

УДК 004.23

Компан С.В.<sup>1</sup>, аспірант

### Побудова інформаційної системи “Облік робочого часу працівників вищого навчального закладу”

<sup>1</sup> Київський національний університет імені Тараса Шевченка, 83000, м. Київ, пр-т. Глушкова 4д,  
e-mail: [robin\\_2005@mail.ru](mailto:robin_2005@mail.ru)

S.V. Kompan<sup>1</sup>, postgraduate

### Information system construction “Working time accounting of higher educational establishments personnel”

<sup>1</sup> Taras Shevchenko National University of Kyiv, 83000, Kyiv, Glushkova st., 4d,  
e-mail: [robin\\_2005@mail.ru](mailto:robin_2005@mail.ru)

*В статті проводиться аналіз предметної області для побудови інформаційної системи «Облік робочого часу працівників вищого навчального закладу». Визначені та документовані основні сутності предметної області та зв'язки між ними. Визначені обмеження, які накладаються на сутності. Визначені вимоги до інтерфейсу користувача, а також його практична реалізація. Показана доцільність використання типу даних ltree у базі даних та застосування отриманого раніше теоретичного результату побудови батьківського класу за двома класами, розповсюдивши результат на таблиці бази даних.*

*Ключові слова: організаційно-штатна структура, облік робочого часу, інформаційна система, проектування інформаційної системи, успадкування таблиць.*

*One of the factors of enterprise expenditure minimization and optimization of proprietorship in accordance with current market situation is information systems. The successful functioning of the latter is defined by the quality of its design and practical realization. The paper presents analysis of the subject area for information system construction “Working Time Accounting of Higher Educational Establishments Personnel”. The essence of the subject area under research is singled out and documented and the relationship between its basic notions is described. The restrictions that are superimposed on the essence are singled out. The user interface demands, as well as the interface practical realization, are outlined. The expedience of database use of the type ltree and the application of the earlier obtained research result of parent class construction for two classes by spreading the result on database tables is presented.*

*Key Words: organizational personnel structure, personnel inventory, employee, working time, working time accounting, information system, information system design, table inheritance.*

Статтю представив д.ф.-м.н., професор Буй Д.Б.

Нагадаймо загальне змістовне означення: інформаційна система (ІС) – програмно-апаратний комплекс, який призначений для збереження та обробки інформації предметної області [1]. Стаття присвячена побудові ІС для предметної області, що характеризується такими основними поняттями (онтологіями):

**Постановка задачі.** Необхідно розробити ІС для ведення обліку робочого часу працівників ВНЗ. Для реалізації поставленої задачі потрібно створити відповідну ІС, в основі якої лежить база даних (БД). ІС «Облік робочого часу працівників ВНЗ» призначена для автоматизації обліку робочого часу працівників ВНЗ. Ця ІС повинна мати наступні функціональні можливості.

**Ведення організаційно-штатної структури ВНЗ.** Структуру організаційних одиниць (ОО)

ВНЗ (за іншою термінологією структурних підрозділів ВНЗ) можна представити у вигляді кореневого дерева, яке обмежене чотирма рівнями. А саме, на першому рівні знаходиться сутність “ВНЗ”, на другому рівні – “факультет (інститут)”, “ректорат”, на третьому рівні – “кафедра”, “деканат” і структурні підрозділи ректорату, на четвертому – “науково-дослідна лабораторія” (НДЛ) або науково-дослідний сектор (НДС). Задача ІС полягає у веденні цієї організаційно-штатної структури, тобто треба передбачити можливості редагування атрибутів ОО, знищення та додавання ОО.

**Ведення штатних розписів структурних підрозділів (“деканат”, “кафедри”, тощо).** ІС повинна містити інформацію про штатний розпис ВНЗ, що може представлятися у вигляді таблиці.

Також штатний розпис повинен містити найменування посад, чисельність персоналу, надбавки та доплати, посадові оклади. В таблиці код за класифікатором професій визначається згідно національного класифікатора професій ДК 003:2010 [2].

**Ведення кадрового складу.** ІС повинна містити інформацію про співробітників (штатних та сумісників), які прикріплені до ОО, та можливість виведення такої інформації в звітах. Такою інформацією є: П.І.Б, ОО, до якої прикріплений співробітник, посада (згідно штатного розпису), ставка, науковий ступінь, вчене звання, тощо.

**Ведення графіків відпусток.** ІС повинна містити графік відпусток кожного працівника ВНЗ, який може бути використаний для генерації звітів.

**Облік робочого часу.** ІС повинна мати можливість надавати працівнику фіксувати час початку та закінчення робочого дня, з вказанням причин невиходу на роботу. Також ІС повинна містити інформацію про обідню перерву працівника.

**Генерація таблиць обліку робочого часу та різних звітів.** На основі інформації, яка міститься в ІС, повинна надаватися можливість генерації звітів про робочий час працівників за різноманітними критеріями.

**Ведення історії змін сутності.** ІС повинна надавати можливість вести історію змін атрибутів сутностей. Наприклад, історію зміни прізвища працівника, назв ОО, тощо.

Етап концептуального проектування – це створення концептуальної моделі даних, виходячи з представлення користувачів про предметну область. Для виконання цього етапу потрібно виконати ряд послідовних кроків.

#### **Визначення сутностей та їх документування**

Розглянемо основні сутності нашої предметної області:

- структурні підрозділи: ВНЗ, факультет (інститут), кафедра (деканат), НДІ (НДС);
- науково-дослідні теми – визначаються переліком тем, які виконуються в ВНЗ;
- штатний розпис, містить інформацію про посади структурного підрозділу;
- наукові ступені; наявний перелік (класифікатор) наукових ступенів, які можуть мати працівники ВНЗ: кандидат наук, доктор наук;
- вчене звання; наявний перелік (класифікатор) вчених звань, які можуть мати працівники ВНЗ:

старший науковий співробітник, доцент, професор;

– співробітник; задається інформація про співробітника: П.І.Б, дані для аутентифікації на сайті (логін та пароль), поштова адреса, вчене звання, науковий ступінь, фактичний і плановий графіки роботи, наявність адміністративних прав;

– посади; наявний перелік (класифікатор) посад штатних розписів: наприклад, інженер певної категорії, молодший науковий співробітник, науковий співробітник, асистент, доцент, професор, декан;

– облік робочого часу; визначається інформація про час початку і закінчення робочого дня працівника;

– причини відсутності на робочому місці; є перелік (класифікатор) причин, за яких працівник може бути відсутнім на робочому місці (лікарняний, відпустка, тощо);

– плановий графік роботи; визначається графік роботи, затверджений адміністрацією;

– робочий день; маються на увазі часові інтервали, під час яких працівник знаходився на робочому місці;

– відпустка; графік відпустки працівника на основному місці роботи;

– час життя сутності; ведення інформації про історію змін характеристик сутності.

#### **Визначення зв'язків між сутностями та їх документування**

##### **Зв'язки між структурними підрозділами.**

Структурний підрозділ підпорядковується одному структурному підрозділу (СП) вищого рівня, за винятком міжкафедральних факультетських лабораторій та міжфакультетських лабораторій. Кожна “наступна” сутність зв'язується з “попередньою” у відношенні “один до багатьох”.

Таблиці, що відповідають СП, мають однакові атрибути. Тоді можна скористатися результатом [3], який розповсюджується на таблиці, що задають сутності в БД. А саме, скористаємося механізмом успадкуванням таблиць, яке надає система управління базами даних (СУБД) PostgreSQL [4] та винесемо спільні атрибути в нову таблицю, яка буде батьківською; тоді таблиці, які описують СП, будуть похідними. Також потрібно врахувати той факт, що в батьківській таблиці будуть міститися всі записи похідних таблиць, які будуть мати атрибути схеми батьківської таблиці. Це дає можливість при вибірці записів з декількох таблиць замість використання в SQL-запиті конструкції UNION [ALL] (призначеної для об'єднання рядків таблиць), використовувати тільки батьківську таблицю. Виходячи з

наведених міркувань, побудуємо схему наступним чином

**Зв'язок між сутностями факультет (інститут) – деканат.** На кожному факультеті (інституті) є один деканат, тому такі сутності знаходяться у зв'язку один до одного (1:1).

**Зв'язок між сутностями працівник – вчене звання (науковий ступінь).** У випадку, коли працівник має декілька вчених звань (наукових ступенів), під час табелювання береться найвище звання (науковий ступінь). Тому ці сутності пов'язані між собою у відношенні 1:1.

**Зв'язок між сутностями працівник – плановий графік роботи.** Кожен працівник має графік роботи, затверджений адміністрацією, а також має фактичний графік роботи, який визначається з сутності робочий день.

**Зв'язок між сутностями працівник, теми та штатний розпис.** Кожен працівник на штатних засадах може працювати лише в одному СП. В той же час, він може працювати за сумісництвом в декількох інших СП. Працівник зараховується на тему (на одну на штатних засадах, на декілька тем за сумісництвом). Кожна тема виконується лише в одному СП. В СП можуть виконуватися декілька тем. Таким чином, сутності працівник і тема знаходяться в зв'язку багато до багатьох (M:N), а сутності тема і СП – у зв'язку багато до одного (M:1).

**Зв'язок між сутностями причина та робочий день.** В один робочий день працівник може мати одну причину невиходу на роботу. Тому вказані сутності знаходяться у відношенні 1:1. Розглянемо призначення атрибутів в таблицях. Зробимо припущення: якщо в двох таблицях присутні атрибути з однаковим іменем, то вони несуть однакове семантичне значення.

1) Таблиці Корінь СО, ВНЗ, Ректорат, Кафедра, Сектор (НДЛ) мають такі атрибути: id, depts\_name, pidrozdil, unit. В таблицях атрибут id – первинний ключ таблиці, depts\_name – ім'я підрозділу, pidrozdil – визначає місце даного підрозділу в організаційній структурі, яка представляється у вигляді деревоподібної структури та задається у вигляді чисел, розділених точкою; кожне число визначає приналежність до відповідного СП, unit – приймає одне з наступних значень: ВНЗ, ректорат, факультет, кафедра, сектор. Сутність факультет (інститут) має ще два додаткових атрибути iddecanat та idaspirantura, які слугують для зв'язку з таблицями Деканат і Відділ аспірантури.

2) В таблицях Деканат і Відділ аспірантури містяться атрибути, значення яких ми розглянули вище.

3) Таблиця Штатний розпис містить такі атрибути: id, posada – назва посади працівника згідно з класифікатором посад, pidrozdil, kodklasifikator – код посади згідно класифікатора посад, nadbavka – визначає для працівника значення надбавки за науковий стаж.

4) Таблиця РоботаПрацівник призначена для реалізації зв'язку M:N між таблицями Робота, Працівник, Штатний розпис; вона містить атрибути, в яких містяться ідентифікатори записів вищеназаних таблиць.

5) Таблиці Працівник та Робота (теми) містять такі атрибути: id, pidrozdil, chief та secretary – визначають відповідно керівника та секретаря (табельника) з таблиці Працівник, name – назва теми, startdata, enddata – відповідно дата початку і закінчення теми. Для назви робіт, які не є науковими темами, будемо використовувати атрибут pidrozdil.

**Обмеження, які накладаються на атрибути таблиць в БД**

Розглянемо обмеження на значення атрибутів таблиці.

- id – первинний ключ сутності “підрозділ”;
- path – має унікальне значення та останнє число повинно співпадати з id;
- Працівник, який працює в одному СП (на темі) має одну посаду, тому введемо обмеження на таблицю РоботаПрацівник: атрибути workid, userid – композитний первинний ключ.

**Реалізація збереження історії змін сутності**

Сутність протягом часу свого існування може зазнавати зміни. Так наприклад, у сутності факультет з часом може змінитися назва. Актуальна назва факультету необхідна при побудові звітів. Тому постає задача збереження історії змін сутності за часом. Цю задачу будемо вирішувати наступним чином: створимо додаткову таблицю (в ім'я додамо TTL – time to live), яка є копією таблиці, по якій потрібно відстежувати історію, додавши до нової таблиці два атрибута start і end. В цих атрибутах буде зберігатися дата початку та закінчення існування запису. Для поточного стану запису атрибут end буде рівний особливому значенню NULL.

Таблиця, по якій відстежується історія змін, буде зв'язана з таблицею, в якій ці зміни зберігаються, у відношенні 1:N.

Історія зміни сутності буде заноситися в таблицю за допомогою тригера, який буде спрацьовувати на додавання нового рядка (Insert),

оновлення атрибутів рядка (Update) та знищення рядка (Delete).

Уточнимо постановку задачі. ІС повинна надати користувачу можливість працювати в двох режимах: режим користувача та адміністративний режим (при наявності відповідного права). В режимі користувача працівник заходить шляхом вводу логіна та пароля на свою сторінку, на якій присутня інформація: поточний час, який береться з серверу БД, посада працівника згідно з штатним розписом відповідного СП, П.І.Б. працівника, системний час, обідня перерва. Кожен працівник повинен мати затверджений керівництвом графік роботи (будемо називати його плановим) та фактичний – це той час, який витрачає співробітник на робочий процес і який буде зберігатися в БД ІС. У випадку, коли працівник відсутній на робочому місці, ІС повинна надати можливість вказати причину відсутності. На сторінці користувача повинна відображатися інформація про плановий і фактичний графіки роботи.

Також передбачається, що ІС повинна зберігати історію змін назв СП ВНЗ, П.І.Б. працівників, планових графіків робочого часу, розміру ставки, тощо.

#### **Вимоги до інтерфейсу користувача**

На сторінці користувача повинні бути присутні дві кнопки: «початок роботи», «закінчення роботи». При натисненні на них в БД записується час відповідно початку і закінчення роботи працівника. При проектуванні ІС слід передбачити, що працівник може працювати як за основним місцем роботи, так і за сумісництвом на відповідну частину ставки.

В режимі користувача повинна бути присутня кнопка для переключення на режим адміністрування (при умові, що даному користувачеві надані адміністративні права). Адміністративний режим надає можливість: 1) змінювати інформацію про ВНЗ, СП, П.І.Б. посади працівників, їх планові графіки роботи, розміри ставки; 2) додавати нових користувачів, надавати їм логін, пароль, встановлювати право входити в адміністративний режим; 3) додавати працівника до СП. Також в цьому режимі повинна надаватися можливість введення планового графіку роботи працівника.

#### **Практична реалізація задачі**

Програмну реалізацію ІС проведемо за технологією клієнт-сервер. При розробці ІС слід її розділити на дві складові: інтерфейс користувача, який будемо називати фронтенд (front-end), та серверна частина, куди будуть входити в якості

компонент веб-сервер та БД. Доступ до ІС повинен бути з будь-якого комп'ютера локальної мережі ВНЗ. Тоді доцільно логіку роботи ІС перенести на сторону серверної частини, що полегшить здійснення супроводження ІС.

При розробці ІС в першу чергу потрібно визначитися з СУБД. При цьому виборі треба визначитися з моделлю даних (сукупністю структур даних та операцій над ними). Як відомо, поширеними моделями даних є: реляційна, об'єктна та об'єктно-реляційна. Також при виборі СУБД доцільно враховувати, чи надає конкретна СУБД можливість здійснювати успадкування таблиць. Це надасть більш гнучкі можливості побудови таблиць сутностей. Можливість успадковувати таблиці дає нам об'єктно-реляційна СУБД. В якості реалізації фронтенду та його взаємодії з БД доцільно обрати об'єктно-орієнтовану мову програмування (framework), яка надасть завдяки відомим концепціям об'єктно-орієнтованого програмування (інкапсуляція, успадкування, поліморфізм) широкі можливості та гнучкість при написанні коду програми.

В якості web сервера будемо використовувати Apache [5]. Реалізовувати інтерфейс користувача будемо на мові програмування Python [6], з використанням фреймворку для веб-додатків Django [7], який дозволить здійснювати зв'язок інтерфейсу користувача з БД. В якості СУБД візьмемо об'єктно-реляційну СУБД Postgresql [4, 8]. Дана СУБД має переваги над іншими завдяки тому, що вона є об'єктно-реляційною, з можливістю успадкування таблиць, дозволяє працювати з складними типами даних, підтримує можливість створювати типи даних користувача. В даній СУБД підтримується можливість використання тригерів та збережених процедур. Вона також надає можливість встановлювати обмеження цілісності на атрибути таблиці, а саме: not null – не NULL; unique – унікальність; primary key – первинний ключ; foreign key/references – зовнішні ключи, посилання; check – перевірка; exclude – перевірка унікальності по складній умові.

В БД передбачається використовувати нестандартні типи даних. Як зазначалося вище, такі сутності предметної області як СП можна описати за допомогою деревоподібної структури. Тоді доцільно використати тип даних Itree, який дозволяє зберігати деревоподібні структури в вигляді міток, а також надає широкі можливості пошуку по них.

Дамо обґрунтування переваги використання саме такого типу даних. Предметна область, як

було сказано вище, має такі сутності: ВНЗ, факультет (інститут), кафедра, НДЛ (НДС). Кожному СП відповідає рядок відповідної таблиці БД. Елемент однієї сутності може мати декілька елементів іншої (підпорядкованої) сутності. Наприклад, в ВНЗ є факультети (інститути), кожен факультет містить принаймні три кафедри (які в свою чергу можуть містити декілька лабораторій та секторів).

Для реалізації цієї можливості в БД таблиці, які відповідають вказаним сутностям, повинні бути з'єднані за допомогою зовнішніх ключів. При такому підході в SQL-запиті для визначення, наприклад, всіх лабораторій (секторів) деякої кафедри, яка знаходиться на вказаному факультеті, певного ВНЗ треба застосувати з'єднання чотирьох таблиць.

Як відомо, використання з'єднання суттєво впливає на швидкість виконання запитів; саме тому був обраний тип даних *ltree*.

В результаті замість чотирьох таблиць буде використовуватися тільки одна, в якій в одному з атрибутів буде знаходитися мітка у точковій нотації, що буде визначати приналежність даної структурної одиниці до інших.

#### Список використаних джерел

1. А.С. Грошев Базы данных. Учебное пособие / Грошев А.С. – Архангельск, 2005. – 111 с.
2. Класифікатор професій України ДК 003:2010 [Електронний ресурс] <http://www.dk003.com/>
3. Буй Д.Б. Диаграммы классов ООП: формализация и анализ / Д.Б. Буй, С.В. Компан // Труды ИПММ НАН Украины. – 2013. – Т. 27. – С. 51-65.
4. Документация по PostgreSQL 9.1.1. Наследование / [Електронний ресурс] <http://postgresql.ru.net/manual/ddl-inherit.html>
5. Apache HTTP Server Project [Електронний ресурс] <http://httpd.apache.org>
6. Мова програмування Python [Електронний ресурс] <https://www.python.org/>
7. Фреймворк Django [Електронний ресурс] <https://www.djangoproject.com/>
8. СУБД PostgreSQL [Електронний ресурс] <http://www.postgresql.org/>

Покажемо це на прикладі. Нехай Київський національний університет імені Тараса Шевченка має ключовий ідентифікатор *id=1*, факультет кібернетики має відповідно *id=3*, а кафедра теорії та технології програмування *id=2*. Тоді для того, щоб показати однозначну приналежність кафедри відповідному факультету та університету, будується мітка вигляду 1.3.2.

#### Висновки

В статті наведені функціональні можливості ІС “Облік робочого часу працівників ВНЗ”. Проведено визначення сутностей, їх документування, зв'язки між ними. Визначено обмеження, які накладаються на атрибути таблиць в БД. Запропоновані шляхи практичної реалізації цієї системи. Запропонована реалізація збереження історії змін сутності. Показано практичне застосування результату щодо винесення спільних атрибутів в батьківський клас, викладеного в [3]. Наступним кроком реалізації ІС планується її практична реалізація.

#### References

1. GROSHEV, A.C (2005) *Bazy dannyh. Uchebnoe posobie*. Archngelsk.
2. *Klassifikator profesiy Ukrainy DK 003:2010* [Online] Available from: <http://www.dk003.com/> [Accessed: 19th May 2014].
3. Buy D.B. (2013) *Diagrammy klassov OOP: formalisatsia i analis*. Trudy IPMM NAN Ukrainy, tom 27, pp. 51-65
4. *Postgresql 9.1. Inheritance* [Online] Available from: <http://www.postgresql.org/docs/9.1/static/tutorial-inheritance.html> [Accessed: 19th May 2014].
5. *Apache HTTP Server Project* [Online] Available from: <http://httpd.apache.org> [Accessed: 19th May 2014].
6. *Python* [Online] Available from: <https://www.python.org/> [Accessed: 19th May 2014].
7. *Python Web framework Django* [Online] Available from: <https://www.djangoproject.com/> [Accessed: 19th May 2014].
8. *DBMS PostgreSQL* [Online] Available from: <http://www.postgresql.org/> [Accessed: 19th May 2014].

Надійшла до редколегії 22.05.14