

Compte rendu projet Base de Données 2

BILLAUD MAËL

JOHAN QUENTIN
GOUBON VALENTIN

RAMÉ TRISTAN

Février 2022

1 Première partie

Présentation Générale

Dans ce projet, l'objectif est de réaliser une base de données pas à pas. Pour ce faire, nous avons décidé de représenter la base de données d'un service de livraison de nourriture (Naofood, Deliveroo, ...). Nous avons donc défini ci-dessous un ensemble d'attributs permettant la gestion d'une telle organisation :

$U = \{nom_resto, type_cuisine, adr_resto, tel_resto, ouvert_dimanche, note_resto, nom_client, prenom_client, tel_client, adr_client, nom_menu, prix_menu, nom_coursier, prenom_coursier, note_coursier, date_course, mode_deplacement\}$.

Table générale

Pour des soucis de lisibilités, nous vous joignons notre table initiale, contenant tous les attributs ainsi qu'une dizaine de tuples, sous la forme d'un lien. Celui-ci vous mènera sur notre [Google Sheet](#).

Dépendances fonctionnelles

Vous trouverez ci-dessous les différentes dépendances fonctionnelles que nous avons trouvées pour notre table initiale.

- (1) $adr_resto \rightarrow nom_resto, tel_resto, ouvert_dimanche$
- (2) $nom_client, prenom_client \rightarrow adr_client, tel_client$
- (3) $nom_menu \rightarrow prix_menu, nom_resto$
- (4) $nom_coursier, prenom_coursier \rightarrow mode_deplacement$
- (5) $nom_coursier, prenom_coursier, date_course, nom_menu \rightarrow note_coursier, note_resto$
- (6) $nom_resto \rightarrow type_cuisine$

En calculant la clé à partir des DFs définies plus haut, on obtient :

clé = $\{adr_resto, nom_client, prenom_client, nom_menu, nom_coursier, prenom_coursier, date_course\}$

Algorithmes de normalisation

Passage en 3FN, Algorithme de BERNSTEIN

Pour appliquer l'algorithme suivant, on utilise la clé telle que définie plus haut. De manière évidente et en comparant avec les DFs, $\langle R(U), DF \rangle$ n'est pas en 3FN, on partitionne donc notre schéma en plusieurs ensembles :

- $DF_1 = \{adr_resto \rightarrow nom_resto, tel_resto, ouvert_dimanche\}$
et $U_1 = \{adr_resto, nom_resto, tel_resto, ouvert_dimanche\}$
- $DF_2 = \{nom_client, prenom_client \rightarrow adr_client, tel_client\}$
et $U_2 = \{nom_client, prenom_client, adr_client, tel_client\}$
- $DF_3 = \{nom_menu \rightarrow prix_menu, nom_resto\}$
et $U_3 = \{nom_menu, prix_menu, nom_resto\}$
- $DF_4 = \{nom_coursier, prenom_coursier \rightarrow mode_deplacement\}$
et $U_4 = \{nom_coursier, prenom_coursier, mode_deplacement\}$
- $DF_5 = \{nom_coursier, prenom_coursier, date_course, nom_menu \rightarrow note_coursier, note_resto\}$
et $U_5 = \{nom_coursier, prenom_coursier, date_course, nom_menu, note_coursier, note_resto\}$
- $DF_6 = \{nom_resto \rightarrow type_cuisine\}$ et $U_6 = \{nom_resto, type_cuisine\}$

On constate qu'aucun des schémas obtenus par le partitionnement ne contient de clé de R, on ajoute donc le schéma suivant :

$DF_7 = \{\}$ et $U_7 = \{adr_resto, nom_client, prenom_client, nom_menu, nom_coursier, prenom_coursier, date_course\}$.

De plus, nous n'avons pas perdu d'information ni de dépendances fonctionnelles. Cette normalisation est donc **spi** et **spdf** où toutes les relations sont en **FN3**.

Passage en FNBC, Algorithme de Décomposition

L'algorithme que nous allons utiliser ci-dessous se base sur une décomposition récursive de Df ne contenant pas de clé à gauche. Celle-ci donne à chaque fois deux sous-relations qu'on décompose à nouveau si elles ne sont pas en FNBC.

Par soucis de lisibilité, nous allons désigner chacun des attributs par des lettres pour éviter que notre arbre de décomposition ne soit trop conséquent :

A = nom_resto	
B = type_cuisine	
C = adr_resto	
D = tel_resto	
E = ouvert_dimanche	
F = note_resto	
G = nom_client	(1) C → A,D,F
H = prenom_client	(2) G,H → I,J
I = tel_client	(3) K → L,A
J = adr_client	(4) M,N → Q
K = nom_menu	(5) M,N,P,K → O,F
L = prix_menu	(6) A → B
M = nom_coursier	
N = prenom_coursier	
O = note_coursier	
P = date_course	
Q = mode_deplacement	

La représentation de l'arbre de décomposition se trouve sur la page suivante :

$$U = \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q\}$$

$$DF = \{1, 2, 3, 4, 5, 6\}$$

$$U_1 = \{C, A, D, E\}$$

$$DF_1 = \{1\}$$

$$U_2 = \{B, C, F, G, H, I, J, K, L, M, O, P, Q\}$$

$$DF_2 = \{2, 4, 5\}$$

$$U_{21} = \{G, H, I, J\}$$

$$DF_{21} = \{2\}$$

$$U_{22} = \{B, C, F, G, H, K, L, M, N, O, P, Q\}$$

$$DF_{22} = \{4, 5\}$$

$$U_{221} = \{M, N, P, K, O, F\}$$

$$DF_{221} = \{5\}$$

$$U_{222} = \{B, C, G, H, K, L, M, N, P, Q\}$$

$$DF_{222} = \{4\}$$

$$U_{2221} = \{M, N, Q\}$$

$$DF_{2221} = \{4\}$$

$$U_{2222} = \{B, C, G, H, K, L, M, N, P\}$$

$$DF_{2222} = \{\}$$

Comme nous avons pu le voir avec le schéma ci-dessus, l'algorithme de normalisation par décomposition nous a donné 5 schémas.

- $DF_1 = \{1\}$ et $U_1 = \{adr_resto, nom_resto, tel_resto, ouvert_dimanche\}$
- $DF_{21} = \{2\}$ et $U_{21} = \{nom_client, prenom_client, tel_client, adr_client\}$
- $DF_{221} = \{5\}$ et $U_{221} = \{nom_coursier, prenom_coursier, date_course, nom_menu, note_coursier, note_resto\}$
- $DF_{2221} = \{4\}$ et $U_{2221} = \{nom_coursier, prenom_coursier, mode_deplacement\}$
- $DF_{2222} = \{\}$ et $U_{2222} = \{type_cuisine, adr_resto, nom_client, prenom_client, nom_menu, prix_menu, nom_coursier, prenom_coursier, date_course\}$

On peut remarquer que nous avons perdu 2 dépendances fonctionnelles au cours de la décomposition. Cette dernière est donc **uniquement spi** où toutes les relations sont en **FNBC**.

Modèle Relationnel de la Base de Données

Grâce à nos algorithmes de normalisation, nous avons pu trouver deux schémas différents. Nous n'allons maintenant essayer de déterminer lequel des deux schémas obtenu nous donne le résultat le plus cohérents (sens des relations) et le plus qualitatif (formes normales).

D'un côté, un schéma trouvé par l'algorithme de Bernstein est en **3^{ème} Forme Normale** alors que celui obtenu par l'algorithme de décomposition est en **Forme Normale de Boyce-Codd**. Ce dernier étant un plus grand gage de qualité que le 3FN, nous allons donc le préférer.

Cependant, lors que l'on regarde le schéma obtenu avec notre algorithme de décomposition, ce dernier n'est pas vraiment cohérent. En effet, l'attribut *type de cuisine* est dans la relation qui contient les informations relatives à une commande alors qu'il serait plus logique qu'elle se rattache à celle correspondant au restaurant.

Nous allons donc essayer d'appliquer à nouveau l'algorithme de décomposition sur notre schéma initial mais avec des choix différents pour trouver un résultat plus satisfaisant. Dans cette décomposition, nous garderons les notations utilisées précédemment.

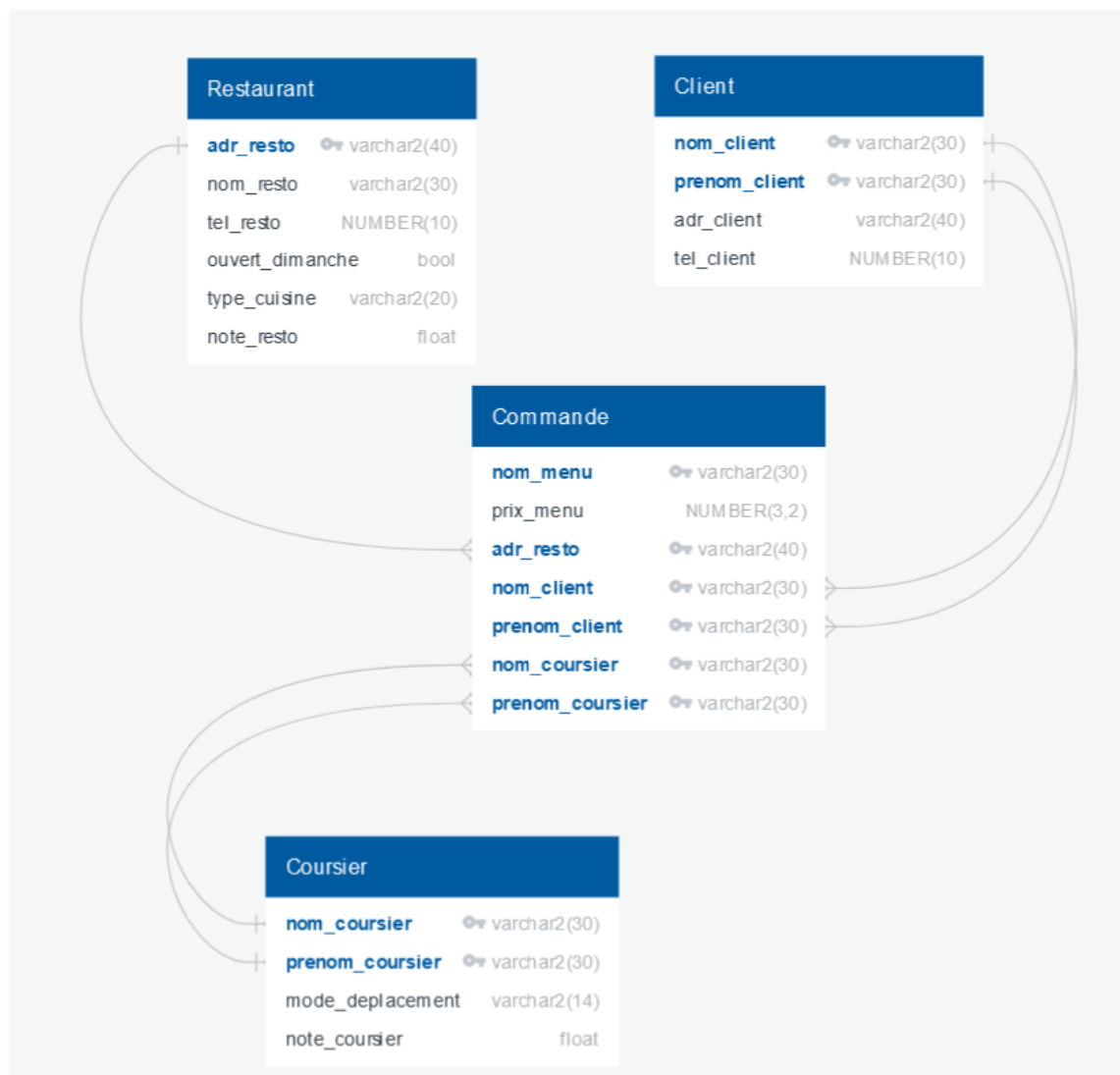


FIGURE 1 – Modèle Relationnel de la Base de Données *Service de Livraison de Nourriture*

Suite du projet
