

NORMALISATION

Schéma Salle / Batiment

- 1NF: possède au moins une clé (Salle.code, Batiment.lettre) et tous ses attributs (Salle.numero, Salle.type, Salle.nb_max, Salle.bat) sont atomiques ✓
- 2NF: tout attribut n'appartenant à aucune clé candidate ne dépend pas d'une partie seulement d'une clé candidate ✓
- 3NF: toutes les DFE vers les attributs (bat => Batiment.lettre) n'appartenant pas à la clé (Salle.code), sont issues de la clé (Batiment.lettre) ✓

Schéma Spectacle / Séance / Pièce de Théâtre / Concert / Stand Up

- 1NF: possède au moins une clé (Spectacle.nom) et tous ses attributs (Spectacle.durée, Spectacle.asso_organisatrice, PieceTheatre.auteur, PieceTheatre.année, PieceTheatre.type, Concert.compositeur, Concert.année, Concert.genre, StandUp.genre, Seance.datetime) sont atomiques ✓
- 2NF: tout attribut n'appartenant à aucune clé candidate ne dépend pas d'une partie seulement d'une clé candidate ✗

Pour passer en 2NF:

Scinder Seance en Seance / Repartition:

Seance(#nom=>Spectacle.nom, #datetime=>Repartition.datetime)

Repartition(#datetime : date, salle=>Salle.code)

Nous sommes maintenant en 2NF et même en 3NF (toutes les DFE vers les attributs sont issues d'une clé).

CODE SQL

```
DROP TABLE IF EXISTS Billet;
DROP TABLE IF EXISTS CatBillet;
DROP TABLE IF EXISTS MembreAsso;
DROP TABLE IF EXISTS Exterieur;
DROP TABLE IF EXISTS Personnel;
DROP TABLE IF EXISTS Role;
DROP TABLE IF EXISTS PieceTheatre;
DROP TABLE IF EXISTS StandUp;
DROP TABLE IF EXISTS Concert;
DROP TABLE IF EXISTS Seance;
DROP TABLE IF EXISTS Spectacle;
DROP TABLE IF EXISTS Association;
DROP TABLE IF EXISTS Etudiant;
DROP TABLE IF EXISTS Personne;
DROP TABLE IF EXISTS CategorieAsso;
DROP TABLE IF EXISTS Salle;
DROP TABLE IF EXISTS Batiment;
DROP VIEW IF EXISTS v_typeSpectacle;
DROP VIEW IF EXISTS v_billet;
```

```
CREATE TABLE Batiment
(
  lettre CHAR PRIMARY KEY
);
```

```
CREATE TABLE Salle
(
  code CHAR(4) PRIMARY KEY,
  numero NUMERIC(3) NOT NULL,
  type VARCHAR NOT NULL CHECK (type='cours' OR type='amphi' OR type='bureau'),
  nb_max INTEGER NOT NULL CHECK (nb_max > 0),
  bat CHAR NOT NULL,
  FOREIGN KEY(bat) REFERENCES Batiment(lettre) ON UPDATE CASCADE
);
```

```
CREATE TABLE CategorieAsso
(
  intitule VARCHAR PRIMARY KEY
);
```

```
CREATE TABLE Personne(
  CIN CHAR(36) PRIMARY KEY,
  nom VARCHAR NOT NULL,
  prenom VARCHAR NOT NULL
);
```

```
CREATE TABLE Etudiant (
  CIN CHAR(36) PRIMARY KEY,
  FOREIGN KEY (CIN) REFERENCES Personne(CIN) ON UPDATE CASCADE
```

);

CREATE TABLE Association

```
(
nom VARCHAR PRIMARY KEY,
mail VARCHAR UNIQUE NOT NULL,
date_creation date NOT NULL,
site_web VARCHAR,
type VARCHAR NOT NULL,
salle CHAR(4) NOT NULL,
treso_asso CHAR(36) NOT NULL,
presid_asso CHAR(36) NOT NULL,
FOREIGN KEY(type) REFERENCES CategorieAsso(intitule),
FOREIGN KEY(salle) REFERENCES Salle(code) ON UPDATE CASCADE,
FOREIGN KEY(treso_asso) REFERENCES Etudiant(CIN) ON UPDATE CASCADE,
FOREIGN KEY(presid_asso) REFERENCES Etudiant(CIN) ON UPDATE CASCADE,
CHECK (treso_asso<>presid_asso)
);
```

CREATE TABLE Spectacle

```
(
nom VARCHAR PRIMARY KEY,
duree TIME NOT NULL,
asso_organisatrice VARCHAR NOT NULL,
type_sp JSON,
seance JSON,
FOREIGN KEY (asso_organisatrice) REFERENCES Association(nom) ON UPDATE CASCADE
);
```

CREATE TABLE Role

```
(
CIN CHAR(36) NOT NULL,
nom_spec VARCHAR NOT NULL,
descriptif VARCHAR NOT NULL,
FOREIGN KEY (CIN) REFERENCES Personne(CIN) ON UPDATE CASCADE,
FOREIGN KEY (nom_spec) REFERENCES Spectacle(nom) ON UPDATE CASCADE ON DELETE CASCADE,
PRIMARY KEY (CIN, nom_spec)
);
```

CREATE TABLE Personnel (

```
CIN CHAR(36) PRIMARY KEY,
statut VARCHAR NOT NULL,
FOREIGN KEY (CIN) REFERENCES Personne (CIN) ON UPDATE CASCADE
);
```

CREATE TABLE Extérieur (

```
CIN CHAR(36) PRIMARY KEY,
```

```
telephone CHAR(10) NOT NULL,  
organisme VARCHAR NOT NULL,  
FOREIGN KEY (CIN) REFERENCES Personne(CIN) ON UPDATE CASCADE  
);
```

```
CREATE TABLE MembreAsso(  
nom_asso VARCHAR,  
membre CHAR(36),  
PRIMARY KEY(nom_asso, membre),  
FOREIGN KEY(nom_asso) REFERENCES Association(nom) ON UPDATE CASCADE,  
FOREIGN KEY(membre) REFERENCES Etudiant(CIN) ON UPDATE CASCADE  
);
```

```
CREATE TABLE CatBillet (  
categorie VARCHAR NOT NULL,  
Tarif FLOAT NOT NULL CHECK (Tarif >= 0),  
PRIMARY KEY(Categorie)  
);
```

```
CREATE TABLE Billet (  
identifiant SERIAL PRIMARY KEY,  
categorie VARCHAR NOT NULL,  
date_creation DATE NOT NULL,  
seance JSON NOT NULL,  
acheteur CHAR(36),  
FOREIGN KEY (categorie) REFERENCES CatBillet (categorie),  
FOREIGN KEY (acheteur) REFERENCES Personne(CIN) ON UPDATE CASCADE  
);
```

```
INSERT INTO Batiment VALUES ('A');  
INSERT INTO Batiment VALUES ('B');  
INSERT INTO Batiment VALUES ('C');
```

```
INSERT INTO Salle VALUES ('A101', 101, 'amphi', 200, 'A');  
INSERT INTO Salle VALUES ('A102', 102, 'cours', 30, 'A');  
INSERT INTO Salle VALUES ('B200', 200, 'bureau', 4, 'B');  
INSERT INTO Salle VALUES ('C201', 201, 'cours', 30, 'C');
```

```
INSERT INTO Personne VALUES ('A0382748Z02928375KDJDH19238', 'CUVILLER', 'Charles');  
INSERT INTO Personne VALUES ('A0734748Z01993657PPAH19238', 'BEAUMER', 'Marie');  
INSERT INTO Personne VALUES ('A0991748Z03901928NDKH19238', 'XIE', 'Sihan');  
INSERT INTO Personne VALUES ('A0221748Z049162731PZH19238', 'VERRIER', 'Garance');  
INSERT INTO Personne VALUES ('A0382fefwe2928375KDJDH19238', 'Hubert', 'Anna');  
INSERT INTO Personne VALUES ('A0734748Z0fefewf7PPAH19238', 'Hubert', 'Anne');  
INSERT INTO Personne VALUES ('A0991748Z03901928NDKefewfe', 'Hubert', 'Amelie');  
INSERT INTO Personne VALUES ('fewfweewZ049162731PZH19238', 'Hubert', 'Alice');  
INSERT INTO Personne (cin, nom, prenom)VALUES ('A1635248Z27469735SGRT26533', 'Martin',  
'Albane');  
INSERT INTO Personne (cin, nom, prenom)VALUES ('A7153715T71735171HJUW28287', 'Bernard',  
'Alice');
```

```

INSERT INTO Personne (cin, nom, prenom)VALUES ('A1673371Z32718292HHGU71633', 'Thomas',
'Lucie');
INSERT INTO Personne (cin, nom, prenom)VALUES ('A7682382Z27357193UJSW85391', 'Petit',
'Nicole');
INSERT INTO Personne (cin, nom, prenom)VALUES ('A2343244D32525314FSEEC1635', 'Robert',
'Odile');
INSERT INTO Personne (cin, nom, prenom)VALUES ('A6755432O82631042DHQTD1763', 'Richard',
'Sylvie');
INSERT INTO Personne (cin, nom, prenom)VALUES ('A2176219D12422442FTRSW2342', 'Durand',
'Lucas');

```

```

INSERT INTO Etudiant VALUES ('A0382748Z02928375KDJDH19238');
INSERT INTO Etudiant VALUES ('A0991748Z03901928NDKH19238');
INSERT INTO Etudiant VALUES ('A0734748Z01993657PPAH19238');
INSERT INTO Etudiant VALUES ('A0221748Z049162731PZH19238');
INSERT INTO Etudiant VALUES ('A0382748Z02928375KDJDH19238');
INSERT INTO Etudiant VALUES ('A0734748Z01993657PPAH19238');
INSERT INTO Etudiant VALUES ('A0991748Z03901928NDKH19238');
INSERT INTO Etudiant VALUES ('A0221748Z049162731PZH19238');

```

```

INSERT INTO Personnel (cin, statut)VALUES ('A1635248Z27469735SGRT26533', 'enseignant');
INSERT INTO Personnel (cin, statut)VALUES ('A7153715T71735171HJUW28287', 'enseignant');
INSERT INTO Personnel (cin, statut)VALUES ('A2343244D32525314FSEEC1635', 'personnel
administratif');
INSERT INTO Personnel (cin, statut)VALUES ('A7682382Z27357193UJSW85391', 'personnel
administratif');
INSERT INTO Personnel (cin, statut)VALUES ('A1673371Z32718292HHGU71633', 'personnel
technique');

```

```

INSERT INTO Extérieur(cin, telephone, organisme)VALUES
('A6755432O82631042DHQTD1763','0629753891','ORMS');
INSERT INTO Extérieur(cin, telephone, organisme)VALUES
('A2176219D12422442FTRSW2342','0752631864','CSI');

```

```

INSERT INTO CatégorieAsso VALUES ('musique');
INSERT INTO CatégorieAsso VALUES ('theatre');
INSERT INTO CatégorieAsso VALUES ('arts plastiques');
INSERT INTO CatégorieAsso VALUES ('sorties');

```

```

INSERT INTO Association VALUES ('La PicaTeam', 'picateam@gmail.com',
to_date('1973-12-10','YYYY-MM-DD'), 'picateam.com', 'sorties', 'A102',
'A0991748Z03901928NDKH19238', 'A0734748Z01993657PPAH19238' );
INSERT INTO Association VALUES ('UTCroute', 'cassecroute@gmail.com',
to_date('2010-10-13','YYYY-MM-DD'), 'utcroute.com', 'arts plastiques', 'B200',
'A0221748Z049162731PZH19238', 'A0734748Z01993657PPAH19238' );

```

```

INSERT INTO MembreAsso(nom_asso, membre) VALUES ('La
PicaTeam','A0382748Z02928375KDJDH19238');
INSERT INTO MembreAsso(nom_asso, membre) VALUES ('La PicaTeam',
'A0991748Z03901928NDKH19238');
INSERT INTO MembreAsso(nom_asso, membre) VALUES ('La PicaTeam',
'A0734748Z01993657PPAH19238');

```

```

INSERT INTO MembreAsso(nom_asso, membre) VALUES ('La PicaTeam',
'A0991748Z03901928NDKefewfe');
INSERT INTO MembreAsso(nom_asso, membre) VALUES ('La PicaTeam',
'fewfweewZ049162731PZH19238');
INSERT INTO MembreAsso(nom_asso, membre) VALUES
('UTCroute','A0382748Z02928375KDJDH19238');
INSERT INTO MembreAsso(nom_asso, membre) VALUES ('UTCroute',
'A0734748Z01993657PPAH19238');
INSERT INTO MembreAsso(nom_asso, membre) VALUES ('UTCroute',
'A0221748Z049162731PZH19238');
INSERT INTO MembreAsso(nom_asso, membre) VALUES ('UTCroute',
'A0382748Z02928375KDJDH19238');
INSERT INTO MembreAsso(nom_asso, membre) VALUES ('UTCroute',
'A0991748Z03901928NDKefewfe');

```

```

INSERT INTO CatBillet VALUES ('invitation', 20.00);
INSERT INTO CatBillet VALUES ('billet etudiant', 12.50);
INSERT INTO CatBillet VALUES ('gratuit', 0.00);
INSERT INTO CatBillet VALUES ('billet exterieur', 27.50);

```

```

INSERT INTO Spectacle VALUES (
'Classicals',
'02:20',
'La PicaTeam',
'{"type":"Concert","compositeur":"Mozart","annee":1780,"genre":"Classique"}',
'{"ID":1,"salle": "A101", "datetime" : "2022-12-20 21:00"}'
);

```

```

INSERT INTO Spectacle VALUES (
'Rumeur',
'01:05',
'UTCroute',
'{"type":"StandUp","genre":"Debat"}',
'{"ID":2,"salle": "B200", "datetime" : "2023-01-20 21:00"}'
);

```

```

INSERT INTO Spectacle VALUES (
'Maree basse',
'01:05',
'UTCroute',
'{"type":"PieceTheatre","auteur":"Beranger","annee":2021,"genre":"Comedie"}',
'{"ID":3,"salle": "A101", "datetime" : "2022-12-20 21:00"}'
);

```

```

INSERT INTO Billet(categorie, date_creation, seance, acheteur) VALUES
('invitation',
to_date('2021-05-19','YYYY-MM-DD'),
'{"ID":4,"salle": "A101", "datetime" : "2022-12-20 21:00"}',
'A0382748Z02928375KDJDH19238');

```

```

INSERT INTO Billet(categorie, date_creation, seance, acheteur) VALUES
('billet etudiant',

```

```

to_date('2021-05-19','YYYY-MM-DD'),
{'ID':5,"salle": "A101", "datetime" : "2022-12-20 21:00"},
'A0382fefwe2928375KDJDH19238'
);

```

```

INSERT INTO Role(descriptif, cin, nom_spec) VALUES ('metteur en scene',
'A0382748Z02928375KDJDH19238', 'Maree basse');
INSERT INTO Role(descriptif, cin, nom_spec) VALUES ('acteur', 'A0991748Z03901928NDKH19238',
'Maree basse');
INSERT INTO Role(descriptif, cin, nom_spec) VALUES ('acteur', 'A0221748Z049162731PZH19238',
'Maree basse');
INSERT INTO Role(descriptif, cin, nom_spec) VALUES ('figurant',
'A1635248Z27469735SGRT26533', 'Maree basse');
INSERT INTO Role(descriptif, cin, nom_spec) VALUES ('souffleur',
'A7153715T71735171HJUW28287', 'Rumeur');
INSERT INTO Role(descriptif, cin, nom_spec) VALUES ('metteur en scene',
'A0382748Z02928375KDJDH19238', 'Rumeur');
INSERT INTO Role(descriptif, cin, nom_spec) VALUES ('acteur', 'A0734748Z01993657PPAH19238',
'Rumeur');
INSERT INTO Role(descriptif, cin, nom_spec) VALUES ('chef eclairagiste',
'A6755432O82631042DHQTD1763', 'Rumeur');
INSERT INTO Role(descriptif, cin, nom_spec) VALUES ('regisseur',
'A0734748Z01993657PPAH19238', 'Classicals');
INSERT INTO Role(descriptif, cin, nom_spec) VALUES ('acteur', 'A0221748Z049162731PZH19238',
'Classicals');
INSERT INTO Role(descriptif, cin, nom_spec) VALUES ('figurant',
'A6755432O82631042DHQTD1763', 'Classicals');
INSERT INTO Role(descriptif, cin, nom_spec) VALUES ('maquilleuse',
'A2176219D12422442FTRSW2342', 'Classicals');

```

```

CREATE VIEW v_typeSpectacle (Spectacle, type_nom, type_genre) AS
SELECT s.nom, t->'type' AS type_nom, t->'genre' AS type_genre
FROM Spectacle s, JSON_ARRAY_ELEMENTS(s.type_sp) t;

```

```

CREATE VIEW v_billet (nb_billets, categorie, salle) AS
SELECT count(b.identifiant) AS nb_billet, b.categorie AS categorie, s->'salle' AS salle
FROM Billet b, JSON_ARRAY_ELEMENTS(b.seance) s
GROUP BY b.categorie, s->'salle';

```

REQUEST

--Tous les étudiants

```
SELECT ps.cin, ps.nom, ps.prenom  
FROM Etudiant e JOIN Personne ps ON e.cin = ps.cin;
```

--Tout le personnel

```
SELECT ps.cin, ps.nom, ps.prenom, pl.statut  
FROM Personnel pl JOIN Personne ps ON pl.cin = ps.cin;
```

--Tous les membres extérieurs

```
SELECT ps.cin, ps.nom, ps.prenom, ex.telephone, ex.organisme  
FROM Extérieur ex JOIN Personne ps ON ex.cin = ps.cin;
```

--Rechercher un extérieur par CIN

```
SELECT ps.cin, ps.nom, ps.prenom, ex.telephone, ex.organisme  
FROM Extérieur ex JOIN Personne ps ON ex.cin = ps.cin  
WHERE ps.cin='A6755432082631042DHQTD1763';
```

--Rechercher un étudiant par nom et prénom

```
SELECT ps.cin, ps.nom, ps.prenom  
FROM Etudiant et JOIN Personne ps ON et.cin = ps.cin  
WHERE ps.nom='BEAUMER' AND ps.prenom='Marie';
```

--Nombre de salle dans chaque bâtiment

```
SELECT Salle.Bat, COUNT(code)  
FROM Salle  
GROUP BY Salle.Bat;
```

--Tous les spectacles de type Concert

```
SELECT s.nom, s.duree, s.asso_organisatrice, s.type_sp->>'compositeur', s.type_sp->>'annee',  
s.type_sp->>'genre'  
FROM Spectacle s  
WHERE s.type_sp->>'type'='Concert';
```

--Tous les spectacles de type StandUp

```
SELECT s.nom, s.duree, s.asso_organisatrice, s.type_sp->>'genrer'  
FROM Spectacle s  
WHERE s.type_sp->>'type'='StandUp';
```

--Tous les spectacles de type PieceTheatre

```
SELECT s.nom, s.duree, s.asso_organisatrice, s.type_sp->>'auteur', s.type_sp->>'annee',  
s.type_sp->>'type'  
FROM Spectacle s  
WHERE s.type_sp->>'type'='PieceTheatre';
```

--Afficher les rôles d'un spectacle

```
SELECT p.cin, p.nom, p.prenom, r.descriptif  
FROM role r, personne p  
WHERE r.cin=p.cin AND r.nom_spec='Maree basse';
```

--Afficher le rôle d'une personne dans un spectacle


```
SELECT descriptif
FROM Role r
WHERE r.CIN='A0382748Z02928375KDJDH19238' AND r.nom_spec='Maree basse';
```

```
--Toutes les séances d'un spectacle
SELECT nom , seance->>'datetime' AS date
FROM Spectacle
WHERE nom='Maree basse';
( à vérifier pour ID sinon ça change le numéro de ligne du tableau)
```

```
--Tous les billets d'une séance
SELECT b.identifiant, b.categorie, c.tarif, b.acheteur
FROM Billet b, CatBillet c
WHERE b.seance->>'ID' AS id ='3' AND b.categorie=c.categorie
ORDER BY identifiant ASC;
```

```
--Nombre de billets vendus et le revenu par catégorie pour une séance donnée
SELECT billet.categorie, count(identifiant) as billets_vendus, sum(tarif)
FROM billet, catBillet WHERE seance->>'ID'='3' AND acheteur IS not null AND
billet.categorie=catBillet.categorie
Group by billet.categorie;
```

```
--Nombre de billets restants par catégorie pour une séance donnée
SELECT billet.categorie, count(identifiant) as billets_non_vendus
FROM billet
WHERE seance->>'ID'='3' AND acheteur IS null
Group by categorie;
```

```
--Les informations concernant un billet
SELECT b.identifiant, b.categorie, c.tarif, b.acheteur, b.seance, s.nom, s.salle, s.date_time
FROM Billet b, CatBillet c, Spectacle s
WHERE b.identifiant=1 AND b.seance->>'ID'=s.seance->>'ID' AND b.categorie=c.categorie;
```

```
--Types de billets qui peuvent encore être achetés pour une séance donnée
SELECT DISTINCT b.categorie, c.tarif
FROM Billet b, CatBillet c
WHERE b.seance->>'ID'='3' AND b.acheteur IS NULL AND b.categorie=c.categorie
```