Intégration des dépendances fonctionnelles dans la définition de schéma des graphes de propriétés

Maude Manouvrier[1], Khalid Belhajjame[1]

1. Université Paris-Dauphine, PSL Research University
CNRS UMR [7243] LAMSADE
Place du Maréchal de Lattre de Tassigny 75775 Paris cedex 16, France
prenom.nom@dauphine.fr

REFERENCE DE L'ARTICLE INTERNATIONAL
Cet article est une synthèse de l'article : Manouvrier, M., Belhajjame, K. (2024). PG-FD: Mapping Functional Dependencies to the Future Property Graph Schema Standard. In Advances in Databases and Information Systems. ADBIS 2024. LNCS vol 14918. Springer, Cham, pp. 45-59. https://doi.org/10.1007/978-3-031-70626-4_4

Un graphe de propriétés est composé de nœuds et d'arêtes, associés à une étiquette et décrits par un ensemble de propriétés ou d'attributs, généralement représentés par des couples clé-valeur. Les graphes de propriétés sont largement utilisés dans de nombreux domaines. En avril 2024, le langage GQL, pour *Graph Query Language* (cf. Francis et al., 2023), a été officiellement publié en tant que norme ISO/CEI pour interroger les bases de données graphes. Angles et al. (2023) ont proposé d'étendre la norme GQL en définissant un formalisme, nommé *PG-Schema,* pour spécifier les schémas des graphes de propriétés, au sens classique du schéma dans les bases de données relationnelles (Barret et al., 2024). *PG-Schema* permet de définir des types de nœuds et d'arêtes, ainsi que des contraintes d'intégrité.

Les dépendances fonctionnelles (cf. Codd, 1972) sont des contraintes d'intégrité particulières. En base de données relationnelles, une dépendance fonctionnelle (DF) se définit par $X \rightarrow Y$, où X et Y sont des ensembles d'attributs d'un schéma R, et signifie que pour toutes les instances r de schéma R et pour tous les nuplets t_1 et t_2 de r, si t_1 et t_2 ont la même valeur pour X alors ils ont la même valeur pour Y. Les DFs sont notamment utilisées pour déterminer les identificateurs ou clés, contrôler la cohérence des données, optimiser les requêtes ou nettoyer des données. Plusieurs approches ont défini des dépendances fonctionnelles pour les bases de données graphes. Parmi ces approches, on peut citer GED (*Graph Entity Dependency*) de Fan et Lu (2019), gFD (*Graph-tailored functional dependency*) de Skavantzos et Link

(2023) et GD (*Graph Dependency*) de Zheng et al. (2023). GED et GD se basent sur des motifs de graphe (*Graph Pattern*), c'est-à-dire des sous-graphes orientés, dont les nœuds sont associés à des variables, permettant de spécifier le champ d'application de la DF. L'approche gFD se base, quant à elle, sur des conditions d'existence. Ces approches offrent une base théorique pour définir ou vérifier les DFs dans les bases de données graphes mais ne définissent pas de langage pour exprimer ces dépendances. L'objectif de notre proposition est de pallier ce manque et de spécifier comment traduire ces dépendances dans la future norme de définition de schéma de graphes, *PG-Schema*, de Angles et al. (2023).

Notre article présente une double contribution. Premièrement, il fournit une synthèse des approches de la littérature définissant des dépendances fonctionnelles pour les graphes, en soulignant notamment comment ces propositions sont liées les unes aux autres. Il présente également des règles de traduction des DFs pour les graphes en *PG-Schema* et démontre que ces règles respectent trois propriétés importantes, à savoir la calculabilité, la préservation de l'information et la préservation de la sémantique. Ces règles de correspondance sont mises en œuvre dans un prototype développé en Python et nommé PG-FD. Notre approche est, à notre connaissance, la première solution capable de transformer les dépendances fonctionnelles de graphes dans le standard *PG-Schema,* tout en préservant leur sémantique. Pour la suite de nos travaux, nous souhaitons mener des expérimentations plus poussées de notre prototype en utilisant des graphes réels et prendre en compte d'autres types de contraintes dans les graphes de propriétés.

Bibliographie

Angles, R., Bonifati, A., Dumbrava, S., *et al.* (2023). PG-Schema: Schemas for property graphs, *ACM on Management of Data*, vol. 1, n° 2, p. 1-25.

Barret, N., Enache, T., Manolescu, I., *et al.* (2024). Finding the PG schema of any (semi) structured dataset: a tale of graphs and abstraction. In *Proceedings of IEEE 40th International Conf. on Data Engineering Workshops (ICDEW)*, *Utrecht, Netherlands*, pp. 365-369.

Codd, E.F (1972). Further normalization of the data base relational model. *Data base systems*, vol. 6, p. 33–64

Fan, W., Lu, P. (2019). Dependencies for graphs. *ACM Transactions on Database Systems (TODS)*, vol. 44, n° 2, p. 1-40.

Francis, N., Gheerbrant, A., Guagliardo, P., *et al*. (2023). A Researcher's Digest of GQL. In *Proceedings of 6th International Conf. on Database Theory (ICDT), Ioannina, Greece*. pp. 1:1-1:2.

Skavantzos, P., Link, S. (2023). Normalizing Property Graphs. *VLDB Endowment*, vol. 16, n° 11, p.3031–3043.

Zheng, X., Dasgupta, S., Gupta, A. (2023). P2KG: Declarative construction and quality evaluation of knowledge graph from polystores. In: *Proceedings of the 27th European Conf. on Advances in Databases and Information Systems (ADBIS), Barcelona, Spain,* pp. 427-439.