lieux

```
INSERT INTO LIEU (nomLieu, adresse, nbPlacesAssises, nbPlacesDebout) VALUES  
( 'Grande Salle', '123 Rue de la Musique', 500, 200),  
( 'Théâtre Municipal', '456 Avenue des Arts', 300, 100),  
( 'Parc des Expositions', '789 Boulevard du Rock', 800, 500);
```

-- Insertion de styles de musique

```
INSERT INTO STYLE (nomStyle) VALUES  
( 'Rock'),  
( 'Blues'),  
( 'Jazz'),  
( 'Metal'),  
( 'Pop');
```

-- Insertion de soirées

```
INSERT INTO SOIREE (nomSoiree, thematique, dateSoiree, horaireDebut, idLieu) VALUES  
( 'Soirée Rock Legends', 'Classic Rock', '2024-11-20', '19:00:00', 1),  
( 'Soirée Metal Madness', 'Metal', '2024-11-21', '20:00:00', 2),  
( 'Soirée Blues Vibes', 'Blues', '2024-11-22', '18:30:00', 1),
```

('Soirée Jazz Night', 'Jazz', '2024-11-23', '19:00:00', 3),
('Soirée Rock Revival', 'Rock', '2024-11-24', '20:30:00', 2),
('Soirée Pop Party', 'Pop', '2024-11-25', '21:00:00', 3);

-- Insertion de spectacles avec la durée en minutes

INSERT INTO SPECTACLE (titre, description, video, horaireSpec, dureeSpec, idStyle)
VALUES

('The Classic Rock Show', 'Un hommage aux légendes du rock', 'http://video.com/rockshow',
'19:30:00', 90, 1),
('Metal Overdrive', 'Concert de heavy metal intense', 'http://video.com/metaloverdrive',
'20:15:00', 120, 4),
('Blues Revival', 'Ambiance blues avec les meilleurs artistes', 'http://video.com/bluesrevival',
'18:45:00', 110, 2),
('Smooth Jazz Evening', 'Jazz classique et moderne', 'http://video.com/jazzevening',
'19:15:00', 75, 3),
('Rock Revival', 'Retour aux classiques du rock', 'http://video.com/rockrevival', '20:45:00', 105,
1),
('Pop Extravaganza', 'Concert pop pour toute la famille', 'http://video.com/popextravaganza',
'21:15:00', 80, 5);

-- Insertion d'artistes

INSERT INTO ARTISTE (nomArtiste) VALUES

('The Rockers'),
('Metal Fury'),
('Blues Brothers'),
('Jazz Masters'),
('Pop Stars'),
('Heavy Hitters'),
('Smooth Vibes');

-- Association des artistes aux spectacles (table JOUER) pour plusieurs groupes par spectacle

INSERT INTO JOUER (idSpectacle, idArtiste) VALUES

(1, 1), -- The Rockers jouent dans "The Classic Rock Show"
(1, 6), -- Heavy Hitters jouent aussi dans "The Classic Rock Show"
(2, 2), -- Metal Fury joue dans "Metal Overdrive"
(2, 6), -- Heavy Hitters jouent aussi dans "Metal Overdrive"
(3, 3), -- Blues Brothers jouent dans "Blues Revival"
(4, 4), -- Jazz Masters jouent dans "Smooth Jazz Evening"
(4, 7), -- Smooth Vibes jouent aussi dans "Smooth Jazz Evening"
(5, 1), -- The Rockers jouent aussi dans "Rock Revival"
(6, 5); -- Pop Stars jouent dans "Pop Extravaganza"

-- Association des spectacles aux soirées (table APPARTIENT)

INSERT INTO APPARTIENT (idSoiree, idSpectacle) VALUES

(1, 1), -- "The Classic Rock Show" dans "Soirée Rock Legends"
(2, 2), -- "Metal Overdrive" dans "Soirée Metal Madness"
(3, 3), -- "Blues Revival" dans "Soirée Blues Vibes"
(4, 4), -- "Smooth Jazz Evening" dans "Soirée Jazz Night"

(5, 5), -- "Rock Revival" dans "Soirée Rock Revival"
(6, 6); -- "Pop Extravaganza" dans "Soirée Pop Party"

-- Insertion d'images pour les spectacles et les lieux (table IMAGE)

INSERT INTO IMAGE (chemin) VALUES

('rock_show.jpg'),
('metal_overdrive.jpg'),
('blues_revival.jpg'),
('jazz_evening.jpg'),
('rock_revival.jpg'),
('pop_extravaganza.jpg'),
('grande_salle.jpg'),
('theatre_municipal.jpg'),
('parc_expositions.jpg');

-- Association des images aux spectacles (table IMAGESPEC)

INSERT INTO IMAGESPEC (idSpectacle, idImage) VALUES

(1, 1), -- Image pour "The Classic Rock Show"
(2, 2), -- Image pour "Metal Overdrive"
(3, 3), -- Image pour "Blues Revival"
(4, 4), -- Image pour "Smooth Jazz Evening"
(5, 5), -- Image pour "Rock Revival"
(6, 6); -- Image pour "Pop Extravaganza"

-- Association des images aux lieux (table IMAGELIEU)

INSERT INTO IMAGELIEU (idLieu, idImage) VALUES

(1, 7), -- Image pour "Grande Salle"
(2, 8), -- Image pour "Théâtre Municipal"
(3, 9); -- Image pour "Parc des Expositions"

-- Insertion de comptes d'utilisateurs (table USERS)

INSERT INTO USERS (username, email, password, role) VALUES

('admin_user', 'admin@example.com', 'hashed_password_admin', 'admin'),
('staff_user', 'staff@example.com', 'hashed_password_staff', 'staff'),
('standard_user', 'user@example.com', 'hashed_password_user', 'standard');

-- Insertion de préférences utilisateurs (table PREFERENCE)

INSERT INTO PREFERENCE (idUser, idSpectacle) VALUES

(3, 1), -- standard_user a ajouté "The Classic Rock Show" à ses préférences
(3, 3), -- standard_user a ajouté "Blues Revival" à ses préférences
(2, 2); -- staff_user a ajouté "Metal Overdrive" à ses préférences