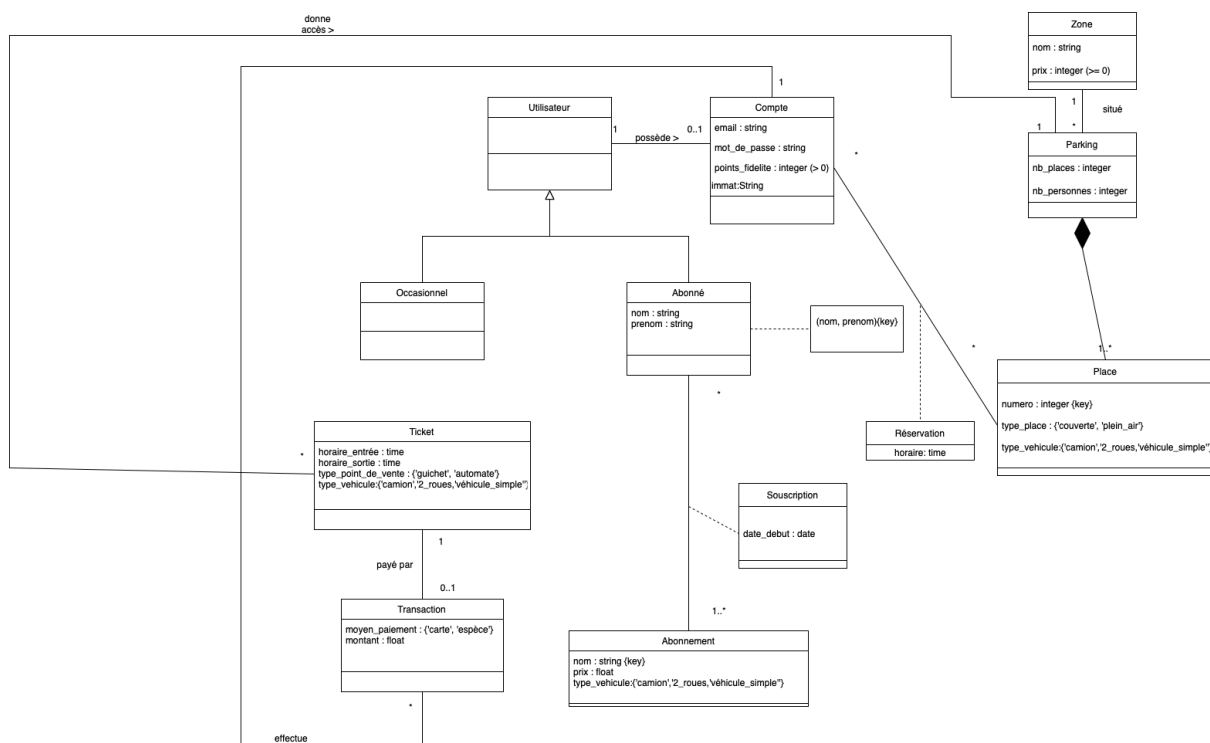


Rapport NF18 R-JSon

1) Modifications apportées à la BDD

Le groupe TD6 a pour but de concevoir une BDD pour la gestion de parkings d'une entreprise dans une ville. Ce rapport présente un résumé non exhaustif de la seconde partie de notre projet destiné aux Bases de Données Non-Relationnelles mais présente les points les plus pertinents réalisés dans cette étude.

Dans un souci de conception, nous avons gardé la structure globale de notre UML que vous pouvez voir ci-dessous.



Néanmoins, des modifications des classes (1) et des nouvelles vues (2) ont été réalisées par l'utilisation du non-relational.

Par exemple, concernant (1), certaines classes sont devenues des attributs JSON d'une autre classe. Nous pouvons citer les classe Abonné, Occasionnel et Abonnement, qui sont devenues plus qu'UN JSON de la classe Compte avec le format de données suivant que l'on nommera Utilisateur:

 $\{$

```

    "nom" : "string", "prenom" : "string", "abonnement" : "abonnement",
    type_utilisateur : 'enum{occas, abonne}'
}

```

L'avantage de ce procédé par rapport à notre étude en relationnel est qu'il n'y plus besoin de faire de jointure pour avoir le nom d'un utilisateur et le type de son abonnement. On a directement accès ou plus facilement accès grâce à des requêtes plus simples au nom d'un utilisateur, son type d'abonnement (car nous avons fusionné occasionnel ou abonné sous une même clé nommée type_utilisateur) et son abonnement s'il en a un.

Voyons via la requête ci-dessous:

```

--liste des Comptes :
SELECT * FROM Compte;

```

Les informations concernant les comptes utilisateur :

idcompte	email	mot_de_passe	pts_de_fidelite	immat	utilisateur
1	y.ouraq@etu.utc.fr	youraq	36	33 SQL 78	{ "nom" : "OURAQ", "prenom" : "yassine", "type_utilisateur" : "abonne", "abonnement" : { "nom" : "vehicul "prix" : 39.99, "type_vehicule" "date_debut" : " } }
2	n.yi@etu.utc.fr	nyi	3	38 EST DF	{ "nom" : "VI", "prenom" : "nicolas", "type_utilisateur" : "abonne",

Dans l'image ci-dessus nous pouvons voir que les valeurs retournées à l'attribut abonnement ne sont pas des valeurs atomiques, il s'agit d'un autre point intéressant. Dans le JSON Utilisateur, l'attribut abonnement est aussi un attribut JSON avec abonnement sous le format de données suivant:

```
[
```

```

        {"nom" : "string", "prix" : "float", "type_vehicule" : "enum_vehicule",
"date_debut" : "date"}
    ]

```

Nous avons donc dû être amenés à manipuler des imbrications.

Exemple :

```

{
  "nom" : "SAOUD",
  "prenom" : "adel",
  "type_utilisateur" : "abonne",
  "abonnement" :
    [
      {
        "nom" : "2_roues",
        "prix" : 19.99,
        "type_vehicule" : "2_roues",
        "date_debut" : "2021-08-03"
      },
      {
        "nom" : "Camion",
        "prix" : 79.99,
        "type_vehicule" : "Camion",
        "date_debut" : "2021-08-01"
      }
    ]
}

```

Ensuite, concernant (2), comme nous avons créé des attributs JSON, nous avons dû recréer des vues par la disparition et fusion de certaines classes. on recrée des vues pour regrouper des informations qui nous seront utiles lors de la mise en place des requêtes.

La vue avec le plus d'informations que nous avons créé est la vue AbonnementAbonne qui renvoie le nom, le prénom, l'identifiant de l'utilisateur, le prix de chacun des abonnements de l'utilisateur et le type d'abonnement (2 roues, véhicule simple ou camion).

```

20  --Vue Abonnements :
21  CREATE VIEW AbonnementAbonne AS (
22      SELECT Compte.utilisateur->>'nom' AS nom,
23      Compte.utilisateur->>'prenom' AS prenom,
24      Compte.idCompte,
25      a->>'nom' AS abonnement,
26      CAST(a->>'prix' AS FLOAT) AS cout,
27      a->>'type_vehicule' AS type_vehicule,
28      a->>'date_debut' AS date_debut
29      FROM Compte, JSON_ARRAY_ELEMENTS(utilisateur->'abonnement') AS a
30  );
31

```

D'autres vues ont été créées comme Abonné et occasionnel mais ne seront pas présentées, elles seront consultables toutefois sur le Gitlab.

On rappellera que la création des vues permet dans un second temps de recréer nos requêtes. Voici deux exemples qui révèlent notre travail et les différences perçues entre notre travail en relationnel et en non relationnel.

Exemple de l'ancienne insertion des données dans Abonne, Abonnement, Souscription :

```
51
52 --Abonnement(#nom: varchar, prix: float, type_vehicule: enumerate{camion, 2_roues,
vehicule_simple})
53 INSERT INTO Abonnement VALUES('Camion', '79.99', 'camion');
54 INSERT INTO Abonnement VALUES('2_roues', '19.99', '2_roues');
55 INSERT INTO Abonnement VALUES('vehicule_simple', '39.99', 'vehicule_simple');
56
57
58
59
60
61 --CREATE TABLE Abonne(compte INTEGER PRIMARY KEY REFERENCES Compte(idCompte), nom
VARCHAR NOT NULL, prenom VARCHAR NOT NULL);
62
63 INSERT INTO Abonne VALUES('1', 'OURAQ', 'yassine');
64 INSERT INTO Abonne VALUES('2', 'YI', 'nicolas');
65 INSERT INTO Abonne VALUES('3', 'SAOUD', 'adel');
66 INSERT INTO Abonne VALUES('4', 'SOLAN', 'matthieu');
67
68 --CREATE TABLE Souscription(abonnement REFERENCES Abonnement(nom), abonne REFERENCES
Abonne(compte), date_debut);
69
70 INSERT INTO Souscription VALUES('Camion', '3', '2021-08-03');
71 INSERT INTO Souscription VALUES('2_roues', '2', '2021-08-01');
72 INSERT INTO Souscription VALUES('vehicule_simple', '1', '2021-08-07');
73 INSERT INTO Souscription VALUES('2_roues', '3', '2021-08-03');
74 INSERT INTO Souscription VALUES('vehicule_simple', '4', '2021-07-01'); -- cet ancien
abonne est devenu un occasionnel sa soucription a expire
```

Exemple de nouvelle insertion des données sous forme de JSON dans Compte :

```

INSERT INTO Compte VALUES (
'1',
'y.ouraq@etu.utc.fr',
'youraq',
'36',
'33 Sql 78',
,
{
    "nom" : "OURAQ",
    "prenom" : "yassine",
    "type_utilisateur" : "abonne",
    "abonnement" :
        [
            {
                "nom" : "vehicule_simple",
                "prix" : 39.99,
                "type_vehicule" : "vehicule_simple",
                "date_debut" : "2021-08-07"
            }
        ]
}
,
);

```

2) Structure des attributs JSON

Format de données JSON pour *abonnement* :

```

[
    {
        "nom" : "string", "prix" : "float", "type_vehicule" : "enum_vehicule",
        "date_debut" : "date"
    }
]

```

Format de données JSON pour *utilisateur* :

```

{
    "nom" : "string", "prenom" : "string", "abonnement" : abonnement,
    "type_utilisateur" : 'enum{occas, abonné}'
}

```

Exemple de JSON inséré dans l'attribut Compte.utilisateur :

```

{
    "nom" : "SAOUD",
    "prenom" : "adel",
    "type_utilisateur" : "abonne",
    "abonnement" :
        [
            {
                "nom" : "2_roues",

```

```
        "prix" : 19.99,  
        "type_vehicule" : "2_roues",  
        "date_debut" : "2021-08-03"  
    },  
    {  
        "nom" : "Camion",  
        "prix" : 79.99,  
        "type_vehicule" : "Camion",  
        "date_debut" : "2021-08-01"  
    }  
]  
}
```