

УДК 378.147:004.65

Г. С. ПОГРОМСЬКА

м. Миколаїв

КЛЮЧОВІ АСПЕКТИ МЕТОДОЛОГІЇ ПРОЕКТУВАННЯ СУЧАСНИХ БАЗ ДАНИХ

У статті висвітлені базові теоретичні питання, що постають під час викладання курсу «Методологія проектування сучасних баз даних». Виконано порівняльний аналіз у підходах до проектування реляційних та об'єктно-орієнтованих БД та продемонстрована неефективність моноспрямованих систем управління базами даних (СУБД) для побудови БД, що входять до складу сучасних інформаційних систем.

Ключові слова: бази даних (БД), система управління базами даних (СУБД), інформаційна система (ІС), етапи проектування, реляційні СУБД, об'єктно-орієнтовані СУБД, постреляційні СУБД, моноспрямовані СУБД.

Постановка проблеми. У більшості сучасних галузей діяльності людини постає проблема створення та використання інформаційних систем. Сьогодні такі системи є нагальною потребою, і попит на грамотних спеціалістів в цій галузі постійно зростає. Оскільки усі інформаційні системи базуються на концепції баз даних, то природно, що без знайомства з засадами проектування баз даних в наш час неможливо бути кваліфікованим програмістом. Тому можна зазначити, що навички роботи в цій галузі не тільки підвищують інтелектуальний потенціал програміста, але і є в цьому питанні одним з основоположних факторів.

Дисципліна «Методологія проектування сучасних баз даних» викладається на спеціальності «Інформатика» на 6 курсі в обсязі 120 годин (лекції – 18 годин, лабораторні роботи – 12 годин та 56 годин на самостійну роботу). Метою дисципліни «Методологія проектування сучасних баз даних» є вивчення теоретичних основ методології проектування баз даних та набуття практичних навичок з етапів проектування баз даних із застосуванням CASE-засобів. За завдання покладено: розкрити сутність методології проектування БД, вивчити етапи методології проектування БД, забезпечити практичні навички роботи з інфонологічного та фізичного проектування БД із застосуванням CASE-засобів AllFusion Ervin Data Modeler та Rational Rose.

Методологія реалізується через певні технології та стандарти, що їх підтримують, методики та інструментальні засоби, які забезпечують виконання процесів життєвого циклу. Оскільки на даний час все більшої популярності набувають гібридні або постреляційні СУБД, які реалізують відразу кілька моделей даних в рамках єдиного сховища даних, то увагу в зазначеному курсі необхідно приділити особливостям у підходах до методології проектування реляційних та об'єктно-орієнтованих БД.

Під час проектування баз даних розв'язуються два основних завдання:

1. Відображення об'єктів предметної області в абстрактні об'єкти моделі даних так, щоб це відображення не суперечило семантиці предметної області та було ефективним та зручним.

2. Забезпечення ефективності виконання запитів до БД (розташування даних у зовнішній пам'яті, створення додаткових структур тощо).

На якість баз даних (БД), які лежать в основі сучасних інформаційних систем (ІС), та особливості їх проектування впливає дуже багато факторів: сучасні об'єми даних, що зберігаються, вимоги до їх доступності та швидкості обробки, динаміка розвитку систем, які в свою чергу потребують ретельного дослідження.

Аналіз досліджень і публікацій. Стосовно класичних методів проектування, то на даний час на практиці застосовуються [4]:

- СУБД, які підтримують реляційну модель даних 1971 р. з деякими розширеннями [1];
- ієрархічна «каскадна» схема структурного проектування БД при підході «зверху-донизу»;
- CASE-системи для структурного проектування баз даних, ІС в цілому або, до того ж, прикладних програм ІС. Найбільш часто застосовуються: варіанти ER-моделі даних, таблична реляційна модель 1971 року, розширена

додатковим набором опису обмежень цілісності (посилальна цілісність, бізнес-правила); для аналізу «процесного» джерела відомостей частіше за все подаються моделі потоків даних або SADT, можливо, розширені додатковими зв'язками по управлінню (ці зв'язки неможна змішувати з виділеними потоками умов виконання функцій в нотації IDEF0);

- утилити динамічного адміністрування БД.

Опис класичних моделей та концепцій, а також їх обмежень розглянуто Є. Коддом у [18, 19]. Критику проектування по класичних правилам повноти та цілісності можна дослідити з роботи [15]. Причини виникнення нових вимог до корпоративних БД та, як наслідок, нові технічні вимоги до БД можна розглянути у дослідженнях авторів Є. Зіндер [5] та В. Меллінг [5, 15]. У своїй роботі [5] Є. Зіндер окреслив нові вимоги до корпоративних БД на прикладі двох аспектів створення нових корпоративних ІС; у роботі [4] йдеться про зміну підходів до вибору СУБД (в першу чергу для проектування корпоративних БД, експлуатація та розвиток яких планується як мінімум на декілька років), використання економічних обставин та критеріїв для вибору СУБД досліджував.

Перетворення каскадних схем організації проектування предметної області для ІС до циклічної форми розглянута Дж. Фоксом [16]. Так, організація розробки IBM corp. передбачала безперервний контрольований розвиток програмної системи у вигляді передачі в експлуатацію її нових версій.

З аналізу досліджень можна стверджувати, що автори виділяють підходи до проектування БД: проектування БД як об'єкта, який свідомо відділений від прикладних програм, та об'єктно-орієнтоване проектування, в якому об'єкт інкапсулює дані та методи їх обробки. Але не показана неефективність моноспрямованих (реляційних або об'єктно-орієнтованих) СУБД для побудови БД, що входять до складу сучасних інформаційних систем.

Постановка завдання. Метою статті є зробити порівняння у підходах до проектування реляційних та об'єктно-орієнтованих БД з аналізом їх ефективності для побудови БД, що входять до складу сучасних інформаційних систем, та описати аспекти роботи з даними в постреляційних СУБД.

Виклад основного матеріалу. Зазвичай життєвий цикл БД має певні етапи концепту-

ального і інфологічного проектування, розробки, впровадження, супроводу та розвитку. Розглянемо кожен з цих етапів.

На першому етапі – етапі концептуального проектування, здійснюється аналіз властивостей і характеристик предметної області, що досліджується та формування канонічних структур баз даних, які можуть відображатися у вигляді графів, де роль вузлів відіграють об'єкти предметної області, а дуг – співвідношення між ними. Для опису канонічної структури бази даних використовуються різні технології та інструментальні засоби, наприклад CA ERwin Data Modeler (CASE-продукт AllFusion Modeler Suite) і реалізована в ньому нотація IDEF1X [17], Rational Rose і реалізована в ньому нотація UML (Unified Modeling Language – уніфікована мова моделювання) [22]. UML дозволяє описати предметну область на найбільш природній мові SQL – класи, об'єкти і відношення між ними. Мова опису предметної області на даному етапі конче важлива: проектувальник робить аналіз і моделює її в обов'язковій взаємодії з користувачами, більшість з яких не є технічними фахівцями, тому для коректної інтерпретації моделей мова їх опису повинна бути простою і зрозумілою. На даному етапі процес моделювання відбувається без прив'язки до конкретної СУБД.

Обмеження обраної СУБД враховуються на наступному етапі, коли канонічна структура перетворюється в логічну структуру баз даних. Розглянемо специфіку побудови логічних структур для реляційних і об'єктно-орієнтованих СУБД.

Реляційні бази даних являють собою безліч двовимірних таблиць і відношень між ними, що задаються структурою зовнішніх ключів. Канонічна структура зазвичай складається зі складних об'єктів та зв'язків – міжоб'єктних і внутрішньооб'єктних (складні типи даних). Тому процес проектування логічних структур баз даних реляційного типу зводиться до наступної послідовності операцій: аналіз предметної області та виділення базових типів сутностей, нормалізація типів сутностей і формування логічних записів, створення відповідних зв'язків між записами [10].

Методи проектування реляційних баз даних мають формальну строгість, але їм притаманний ряд недоліків. Логічні структури БД

для інформаційних систем, що при побудові використовують велику кількість інформаційних одиниць, (зважаючи на значну кількість багатозначних залежностей між даними) можуть складатися з багатьох таблиць, що робить такі БД погано доступними для огляду і управління. Більше того, за рахунок декомпозиції об'єктів предметної області на плоскі нормалізовані відносини втрачається семантика досліджуваної предметної області, що ускладнює супровід і модернізацію систем. Дані методи не дозволяють адекватно моделювати окремі властивості даних [2].

Для адекватного моделювання складних структур даних проектувальник повинен мати можливість призначати власні типи даних, не обмежуючись тими даними, які надаються певною реляційною СУБД. Реляційна модель не дозволяє також визначити набір операцій, які пов'язані з даними певного типу, що часто є природною вимогою при моделюванні об'єктів предметної області. Операції доводиться задавати в конкретному додатку. Тому принципи використання таких методів проектування вимагають високої кваліфікації проектувальників.

Засадами об'єктно-орієнтованого підходу до моделювання предметних областей є поняття об'єкт, властивості спадкування, інкапсуляції та поліморфізму [12, 14].

Властивість спадкування надає можливість створювати з об'єктів нові об'єкти. Нові об'єкти успадковують структуру і поведінку своїх попередників, до яких додаються нові характеристики, що відображають їх індивідуальність.

Властивість інкапсуляції означає, що об'єкти мають деяку структуру і певний набір операцій, тобто поведінку. Внутрішня структура об'єкта прихована від користувача. Зміна стану об'єкта, маніпулювання об'єктом можливі лише за допомогою спеціальних методів, які визначаються заданим набором операцій.

Властивість поліморфізму означає, що різні об'єкти мають змогу отримувати однакові повідомлення, але реагувати на них по-різному – залежно від того, яким чином у них реалізовані методи реакції на повідомлення.

Природний перехід від концептуальної до логічної структури БД забезпечують об'єктно-орієнтовані технології. При проектуванні

об'єктно-орієнтованих БД (ООБД), на відміну від реляційних БД, не вимагається декомпозиція і нормалізація об'єктів, які були виділені на етапі концептуального проектування. Об'єкти подаються у попередньому вигляді, що надає об'єктно-орієнтованим структурам наочності та прозорості і дозволяє зменшити час на їх розробку і підвищити ступінь спадкування. ООБД встановлюють можливість створення і використання складних типів даних. При цьому не потрібно модифікації ядра ООБД і для створення нового типу потрібно успадкувати характеристики будь-якого наявного типу, який найбільш відповідний за своєю поведінкою і станом, розширити відсутні операції та атрибути і перевизначити вже наявні. Отримані об'єктно-орієнтовані структури володіють високим ступенем модульності, що дозволяє вносити до них зміни найбільш простим і безболісним способом. При цьому отримані зміни впливають на один клас (або пов'язану підсистему класів), ними можна ефективно керувати, їх можна перевіряти. Розвинені об'єктно-орієнтовані СУБД містять ефективні інтерфейси інтеграції з відомими інструментальними засобами проектування, що забезпечують автоматичну генерацію логічної структури та її завантаження в ООБД. Наприклад, СУБД Cache від компанії InterSystems містить інтерфейс Roselink, що забезпечує тісну інтеграцію з продуктом Rational Rose [6].

Розглянемо етап розробки. Всі запити користувачів до БД можна поділити на два класи – запити на модифікацію даних і запити на вибірку даних. На цьому етапі складно сказати, яка структура: об'єктна або реляційна, ефективніша. Проста, ефективна і стандартизована мова SQL забезпечує найбільш зручні на даний час механізми для вибірки та аналізу даних і значно перевершує за можливостями і зручності використання інші мови вибірки та аналізу даних. З іншого боку, об'єктно-орієнтовані БД за рахунок підтримки складних типів даних і відношень, механізмів зв'язлінгу і двоетапної фіксації даних надають більш розвинені, у порівнянні з реляційними БД, засоби для роботи з окремими записами в БД [20].

Завдання, пов'язані з масовим завантаженням, вивантаженням і обробкою даних, можна

віднести в окремий клас запитів користувачів. Як правило дані завдання здійснюються в ексклюзивному режимі і вимагають максимальної швидкості реалізації. Відомо, що максимальну швидкість під час роботи зі значними обсягами даних забезпечують ієрархічні бази даних. Можна зробити висновок, що на етапі розробки доцільно застосовувати відразу три способи роботи з даними [20]. Тому, як варіант, створювати різні БД під управлінням різних СУБД та регулярно їх синхронізувати, але, очевидно, це дуже коштовне та сумнівне рішення.

Етап розвитку системи – останній етап. Безумовним лідером по зручності і швидкості є об'єктний підхід, який за рахунок реалізації принципів поліморфізму і наслідування забезпечує найбільш простий і ефективний спосіб адаптації схеми БД в умовах змінюються вимог користувачів [12, 14].

Здійснений вище аналіз показав неефективність моноспрямованих СУБД (реляційних або об'єктно-орієнтованих СУБД) для побудови БД, які є складовими сучасних ІС. Так, на етапах проектування (концептуального і інфологічного), супроводу та розвитку доцільно застосовувати об'єктно-орієнтовані технології. На етапі розробки, для реалізації завдань вибірки та аналізу даних – мову SQL, для роботи з окремими записами в БД – об'єкти, для масової обробки даних – ієрархічні масиви.

У зв'язку з вище наведеним, все більш популярними стають гібридні або постреляційні СУБД, які реалізують відразу кілька моделей даних в рамках єдиного сховища даних, а найбільш розвинені – і в рамках єдиного сховища метаданих [14].

На ринку представлено декілька гібридних СУБД від різних постачальників. Умовно їх можна поділити на два класи [20]. До першого

класу належать реляційно-об'єктні (об'єктно-реляційні) СУБД, в яких є об'єктна або реляційна надбудова над відповідно реляційною або об'єктною моделлю даних. Зазначену модель активно пропонують постачальники реляційних СУБД. Другий клас – постреляційні СУБД. Вони не будуються ні на реляційній, ні на об'єктній моделі, однак також надають можливість представляти збережені дані у вигляді реляційних таблиць і класів об'єктів. До другого класу СУБД відноситься і СУБД Cache від компанії InterSystems [11].

Двом зазначеним типам гібридних систем властиві ненормалізована модель даних, інкапсульована семантика додатків і безліч зовнішніх інтерфейсів (об'єктних і реляційних). Розглянемо їх особливості.

Уникнути частини обмежень, які властиві ядру, дозволяє об'єктна або реляційна надбудова над існуючим ядром системи. Однак, у такому випадку виникає багаторівнева архітектура (рис. 1), що негативно позначається на продуктивності надбудов і робить важчою саме ядро системи. Така надбудова в більшості випадків обмежена і не відповідає стандартам на реалізацію моделі (SQL92, SQL99) або рекомендаціям комітетів зі стандартизації (ODMG) [20].

Ядро постреляційних СУБД не використовує ні реляційну, ні об'єктну схему – зазвичай його основа побудована на базі мережевої або ієрархічної моделі. Розглянемо доцільність цього. Як відомо, реалізації мережевої і особливо ієрархічної моделі БД різняться високою швидкістю роботи з даними і простотою масштабованості. При цьому гнучкість мовного середовища ієрархічних БД дозволяє досить ефективно втілювати ту чи іншу модель даних. Тому переважна більшість фахівців рекомендують використовувати ієрархічну СУБД в

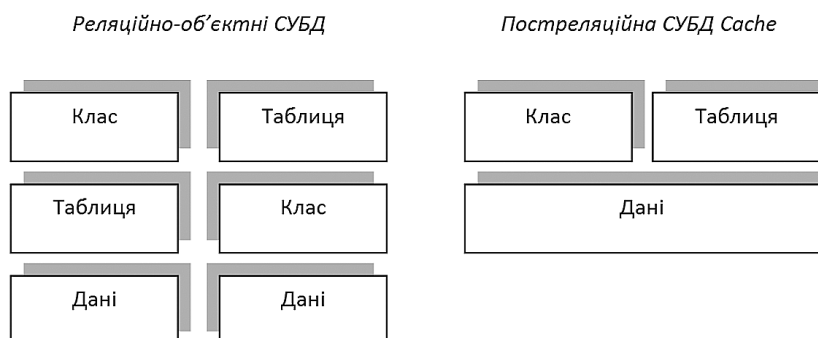


Рис. 1. Порівняння реляційно-об'єктних архітектур та постреляційної СУБД Cache

якості базової основи навіть для моноспрямованих реляційних і об'єктних СУБД. Для роботи з даними в постреляційних СУБД застосовуються механізми, які подають масиви даних ієрархічного або мережевого ядра системи у вигляді класів об'єктів і реляційних таблиць та забезпечують необхідні механізми (наприклад, інтерфейси до зовнішніх інструментарій або вбудовані мови третього покоління) для роботи з ними.

Висновки і перспективи подальших досліджень. У процесі вивчення студентами спеціальності «Інформатика» курсу «Методологія проектування сучасних баз даних» важливо усвідомити, що від правильного визначення придатної моделі даних, обґрунтування раціональної схеми побудови бази даних та основ проектування баз даних багато в чому залежить ефективність функціонування систем, які розробляються на їх основі. Все це потребує усвідомленого застосування методологій проектування сучасних баз даних. Адже, практичне створення та опрацювання баз даних застосовується у багатьох предметних сферах.

Запропоновані в статті аспекти проектування сучасних БД є основою питань з проектування баз даних і не вичерпують усіх змістових положень курсу «Методологія проектування сучасних баз даних», вони розкривають особливості у підходах до проектування реляційних та об'єктно-орієнтованих БД. Основна відмінність і перевага постреляційних СУБД у порівнянні з реляційно-об'єктними СУБД полягає в тому, що в постреляційних СУБД механізми роботи з об'єктами і реляційними таблицями знаходяться на одному логічному рівні, що забезпечує більш високу швидкість доступу і роботи з даними, функціональну повноту, в тому числі відповідність певним стандартам і специфікаціям.

В якості подальших питань щодо розширення викладання дисципліни «Методологія проектування сучасних баз даних» пропонується аналіз механізмів, які застосовуються для роботи з даними у постреляційних СУБД.

Список використаних джерел

1. Дадли К. Соответствие стандарту SQL / К. Дадли // Бюллетень «Мир Oracle». — 1996. — № 1 (39). — С. 7—16.
2. Дейт К. Дж. Введение в системы баз данных / К. Дж. Дейт. — М.: Вильямс, 2006. — 1328 с.

3. Зеленков Ю. А. Объектно-ориентированные СУБД. Введение в базы данных / Ю. А. Зеленков. — Центр Интернет ЯрГУ // [Электронный ресурс]. — Режим доступа: http://www.mstu.edu.ru/study/materials/zelenkov/ch_6_3.html.
4. Зиндер Е. З. Критерии выбора современной СУБД как объекта инвестиций для развития предприятия / Е. З. Зиндер // СУБД. — 1995. — № 1 [Электронный журнал]. — Режим доступа: <http://www.osp.ru/dbms/1995/01/13031398/>.
5. Зиндер Е. З. Новое системное проектирование: информационные технологии и бизнес-реинжиниринг (часть 2) / Е. З. Зиндер // СУБД. — 1996. — № 1 [Электронный журнал]. — Режим доступа: <http://www.osp.ru/dbms/1996/01/13031460/>.
6. Кирстен В. СУБД Cache: объектно-ориентированная разработка приложений / В. Кирстен, И. Иренгер, Б. Рериг, П. Шульте. — СПб.: Питер, 2001. — 247 с.
7. Когаловский М. Р. Перспективные технологии информационных систем / М. Р. Когаловский. — М.: ДМК Пресс; Компания АйТи, 2003. — 288 с.
8. Когаловский М. Р. Энциклопедия технологий баз данных / М. Р. Когаловский. — М.: Финансы и статистика, 2002. — 800 с.
9. Коннолли Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг. — М.: Вильямс, 2003. — 1436 с.
10. Коннолли Т. Базы данных: проектирование, реализация и сопровождение. Теория и практика: Уч. пос. / Т. Коннолли, К. Бегг, А. Страчан. — М.: Изд. дом «Вильямс», 2000. — 1120 с.
11. Кречетов Н. Постреляционная технология Cache' для реализации объектных приложений / Н. Кречетов, Е. Петухова, В. Скворцов, А. Умников, Б. Щукин. — М.: МИФИ, 2001. — 152 с.
12. Кузнецов С. Д. Объектно-ориентированные базы данных — основные концепции, организация и управление: краткий обзор / С. Д. Кузнецов // [Электронный ресурс]. — Режим доступа: http://citforum.ru/database/articles/art_24.shtml.
13. Кузнецов С. Д. Основы баз данных / С. Д. Кузнецов. — М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. — 484 с.
14. Кузнецов С. Д. Основы современных баз даны / С. Д. Кузнецов [Электронный ресурс]. — Режим доступа: <http://www.citforum.ru/database/osbd/contents.shtml>.
15. Меллинг В. П. Корпоративные информационные архитектуры: и все-таки они меняются / В. П. Меллинг // СУБД. — 1995. — № 2 [Электронный журнал]. — Режим доступа: <http://www.osp.ru/dbms/1995/02/13031425/>.
16. Фокс Дж. Программное обеспечение и его разработка / Дж. Фокс. — М.: МИР, 1985. — 386 с.
17. CA technologies // Официальный сайт компании CA [Электронный ресурс]. — Режим доступа: <http://www.ca.com/Solutions/>.
18. Codd E. F. Extending the Database Relational Model to Capture More Meaning / E. F. Codd // ACM Transactions on Database Systems (TODS). — 1979. — № 4. — P. 397—434.
19. Codd E. F. Providing OLAP to User-Analyst: An IT Mandate / E. F. Codd, S. B. Codd, C. T. Salley. — E. F. Codd & Associates, 1993 // [Electronic resource]. — Режим доступа: http://www.minet.uni-jena.de/dbis/lehre/ss2005/sem_dwh/lit/Cod93.pdf.
20. ComputerWorld (Мир компьютера). Электронный журнал. Материалы сайту [Электронный ресурс]. — Режим доступа: www.computer-world.com.ua.

21. InterSystems Caché. Відомості з офіційного сайту InterSystem Corp. Россия (<http://www.intersystems.ru/index.html>) // [Електронний ресурс]. — Режим доступу: <http://www.intersystems.ru/cache/index.html>.

22. Rational Rose family // Відомості з офіційного сайту компанії IBM (<http://www.ibm.com/us/en/>) [Електронний ресурс]. — Режим доступу: <http://www-03.ibm.com/software/products/us/en/ratirosefami/>.

GANNA POGROMS'KA

Mykolayiv

THE KEY ASPECTS OF MODERN DESIGN METHODOLOGY OF DATABASES

The article covers the basic theoretical and practical issues that arise during the course «Methodology of designing modern databases». A comparative analysis of approaches to the design of relational and object-oriented database and demonstrated ineffectiveness monodirected database management systems (DBMS) to build databases that are part of modern information systems.

Keywords: database, database management systems, the information system, the stages of design, the relational DBMS, the object-oriented DBMS, the post-relational DBMS, monodirected DBMS.

А. С. ПОГРОМСКАЯ

г. Николаев

КЛЮЧЕВЫЕ АСПЕКТЫ МЕТОДОЛОГИИ ПРОЕКТИРОВАНИЯ СОВРЕМЕННЫХ БАЗ ДАННЫХ

В статье раскрыты базовые теоретические вопросы, возникающие во время преподавания курса «Методология проектирования современных баз данных». Выполнен сравнительный анализ в подходах к проектированию реляционных и объектно-ориентированных БД и продемонстрирована неэффективность мононаправленных систем управления базами данных (СУБД) для построения БД, входящих в состав современных информационных систем.

Ключевые слова: базы данных (БД), система управления базами данных (СУБД), информационная система (ИС), этапы проектирования, реляционные СУБД, объектно-ориентированные СУБД, построительные СУБД, мононаправленные СУБД.

Стаття надійшла до редколегії 18. 03. 14

УДК 378.147

Н. О. ПРАСОЛ

м. Николаїв

ІНДИВІДУАЛЬНИЙ ГРАФІК НАВЧАННЯ ЯК УМОВА ПЛАНУВАННЯ ІНДИВІДУАЛЬНОЇ ТРАЄКТОРІЇ РОЗВИТКУ

У статті аналізується індивідуальний графік навчання як одна з умов індивідуалізації професійної підготовки майбутніх учителів початкових класів в умовах кредитно-трансферної системи навчального процесу.

Ключові слова: індивідуальний графік, індивідуалізація професійної підготовки, кредитно-трансферна система навчання, майбутній учитель початкових класів.

Постановка проблеми. Головним завданням вищої педагогічної школи на сьогоднішньому етапі модернізації вищої освіти України є забезпечення якості підготовки спеціалістів на рівні міжнародних стандартів. Розв'язання цього завдання можливе при створенні сприятливих умов для реалізації майбутніми учителями свого інтелектуального потенціалу на основі впровадження у процес професійної підготовки інноваційних педагогічних технологій, в першу чергу технології індивідуалізації, суттєвого підвищення ефективності навчального

процесу, безперервної активності і викладача, і студента [1, с. 217].

У контексті глобалізаційних змін актуальною постає проблема професійно-педагогічної підготовки вчителя початкових класів в умовах кредитно-трансферної системи навчання з урахуванням індивідуалізації, яка передбачає подальше розроблення та використання особистісно-орієнтованих технологій навчання, створення індивідуальних програм розвитку особистості майбутнього вчителя. Саме така система сприяє розвитку