Sytnyk N.V., Candidate of Economic Sciences, Professor, Kyiv National Economic University named after Vadym Hetman,
Zinovieva I.S., Candidate of Economic Sciences, Associate Professor, Kyiv National Economic University named after Vadym Hetman,
Denisova O.O., Candidate of Economic Sciences, Associate Professor, Doctoral Candidate, Kyiv National Economic University named after Vadym Hetman

MODELLING AND USE OF GRAPH DATABASES IN INFORMATION CONTROL SYSTEMS

One of the goals of the digital transformation initiative by the Ministry of Education and Science of Ukraine is to optimize and automate management processes within the education sector and higher education institutions (HEIs), specifically concerning the accreditation of educational programs [1, 2].

The development of educational programs for HEIs is crucial, as these programs lay the foundations for the formation of future competences and program learning outcomes in the specialty for which they are developed. Under the Law of Ukraine «On Higher Education», an educational program consists of educational components logically sequenced for study, assigned ECTS credits, and designed to achieve specific learning outcomes, which a student must master to earn a qualification [3].

The analysis of domestic and foreign works on the subject of the study showed that there are a significant number of publications and scientific reviews on improving the quality of education in the higher education system, including through the introduction of a competence-based approach [4]. However, most of these studies are generally recommendatory and lack practical guidance on the tools, methods, and ways to compare the nomenclature of competencies and learning outcomes listed in educational programs against the actual competencies developed in future specialists.

Paper [5] examines the impact of higher education standardisation on the quality of the educational process, in particular, the «contradictions of quality control», among which the author highlights the practice of assessing learning outcomes not in the workplace, but in the classroom (i.e. at the theoretical level), which can lead to a lack of correlation between declarative competencies and real learning outcomes. The publication attempts to reflect the relationship between the regulatory documents in the field of higher education standardisation and the quality of the education system through the educational qualification characteristics of a future specialist and his/her educational training.

The monographic study [6] is devoted to the problem of improving the quality of vocational education in Ukraine, and it is proposed to use «passports of professions» for each speciality, which will contain a list of competences desired by the employer. In the development of such a portrait, considerable attention is paid to the modelling method, which can provide optimal prerequisites for the formation of competitive specialists with a high level of knowledge and skills required by employers.

Publication [7] is devoted to the issue of identifying a set of professional competences of future specialists in the speciality 122 «Computer Science». The authors note that the current educational standard for first (bachelor's) level applicants in this speciality has an important drawback – it was developed by specialists of the Ministry of Education and Science of Ukraine without the involvement of employers. In order to address the issue of improving the quality of education in the higher education system, the authors propose to improve the existing standards of training by involving representatives of the IT sector and employers in the development of professional standards. The paper also emphasises the need for annual review of curricula and discipline materials to bring them as close as possible to the needs of the IT industry and the labour market, as noted in other sources [8]. This raises the problem of the need to revise and update educational programs and, in particular, a set of competences and learning outcomes to bring them in line with market needs, employers' demands, innovative technologies, etc. For example, in the EU countries, a special classification of skills, competences, qualifications and occupations (European Skills, Competences, Qualifications and Occupations or ESCO) is developed and annually updated, which contains, among other things, a list of qualification requirements that a specialist of a certain speciality in the EU countries must meet [9]. Implementation of this classification in Ukraine, according to the authors of the publication, will require not only appropriate changes to the standards and educational programs for training highly qualified specialists, but also their constant updating and revision. The review shows that the publications under consideration prove the relevance and timeliness of the research topic, but, on the other hand, only partially characterise this issue.

Today, the majority of educational programs in HEIs are developed manually by specialists who are members of expert groups, which, according to the authors of the study, emphasises the need to digitalise the processes of preparing it using modern information technologies.

One of the priority tasks of digitalisation is to create information control systems that simplify and optimise the process of creating educational programs. An important step in this direction is information modelling, which is the initial stage of designing automated solutions and involves studying the subject area, defining and designing data structures in accordance with the rules for building the selected data model. This modelling not only allows for efficient data organization but also deepens understanding of the subject area, identifying needs and opportunities for optimization. It helps to identify the key elements of educational programs, their interrelationships and the dynamics of change, which allows for the development of more efficient automated solutions. When modelling the formation of HEI educational program, it is important to take into account the following special characteristics of the process: the presence of textual and poorly structured data, a large number of links between educational components that need to be modified and/or rethought at the stage of formation. Given these features, the modelling issue covers the choice of a data representation model and an appropriate database management system (DBMS) to create a database of an information control system with educational programs development functions, which, among other things, will create an effective tool for analysing possible scenarios for optimising or developing educational programs and will support their rapid adaptation to changes in the requirements of the labour market and the technological environment.

This **study aims** to explore opportunities to optimize and enhance the development of educational and professional programs (EPPs) by automating the process, identifying its main informational components and characteristics, and establishing rules for information modeling within a graph DBMS environment to generally improve the efficiency and quality of EPPs.

Theoretical foundations of the study

The basis for the development of educational programs in HEIs is the current regulatory framework in this area. Based on the legislation and the Higher Education Standard (Standard) [10], each HEI develops its own EPPs and corresponding to them curricula to prepare professionals in the required field.

The standard structure of the EPP at Kyiv National Economic University named after Vadym Hetman includes the following sections:

1) profile of the educational program, for example in the speciality 122 «Computer Science»;

2) a list of the EPP components and their logical sequence (includes: a list of educational components, EPP structure, EPP structural and logical diagram);

3) form of certification of higher education students;

4) matrixes: compliance of the competences defined by the educational program with the compulsory components, ensuring the learning outcomes with the compulsory components of the educational program, compliance of the competences/learning outcomes defined by the Standard with the descriptors of the National Qualifications Framework, compliance of the learning outcomes and competences defined by the Standard;

5) additions.

Based on the EPP, a curriculum is developed that contains a schedule of the educational process, a list and scope of educational components, including[1]:

– a block of compulsory (normative) academic disciplines divided into cycles:
  - general training;
  - professional training;

– a block of elective courses;

– practical training block (internship, computer training, preparation of a qualifying master's thesis).

On the basis of the developed EPP and curricula, the academic staff of the departments prepare work programs and methodological materials for academic disciplines and courseworks, other methodological documentation. The work programs demonstrate how the applicant for the relevant degree of higher education will achieve the learning outcomes declared in the EPP.

As a result of educational activities, competences are formed, which, according to the legislation, are usually divided into: integral, general and special (professional). The list of these competences for the specialty 122 «Computer Science» is also defined by the Standard [10] and may be supplemented by the decision of the project team involved in the development of the EPP.

Modelling of the described process requires, first of all, the reflection of the interrelationships of the EPP, the list of educational

[1] *It is considered on the example of the EPP «Information Control Systems and Technologies« of Kyiv National Economic University named after Vadym Hetman. Department of Information Systems in Economics, 2023.*

components and the competences formed by these components. Among the most significant problems is the complexity and non-triviality of the comparison process in matters related to the presentation of unstructured and/or poorly structured data. Thus, there are issues of determining an adequate data model that can be used to work with the information contained in the EPP and selecting an acceptable DBMS.

Until recently, the development of information systems was dominated by the use of relational databases or RDBMS. Over time, the focus has shifted to the use of non-relational databases (NoSQL) [11]. A study of the subject area shows that when designing an information control system with educational programs development functions, the use of RDBMS has significant limitations:

– they are not well suited for storing poorly structured and unstructured data. A significant amount of data describing the processes described above has a many-to-many («M:N») relationship. From the point of view of design methodology and normalisation theory [12], to implement this kind of relationship, additional tables need to be created that will not have a semantic interpretation in the real terms and concepts of the process of forming the EPP;

– support a static diagram (scheme), which is not desirable to change after its design, while the content of the EPP and the links between educational components can change quite often;

– do not have visualisation tools that would allow them to view and analyse data and the links between them;

– are difficult to scale horizontally.

Based on the above considerations, a study was conducted of non-relational databases, which were created as opposed to RDBMS to provide the ability to support unstructured or poorly structured data, horizontal scaling and work with distributed clusters in the web space. As we know, the main data models used in NoSQL include: key-value, document-oriented, column-oriented and graph-oriented [11, 13, 14].

Comparative characteristics of relational, graph, and other NoSQL models are shown in Table 1. It should be noted that graph data representation models are highlighted in a separate column in the table because they have their own specifics and features relative to other NoSQLs, and some of their characteristics are similar to RDBMS.

A comparative analysis of existing data models has led to the conclusion that the graph model is the most suitable for the subject area under study, since they allow modelling numerous relationships between entities, which is typical for the EPP, and storing, viewing and analysing both structured and unstructured data.

The conclusions obtained are confirmed by the results of studies by other researchers. For example, study [15] substantiates the idea of the expediency of working with graph models of data representation when identifying qualification frameworks in the assessment of various EPPs for their compliance with existing standards and desired sets of competencies. The authors prove that the use of a graph data model allows to improve the quality of perception of a large array of poorly structured data by means of visualisation, perform in-depth analysis of EPPs and form the basis for their further improvement.

According to modern approaches to the classification of graph databases, they are divided into the main types: property graphs and triples (RDF graphs) [16]. On the one hand, these two main types of graphs help to explore and graphically represent related data, on the other hand, the ways of displaying the relationships between data in these types are completely different. For example, triples are used to represent knowledge in the form of triples with the structure <subject, predicate, object>. Each triplet is a form of declarative knowledge representation in which links are built between subjects and objects, which are the nodes of the graph, and predicates are its edges [17]. They are overwhelmingly used on the Semantic Web. The RDF model is well suited for representing ontologies and knowledge bases, so it is very common in artificial intelligence systems and is often used to process the results of scientific research.

Table 1

COMPARATIVE CHARACTERISTICS OF RELATIONAL AND NON-RELATIONAL DATABASES

#	Features	Models of modern databases		
		Relational models	NoSQL models	
			Key/value, column-oriented, document-oriented	Graph
1	*2*	*3*	*4*	*5*
1	Theoretical basis	Normalisation theory, relational algebra	CAP theorem	CAP theorem
2	Claims and transaction support	Support for transactions and ACID requirements	Support for BASE requirements	Support for transactions and ACID requirements

1	*2*	*3*	*4*	*5*
3	Support for the SQL query language	Supports	Does not support	Declarative languages similar to SQL, for example, the Cyrher language in Neo4j DBMS and SPARQL in RDF graphs
4	Support for scaling	Vertical scaling	Horizontal scaling	Horizontal scaling
5	Support for a static data chart (diagram)	Supports	Does not support	Interconnections in the form of a graph, the structure of which is adaptive, i.e. it can change as needed
6	Processing of atomic and aggregated data	Atomic data (aggregate-ignore)	Aggregated data of any structure	Atomic data (aggregate-ignore)
7	Built-in visualisation tools	Absent	Absent	Available (in the absence of built-in tools, can use Gephi)[2]
8	Data normalisation	Structured normalised data with guaranteed consistency	Unstructured, denormalised, poorly consistent data	Normalisation is optional, the database can be normalised or denormalised under certain conditions

Source: developed by the authors

Graphs with properties allow to describe any entities in the subject area and the relationships that exist between them [18]. In a generalised form, a graph model with properties can be represented as follows:

$$G = (N, E), \tag{1}$$

where N – is a finite set of nodes (or vertices, nodes), $N = \{v_1, v_2, \dots, v_m\}$; E – is a set of orientated arcs (or edges),

[2] *Gephi is a graph visualisation software [19].*

$E = \{e_1, e_2, \dots, e_n\}$ each of which represents a connection between different pairs of nodes (v_i, v_j) [20].

Each node of the property graph is described as:

$$N = \{(id, [L_1, \dots, L_h\ ], [(k_1 v_1), \dots, (\ k_m v_m)])\} \tag{2}$$

where id – is the unique identifier of the node, L_i – node label, $(k_j v_j)$ – key-value pairs that characterise the properties of node, while k_j – key identifying the name of the attribute property, v_j – value of the property [16].

Each edge belongs to a certain type (TypeEdges), which characterises the semantics of the connection and necessarily contains the initial (ParentNode) and final (ChildNode) nodes. The structured representation of a node is as follows:

$$E = \{(ParentNode) - \left[s{:}\,TypeEdges\begin{pmatrix} [k_1 v_1], \\ \dots, [k_m v_m] \end{pmatrix}\right] \rightarrow (ChildNode)\} \tag{3}$$

where s – is an edge variable, $(k_j v_j)$ – key-value pairs that characterise the properties of the edge and are optional components; the arrow symbol (→) indicates the orientation of the connection. Arrows between two nodes can be forward (→) or backwards (←) direction. If necessary, two edges of different orientations can be drawn between two entities. When the direction of the connection is difficult to determine or is not important, then the construction $(Node1) - [edges] - (Node2)$.

The limitation of using graph databases based on the RDF format is the difficulty of building N-ary relationships between graph nodes. Therefore, as a rule, property graphs are best suited for heterogeneous data with a large number of attributes and complex graph operations, while triplets are best suited for homogeneous data with a small number of attributes and simple semantic analysis. Therefore, triplets are not suitable for modelling data that characterise the EPP, as it is difficult and impossible to describe all entities and relationships between them in the form of RDF graphs.

The search and analysis of the option for creating and storing data describing the EPP is narrowed down to the choice of a graph DBMS that supports a graph with properties. The authors have identified the following criteria for such a choice:

– DBMS rating, which characterises its current popularity and use in practice;

– availability of graph visualisation tools that will allow viewing and analysing the graph;

– horizontal scaling;
– availability of a query language;
– support for ACID requirements (atomicity, consistency, isolation, reliability) and transactions;
– support for GRUD operations (Great, Read, Update, Delete);
– availability of a library that implements typical operations with graphs;
– the ability to work offline or in the cloud;
– support for the most common programming languages;
– availability of an open source version that does not require a licence, which will reduce additional costs.

As noted in [8], most HEIs, due to a number of objective factors, are unable to fully provide a high level of material and technical base of computer classes and research laboratories. Additional costs for proprietary software are highly undesirable.

Among the graph databases, the number of which is currently 41, the most fully meets the above requirements DBMS Neo4j, which, according to DB-Engines [21], ranks first in the ranking in terms of the level of prevalence of use (see Fig. 1).

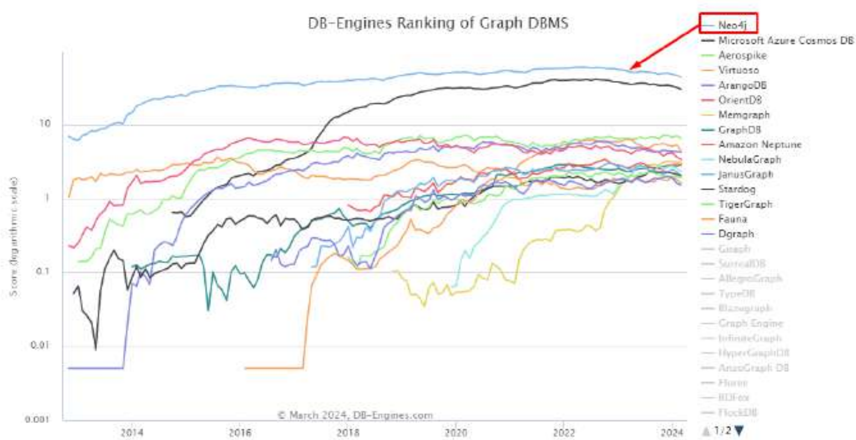

Figure 1. Graph DBMS rating (data as of January 2024)

Source: [21]

A review of the literature shows that a significant number of scientists are inclined to believe that in modelling complex processes of activity, it is worth focusing on the Neo4j DBMS [22–24]. In particular, work [25] substantiates the feasibility of using Neo4j as an accessible and convenient tool for visualising data generated as a result of developing a set of competencies required for healthcare professionals. In [26], it is proposed to use Neo4j to solve complex management problems based on the need for a deep understanding of the semantic meaning of textual information.

The Neo4j DBMS is:

– open source software;

– focused on working with graphs with properties;

– compatible with operating systems: Linux, OS X, Solaris and Windows;

– supports programming languages such as .Net, Clojure, Elixir, Go, Groovy, Haskell, Java, JavaScript, Perl, PHP, Python, Ruby, Scala;

– can run on a standalone basis or in the cloud under the Neo4j AuraDB cloud service;

– has a SQL-like query language Cypher Query Language (Cypher);

– has web browser-based visualisation tools [27].

The Neo4j database model is an oriented graph containing a set:

– nodes, which represent the vertices of the graph and characterise certain objects (entities) of the subject area;

– node labels that characterise the node type and semantics;

– relations or edges that reflect the connections between the nodes of the graph;

– properties – specific data described by key-value pairs and characterising the nodes and edges of the graph.

The key advantage of graph databases is the speed of traversing a large number of nodes and finding the necessary information. The Neo4j performs fast navigation through the graph nodes and has indexing tools that speed up the search for nodes and edges by certain properties or determining the starting node for traversing the graph.

Building a graph and designing a graph database that will be used in the operation of an information control system with the functions of forming educational programs involves identifying entities, their attributes, and the relationships between them, which is what information modelling is actually for.

The modelling process includes:

– defining the types of nodes that represent the entities of a given subject area;

– defining properties of nodes that describe their characteristics;

– defining the edges that describe the connections between nodes;

– defining link rules that limit the permissible connections between nodes.

The importance of information modelling in the development of a graph database is to ensure a clear understanding of the data structure, its consistency and integrity, reduce duplication, and facilitate the effective use of graphs for analytics and visualisation. Thus, modelling data using a graph model is the process of creating an abstract representation of this data in a graph.

The general issues of organising the modelling of graph data models are studied in the scientific and practical literature [11, 16, 28], which demonstrates a significant interest in the development and implementation of graph technologies in information control systems. For example, paper [29] discusses methods and approaches to modelling in graph databases, identifies the advantages and disadvantages of working with graph models. Another paper [17] presents a methodology for modelling one-dimensional and multidimensional data using graphs with properties, which includes approaches to modelling events, behaviour, relationships, aggregation of events and behaviour. The proposed methodology is focused on representing and analysing events in the form of graphs, which will help to increase the efficiency of processing and analysing complex interactions. Work [30] deals with the normalisation of graphs with properties as one of the stages of the modelling process aimed at optimising and improving the performance of software products that use graph models to process large data sets. Papers [31, 32] contain recommendations for representing the data structure, methods of their optimisation, and representation in graph models, in particular, in the Neo4j DBMS environment. The authors note that a deep understanding of the data structure represented in graphs contributes to the effective solution of data analysis tasks. The basic principles of data modelling, recommendations and examples of creating effective graph models are provided in the official Neo4j documentation [33].

Thus, the problem of data modelling in graph databases is relevant and has a wide range of applications in information control systems. The presented research describes the general rules and principles of data modelling, identifies possible problems and potential of using graph databases. In particular, it is established that:

– data modelling in graph databases should be carried out in stages (analysis of the subject area, identification of entities and relationships, selection of a graph data model, model development and implementation);

– for modelling complex data structures with multiple relationships, it is advisable to use graphs with properties, and for application implementation – the Neo4j;

– data presented in graph models can and should be normalised.

Thus, the process of data modelling using graph models involves creating an abstract representation of information objects in the form of a graph. In this context, it is important to present a unified approach to data modelling in graph databases, illustrate the process with examples, provide recommendations and tips on the peculiarities of modelling graphs with properties in the Neo4j environment.

Based on the **results of research**, the authors propose their own approach to information modelling of the EPP data using graphs with properties. The modelling process should include the following steps:

1) studying the subject area, research of data specifics, their structure and semantics;

2) defining the type of data representation model, selecting the type of graph model;

3) modelling of information objects:

3.1) selection of a given subject area entities, modelling of properties, labels;

3.2) data normalisation;

3.3) determining the orientation and building relationships (edges) between entities.

Above, an example of the analysis of the subject area – the process of forming the EPP for HEIs and the justification of the feasibility of choosing a graph data model on the example of a graph with properties for modelling the data of such a subject area was given. Now let's consider the process of modelling information objects on the example of developing a graph database for the EPP «Information Control Systems and Technologies» of the second (master's) level of higher education.

To identify the entities or nodes of a graph, it is recommended to describe the subject area under study in simple narrative sentences that characterise it [34]. The authors recommend following the following rule: each such sentence should contain nouns and verbs (nouns will be comparable to entities, and verbs – to relations between them). Thus, for the chosen subject area, such basic sentences would be the following:

– The specialty provides the acquisition of Competencies;
– Competencies form the Specialty;
– Competencies are revealed by the Discipline;
– The discipline provides Learning Outcomes.

Thus, the main entities of the selected subject area are: *Specialty*, *Competencies*, *Discipline* and *Learning_outcomes*.

Having determined the list of the main objects of the graph, proceed to modelling the properties and labels of the entities. Properties are specific data values that are represented in the key-value format and characterise each individual entity, labels are named elements of a graph node that are used when grouping them into sets.

The generalised structure of the syntax for creating an entity in the Neo4j is as follows:

$$\begin{gathered} \textit{CREATE (Node variables: } label_1, \ldots, \\ label_n \; \{Key_1{:}\,Value_1, \ldots, Key_m{:}\,Value_m\}), \end{gathered} \tag{4}$$

where *Node variables* – an entity variable. It can be entered when creating an entity or later, through queries when accessing nodes. Variables can be designated with separate letters, as it is done in programming, or they can be named with names that reflect their semantics;

label – label of the entity (node). Since a node in a graph can participate in various relationships, for semantic identification, each node is assigned labels that will allow us to group and select certain sets of entities depending on their semantics and purpose. Labels can be used to group nodes by their purpose;

Key:Value – properties of an entity represented as key-value pairs. In the Neo4j, keys are represented as a textual data type – string, values can be represented by an arbitrary data type or an array.

When creating labels, the authors of the study recommend following the following rules:

1) an entity can have one, several, or no labels. Since labels help to distinguish entities, it is better that each node has at least one label, since a node without a label makes it difficult to search and read, which can reduce database performance. But a large number of labels is also not desirable, as it complicates the work with entities and affects the performance of queries. The recommended number of labels for one entity is from 1 to 4. In Neo4j, when creating a node, 4 slots are reserved for labels by default;

2) labels can be orthogonal, i.e. mutually opposite;

3) when modelling labels, we need to take into account the semantics of the subject area and possible queries to the graph. That is, before creating a label, it is advisable to pre-model the queries in which this label will participate;

4) accessing labels in the Neo4j is case-sensitive, while the Cypher language does not generate an error if the name is specified in the wrong case. This aspect requires attention from developers and future users of the graph database.

Let us consider an example of modelling the essence of «speciality». According to the authors of the study, it is important to distinguish for which level of higher education it is intended when developing the EPP for a particular speciality, since the same speciality, for example, 122 «Computer Science», has different requirements, goals and objectives for the first (bachelor's), second (master's) and third (educational and scientific) levels of higher education. Thus, we propose to supplement the «speciality» entity with the labels *Bachelor* (to characterise the first (bachelor's) level) and *Master* (for the second (master's) level).

Thus, the structure of the modelling operator of the «speciality» entity for the second (master's) level of higher education in Cypher will be as follows:

CREATE(s: Master {name: «122_Computer_science», educational_qualification: «Master of Science in Computer Science», branch_knowledge: «12_Information _technologies»})

When modelling entities, it should be borne in mind that all educational components (disciplines) are divided into two groups: compulsory and elective. In turn, compulsory educational components are represented by two groups: general and professional training. That is, labels should be used when modelling the «discipline» entity: *Compulsory*, *Selective* to group them into compulsory and elective educational components. The disciplines of the general training cycle will be identified by the *General* label, and those of the professional training cycle by the *Professional* label.

The structure of the entity to be represented in the column of compulsory disciplines:

– of the general training cycle:

CREATE (d: Compulsory: General {name: «code_discipline_name»,hours: hours, credits: credits, semester: semester number, final control: offset/exam))

– of the professional training cycle:

CREATE (d: Compulsory: Professional {name: «code_discipline_name»,hours: hours, credits: credits, semester: semester number, final control: offset/ exam, swork))

The following is an example of creating a node to represent the data of the compulsory discipline of the professional training cycle «Information Control Systems»:

CREATE (ok1210: Compulsory: Professional {name: «OK1.2.10-Information Control systems», hours: 120, credits: 4, semester: 3, exam: «1 kur_r» })

The structure of the entity for elective subjects will be similar to the entity structures for compulsory subjects, but will contain only one

Selective label for filtering and selecting educational components of the student's choice.

When creating competence essences, one should take into account that there are different types of competences:

– general (universal) – do not depend on the professional orientation of the EPP;

– special or professional (subject) – aimed at gaining knowledge in a particular professional field.

The entities for displaying competences and learning outcomes will be similar in structure to the «speciality» entity and only labels will differ: *general_competencies* – general (universal) competences, *professional_competencies* – special or subject competences, *learning_outcomes* – learning outcomes.

The following are examples of creating a node for general competence (GC1):

CREATE (zk1: general_competencies {name: «GC1 Ability to think abstractly, analyse and synthesise»})

Fig. 2 shows the result of the creation of nodes of the EPP «Information Control Systems and Technologies».

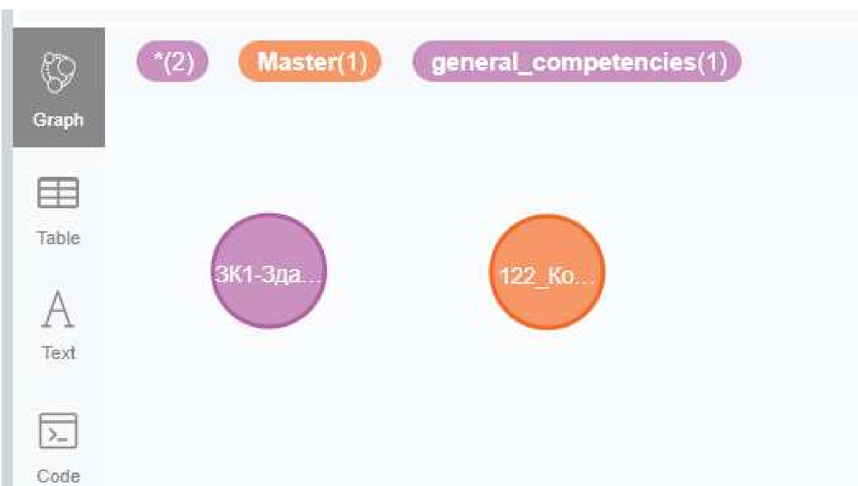

Figure 2. The result of creating the node in the Neo4j

Source: built by the authors

Having modelled the entities that will be represented by the nodes of the graph database, we should pay attention to the issue of their normalisation. It is generally accepted that only the relational data model should be normalised to reduce redundancy and eliminate anomalies when making changes to databases. That is, NoSQL databases can be stored in a denormalised form. However, this statement does not apply to graph databases. Always, after modelling the entity structures, it is necessary to check the fulfilment of the requirements of the normalisation theory, at least for the compliance of the entity representation with the third normal form (3NF) [30]. Given that graph databases, unlike other NoSQL models, are aggregate-ignore, solving the normalisation problem reduces duplication, facilitates and speeds up data modification, and ensures data consistency.

The final step in information modelling is to build relationships between entities, which will be represented by the edges of a graph database.

When modelling relationships, remember that there can be no incomplete relationships in the graph, i.e. each edge must connect two nodes, respectively:

– each created edge can be a directed edge if it leaves the initial (parent) node and enters the final (child) node, thus indicating its orientation;

– the created edge can be undirected, such a relationship can go in any direction (between nodes). Undirected relationships between entities are created in cases where there is no complete certainty about the direction of the relationship, but there is a need to implement certain queries that should reflect the interaction of entities regardless of the physical connection between them. The Neo4j supports the creation of such relationships between entities.

The literature review above suggests that there is currently no clear theory of relationship modelling. Relationships can describe any connections between entities [35]. Therefore, for our part, we believe it is necessary to distinguish the following rules for modelling relationships:

– creating relationships should be done strictly after modelling entities and normalising them;

– the semantics of the subject area should be taken into account and the information needs of end users should be analysed, which will ensure clarity and comprehensibility of structural relationships and help to avoid data inconsistency. Taking into account semantic aspects and subject area analysis when modelling relationships also allows us to optimise database query processes, reducing data processing and increasing the speed of responses to user requests;

– when defining relationships, we should focus on the key connections that are most important for analytics and decision-making. This will allow us to focus on the most critical relationships and optimise analysis and visualisation processes;

– relationship modelling should be limited to those necessary to meet specific information needs, avoiding excessive complexity that could make data management and analysis difficult;

– the formed relationships between entities are basic, but may not always reflect all the needs of users when implementing queries to a graph database. Therefore, an important aspect of relationship modelling is the research and modelling of possible queries to the graph database under design;

– given that oriented edges reduce the number of relationships when implementing queries, when modelling a graph, should avoid using undirected edges whenever possible.

Each relationship in a graph database can include variables, properties, and information about its type, which will reflect the semantics of the relationship.

Since the graphical data model is easily modified and extended, when modelling relationships, first of all, basic ones are created (those highlighted by verbs in the narrative description of the subject area). Other types of relationships, which will require modelling when implementing queries on a research graph model, were not considered in this study.

It should be noted that the current requirements for the formation of the EPP do not require the reflection of the links of elective educational components with other entities of the subject area. On the other hand, elective subjects may have links with the compulsory subjects of the professional training cycle, as they provide an opportunity for in-depth specialised study of certain topics or sections of the compulsory educational components. Given that each compulsory discipline of the professional training cycle has links to the relevant competencies, we can assume that elective courses may also have an indirect (transitory) link to these competencies. In addition, elective educational components that do not have any semantic links with the Compulsory educational components affect learning outcomes by specifying them or expanding them with additional knowledge and skills.

Based on the above, the authors propose to divide the elective courses into two conditional categories:

– Group A – elective courses that complement and expand the study of compulsory educational components;

– Group B courses courses that specify and expand the learning outcomes with additional knowledge and skills.

Therefore, the regulatory framework governing the development of EPPs needs to be modified in terms of taking into account the links between elective disciplines and the competencies and learning outcomes they form. If the legislative framework is so refined, elective educational components can be taken into account in the information modelling of the EPP.

After identifying the relationship and determining its type, it is advisable to proceed to describing its characteristics in a key-value format. These characteristics, for the most part, can be quantified (e.g., reflect distance, price, weight, absolute or percentage shares, certain rating coefficients, time intervals, etc.). As part of the construction of a graph database for this subject area, the authors concluded that there are no properties with quantitative characteristics. Therefore, only names and variables will have relations in the graph modelled by the EPP.

Below is an example of creating a relationship between a speciality (parent entity) and a general competence GC1 (child entity).

MATCH (s: Master {name: «122_Computer_science», educational_qualification: «Master of Science in Computer Science», branch_knowledge: «12_Information _technologies»}),

(zk1: general_competencies {name: «ZK1 – Ability to think abstractly, analyse and synthesise»}) CREATE(s)-[r:S_3K1]->(zk1) RETURN s, zk1

As a result of executing the command, a symmetrical directional connection was created, as shown in Fig. 3.

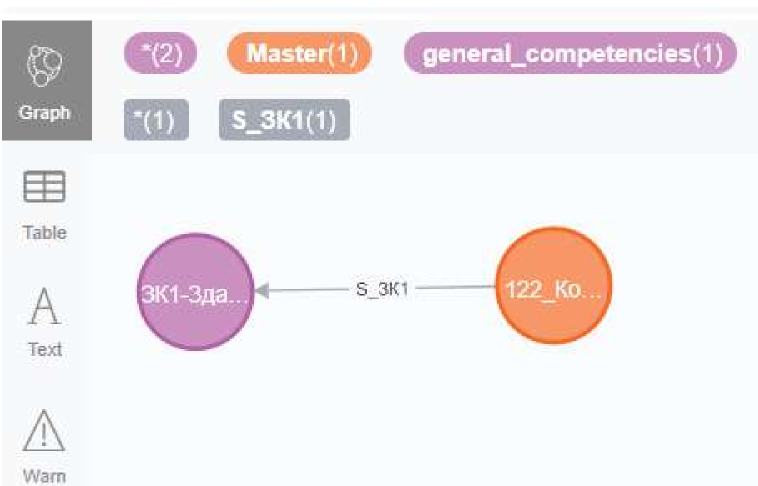

Figure 3. The result of creating the relationship in the Neo4j

Source: compiled by the authors

Other relations between the entities «speciality» and «competence» are created in a similar way, the result is shown in Fig. 4.

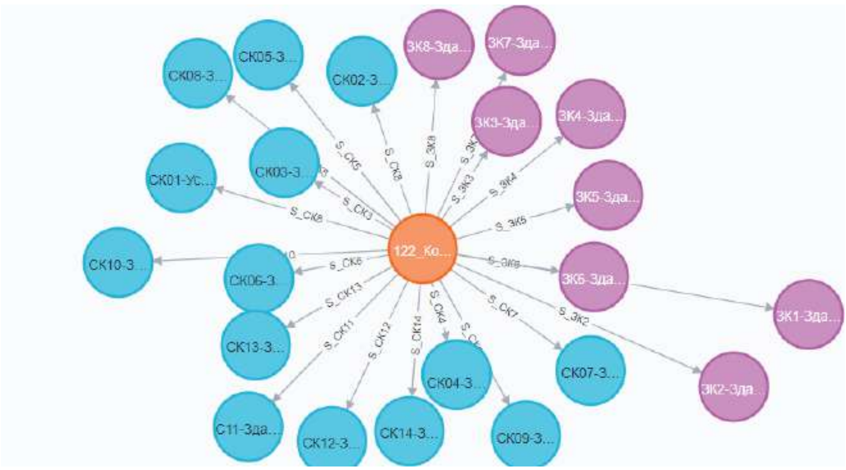

Figure 4. The result of modelling the relationship between the nodes of the EPP «Information Control Systems and Technologies» and general and professional competences in the Neo4j

Source: compiled by the authors

Further, the relations of the graph database are modelled on the basis that each specific general and special (professional) competence is formed by the compulsory educational components of the general and professional training cycles. That is, the parent node of this type of edge is a certain competence, and the final node is the corresponding academic discipline, which should take into account this competence and ensure its formation in students based on the results of training.

Next, it is necessary to build edges that reflect the relationship (links) of compulsory educational components with the corresponding learning outcomes. This group of edges reflects the final essence of students' learning and indicates the results that will be obtained by them in the form of theoretical knowledge and practical skills and abilities. The parent node of such edges should be the academic disciplines, and the child node should be the learning outcomes provided by the study of the relevant disciplines [14].

Based on the results of information modelling, a graph of the subject area is formed, which demonstrates the relationships between entities. Thus, in Fig. 5 shows a fragment of the result of graph database modelling for the EPP «Information Control Systems and Technologies».

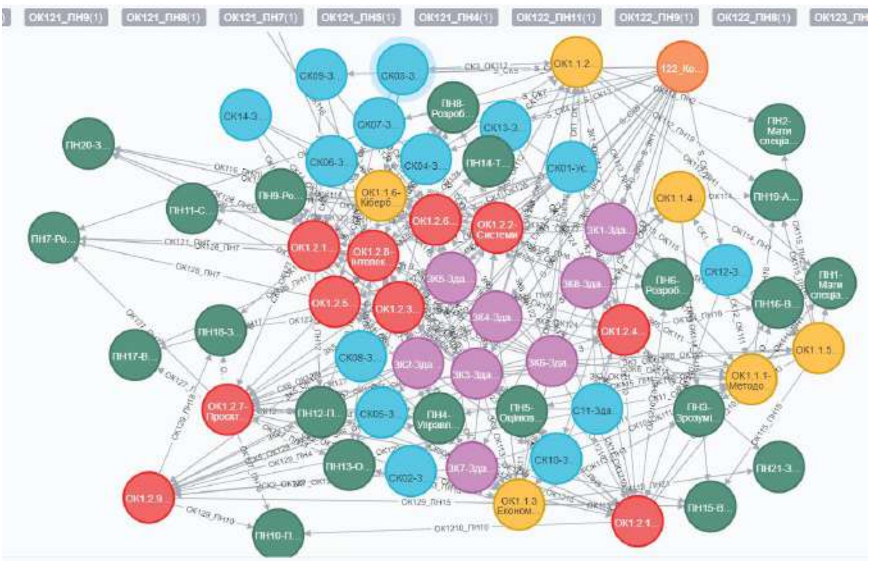

Figure 5. The result (fragment) of information modelling
of the EPP «Information Control Systems and Technologies» in the Neo4j

Source: built by the authors

The created graph allows to efficiently perform various queries for detailed research, analysis and evaluation of the EPP for balance and quality.

Below, as an example, is a request for a sample of competences covered in the discipline of the professional cycle «Information Control Systems»:

MATCH (ok1210: Compulsory: Professional {name: «OK1.2.10-Information control systems», hours: 120, credits: 4, semester: 3, exam: «1 kur_r» })<-[]-(zk) RETURN ok1210

The visualisation of the results of this query in the form of a graph is shown in Fig. 6.

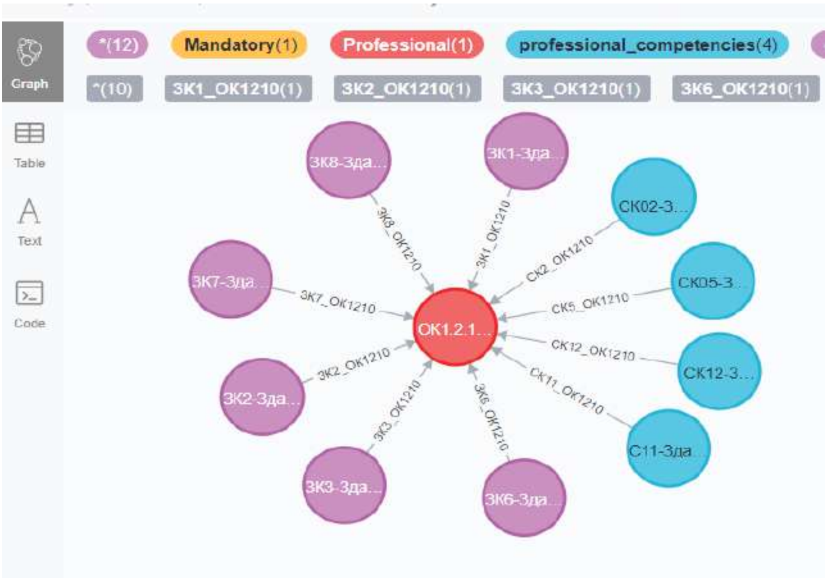

Figure 6. Graph of correspondence between general and professional competences and the discipline «Information Control Systems» in the Neo4j

Source: built by the authors

Conclusions and prospects for further research

The study proves the expediency of using graph databases to optimise the operational processes of HEIs in matters related to the development, analysis and improvement of educational programs.

The considered theoretical, methodological and practical aspects of creating and modelling information objects in graph models of data representation allowed the authors to propose a unified methodology of information modelling and demonstrate its key ideas on the example of creating a graph of the EPP «Information control systems and technologies» of the second (master's) level of higher education, which was introduced at the Kyiv National Economic University named after Vadym Hetman. It has been determined that graph databases provide flexible management of unstructured data, improving data visualisation and analysis, which is critical for modern HEIs. Applied aspects of information modelling are performed using the Neo4j DBMS, the main advantage of which is the ability to effectively store, visualise and analyse complex relationships between numerous entities of the subject area.

The data modelling algorithm proposed by the authors focuses on creating a graph structure that reflects complex interdependencies and interactions between objects of the subject area, while ensuring the flexibility and scalability of the model. Data modelling in graph models should be carried out in stages, including: studying the subject area; determining the type of data representation model; identifying entities, their properties, labels; normalising data in the identified entities; identifying relationships between entities, determining their type and building relationships. The process of data modelling and building graph databases requires developers to study the subject area in detail, to have a deep understanding of user needs, and to take into account the specifics of various types of relationships between entities, without which it is impossible to achieve an accurate and efficient representation of data in graph models.

Information modelling of data using Neo4j allows solving a number of important organisational and methodological tasks: to demonstrate the links between competences and learning outcomes within the EPP; to present the contribution of each educational component to the formation of competences and learning outcomes; to identify and reveal competences and programme learning outcomes that are not formed by the disciplines of the current curriculum; to determine the ratio of the level of competence formation by the disciplines of general and professional training blocks;

The proposed methodology and modelling algorithm not only form the basis for further research, but also open up new opportunities for improving the quality and efficiency of the EPP, providing a basis for innovative approaches to managing the quality of education.

Prospects for further research are to continue the work of the author’s team on the implementation of the developed graph database into an information control system with the functions of forming educational programs to promote digitalisation and automation of management processes in the field of education and activities of HEIs. Also, in future works of the authors, it is planned to expand the created graph by supplementing it with new nodes with appropriate relationships that characterise the subject matter of each educational component by highlighting the topics included in the work program of the discipline. Such filling of the graph with new nodes and edges will allow clustering nodes and combining them by similarity of topics and identifying elements of duplication in the work programs of different educational components. This will allow for an in-depth analysis of the EPPs and informed decisions on their improvement, which will ultimately contribute to improving the quality of university education in general.

References

1. Ministry of Education and Science of Ukraine. (2021). Draft Concept of Digital Transformation of Education and Science for the period up to 2026. *Website*: https://mon.gov.ua/ua/news/koncepciya-cifrovoyi-transformaciyi-osviti-i-nauki-mon-zaproshuye-do-gromadskogo-obgovorennya

2. Ministry of Education and Science of Ukraine. (2024). Digital transformation of education and science. *Website*: https://mon.gov.ua/ua/tag/cifrova-transformaciya-osviti-ta-nauki.

3. Verkhovna Rada of Ukraine (2014). On higher education: Law of Ukraine №1556-VII. *Website*: https://zakon.rada.gov.ua/laws/show/1556-18#Text.

4. Finikov, T.V., & Tereshchuk, V.I. (2018). *Local quality management systems: world experience and Ukrainian practices of building*. Kyiv: Takson.

5. Gogunsky, V.D., and Savelieva, O.S. (2014). The formative role of higher education standards in the organisation of the educational process. *Ways of implementing the credit-modular system of organising the educational process and test forms of controlling students' knowledge: materials of the scientific and methodological seminar*. Odesa: Science and Technology, No. 9, 3–9.

6. Mudra, S.V. (2016). *Managing the quality of training of graduates of higher education institutions*. Kyiv. [Online]. Available: https://dglib.nubip.edu.ua/server/api/core/bitstreams/e11344b7-c525-41ca-9e86-2ce1c89e09aa/content.

7. Proskura, S.L., & Litvinova, S.G. (2019). Formation of professional competence of future bachelors of computer science. *Physical and mathematical education: a scientific journal*. Ministry of Education and Science of Ukraine, Sumy State Pedagogical University named after A.S. Makarenko, Faculty of Physics and Mathematics, 2 (20), 137–146.

8. Sytnyk, N.V., and Zinovieva, I.S. (2021). Modern NoSQL databases in the training of bachelors in Computer Science. *Information Technologies and Learning Tools*, 1, 255–271. https://doi.org/10.33407/itlt.v81i1.3098.

9. EU Institutional Initiative. (2022). New version of the European classification of Skills, Competences, Occupations and Qualifications is live (Digital skills). Available at: https://digital-skills-jobs.europa.eu/en/latest/news/new-version-european-classification-skills-competences-occupations-and-qualifications.

10. Ministry of Education and Science of Ukraine. (2022). On approval of the Standard of Higher Education in the speciality 122 Computer Science for the second (master's) level of higher education: Order No. 393. *Website*: https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni %20standarty/2022/04/28/122-Kompyuterni.nauky-mahistr.393-28.04.22.pdf.

11. Meier, A., & Kaufmann, M. (2019). *SQL & NoSQL Databases*. Wiesbaden: Springer Fachmedien.

12. Sytnik, N.V., and Zinovieva, I.S. (2022). *Organisation of NoSQL databases*: a workshop. Ministry of Education and Science of Ukraine, Kyiv

National Economic University named after Vadym Hetman. Kyiv: KNEU. [Online]. Available: https://ir.kneu.edu.ua/handle/2010/38129.

13. Cielen, D., Meysman, A., & Ali, M. (2016). *Introducing Data Science: Big Data, Machine Learning, and more, using Python tools*. Manning. [Online]. Available: https://bedford-computing.co.uk/learning/wp-content/uploads/2016/09/introducing-data-science-machine-learning-python.pdf.

14. Zinovieva, I., Sytnyk, N., Denisova, O., & Artemchuk, V. (2024). Support for the development of educational programs with graph database technology. *Data-Centric Business and Applications* (Lecture Notes on Data Engineering and Communications Technologies, Vol. 195). https://doi.org/10.1007/978-3-031-54012-7_14.

15. Pasterk, S., & Bollin, A. (2017). Graph-based analysis of computer science curricula for primary education. *In Proceedings – Frontiers in Education Conference*, FIE, October, 1–9. https://doi.org/10.1109/FIE.2017.8190610.

16. Robinson, I., Webber, J., & Eifrem, E. (2015). *Graph Databases*, 2nd Edition. O'Reilly Media, Inc. Available: https://web4.ensiie.fr/~stefania.dumbrava/OReilly_Graph_Databases.pdf.

17. Esser, S., & Fahland, D. (2021). Multi-Dimensional Event Data in Graph Databases. *Journal of Data Semantics*, 10, 109–141. https://doi.org/10.1007/s13740-021-00122-1.

18. Tanveer, A., Sharma, C., Sinha, R., et al. (2023). Tracing security requirements in industrial control systems using graph databases. *Software and Systems Modeling*, 22, 851–870. http://doi.org/10.1007/s10270-022-01019-8.

19. Gephi (2024). The Open Graph Viz Platform. *Site*: https://gephi.org/.

20. Malik, N.R. (1975). Graph theory with applications to engineering and computer science. *Proceedings of the IEEE*, 63(10), 1533–1534. https://doi.org/10.1109/PROC.1975.9996.

21. DB-Engines (2024). DB-Engines Ranking. *Site*: https://db-engines.com/en/ranking.

22. Liu, M., Zhou, B., Li, J., Li, X., & Bao, J. (2023). A Knowledge Graph-Based Approach for Assembly Sequence Recommendations for Wind Turbines. *Machines*, 11, 930. https://doi.org/10.3390/machines11100930.

23. Guia, J., Soares, V.G., & Bernardino, J. (2017). Graph Databases: Neo4j Analysis. *Presented at the International Conference on Enterprise Information Systems*. [Online]. Available: https://pdfs.semanticscholar.org/5164/3300e31bf2c79031c10c8a2007b89c8deb6a.pdf.

24. Angles, R., Bonifati, A., Dumbrava, S., Fletcher, G., Green, A., Hidders, J., Li, B., Libkin, L., Marsault, V., Martens, W., Murlak, F., Plantikow, S., Savkovic, O., Schmidt, M., Sequeda, J., Staworko, S., Tomaszuk, D., Voigt, H., Vrgoc, D., Wu, M., & Zivkovic, D. (2023). PG-Schema: Schemas for Property Graphs. *Proceedings of the ACM on Management of Data*, 1(2), Article 198. https://doi.org/10.1145/3589778

25. Davies, A., Hassey, A., Williams, J., & Moulton, G. (2022). Creation of a core competency framework for clinical informatics: From genesis to

maintaining relevance. *International Journal of Medical Informatics*, 168, 104905. https://doi.org/10.1016/j.ijmedinf.2022.104905.

26. Wen, P., Ma, Y., & Wang, R. (2023). Systematic knowledge modeling and extraction methods for manufacturing process planning based on knowledge graph. *Advanced Engineering Informatics*, 58, 102172. https://doi.org/10.1016/j.aei.2023.102172.

27. Neo4j. (2024). Main page. *Site*: https://neo4j.com.

28. Roy-Hubara, N., & Sturm, A. (2020). Design methods for the new database era: a systematic literature review. *Softw Syst Mode*l, 19, 297–312. https://doi.org/10.1007/s10270-019-00739-8.

29. Pokorny, J. (2017). Modelling of graph databases. *In Advances in Databases and Information Systems*. Cham: Springer Nature Switzerland, 04–17. http://dx.doi.org/10.25073/jaec.201711.44.

30. Skavantzos, P., & Link, S. (2023). Normalizing property graphs. *Proceedings of the VLDB Endowment*, 16(4), 3031–3044. https://dx.doi.org/10.14778/3611479.3611506.

31. Singh, A. (2022). *Graph Database Modeling with Neo4J*. 2nd Edition. Amazon Digital Services LLC – Kdp.

32. Turu, A., & Koroglu, O. (2020). *Graph Databases and Neo4J*. Cham: Springer Nature Switzerland.

33. Neo4j. (2024). Graph Data Modeling Guide. *Site*: https://neo4j.com/docs/getting-started/data-modeling/guide-data-modeling/.

34. Allen, D. (2020). Graph Modeling: All About Relationships. *Site*: https://medium.com/neo4j/graph-data-modeling-all-about-relationships-5060e46820ce.

35. Allen, D. (2020). Graph Modeling: All About Super Nodes. *Site*: https://medium.com/neo4j/graph-modeling-all-about-super-nodes-d6ad7e11015b.