Scientific journal
PHYSICAL AND MATHEMATICAL EDUCATION
Has been issued since 2013.

Науковий журнал
ФІЗИКО-МАТЕМАТИЧНА ОСВІТА
Видається з 2013.

ISSN 2413-158X (online)
ISSN 2413-1571 (print)

http://fmo-journal.fizmatsspu.sumy.ua/

Шамшина Н.В. Методичні особливості вивчення зв'язків та типів об'єднання у базах даних Microsoft Access. Фізико-математична освіта. 2018. Випуск 1(15). С. 339-343.

Shamshina N. Methodical Features Of Studying Relationships And Types Of Joins In Databases Microsoft Access. Physical and Mathematical Education. 2018. Issue 1(15). P. 339-343.

УДК 004.657+372.8

Н.В. Шамшина
Сумський державний педагогічний університет імені А.С. Макаренка, Україна
shamichek@ukr.net
DOI 10.31110/2413-1571-2018-015-1-065

МЕТОДИЧНІ ОСОБЛИВОСТІ ВИВЧЕННЯ ЗВ'ЯЗКІВ ТА ТИПІВ ОБ'ЄДНАННЯ У БАЗАХ ДАНИХ MICROSOFT ACCESS

Анотація. *Стаття присвячена вивченню зв'язків між таблицями та типів об'єднання у реляційній базі даних. Автор розглядає складності у розумінні окремих понять та теоретичних положень, з якими стикаються початківці при вивченні системи управління базами даних Access на практиці. Питання організації зв'язків між таблицями є принциповим для розуміння роботи реляційної бази даних. Вміння створювати зв'язкі, налаштовувати їх властивості, забезпечувати цілісність даних формуються під час розв'язування практичних завдань. Автор доступно и детально пояснює механізм зв'язку, описує типові помилки учнів, формулює правила та алгоритми створення різних типів зв'язку. Особлива увага приділяється типам об'єднання, які підтримуються у режимі Конструктора запитів. В статті описано внутрішні об'єднання – одно-стовпцеві та багато-стовпцеві, зовнішні об'єднання – ліве та праве, само-об'єднання, тета-об'єднання за умовою нерівності, перехресні об'єднання. Об'єднанням називається операція, під час якої виконується зіставлення та поєднання значень у спільних полях зв'язаних таблиць, які є джерелом даних у формі, звіті, або запиті. Тип об'єднання задає спосіб перегляду зв'язаних записів на основі заданого відношення. Кількість записів, які відображаються, залежить від типу об'єднання. Підґрунтям більшості об'єднань (внутрішніх, зовнішніх, само-об'єднань) є рівні значення полів зв'язку. Тета-об'єднання створюють за умовою нерівності значень полів. Перехресне об'єднання (декартів добуток) використовують для генерації комбінацій записів у запиті. Об'єднання, яке створене між полями з невизначеним типом відношення, може привести до помилкових записів тому, що повторення значень в обох полях зв'язку призводить до появи всіх можливих комбінацій співставлення записів. Автором наведено приклади використання різних типів об'єднань при побудові запитів для розв'язування практичних завдань. Матеріал статті містить роз'яснення та методичні рекомендації для вивчення теми «Системи Управління Базами Даних» з дисципліни Інформаційні технології.*

Ключові слова: *методика вивчення, Access, бази даних, між-табличні зв'язки, типи об'єднання.*

Постановка проблеми. Сучасні бази даних (БД) та системи управління базами даних (СУБД) вивчаються на різних рівнях системи освіти. Знайомство з реляційними базами даних відбувається ще в старших класах середньої школи на прикладі СУБД Access. Більш детальний і поглиблений їх розгляд відбувається на старших курсах різних спеціальностей у ВНЗ.

Аналізуючи методичні особливості вивчення теми дослідники виділяють складності у розумінні окремих понять та теоретичних положень, з якими стикаються початківці при вивченні СУБД Access на практиці [3, 4]. Одне з таких питань – питання організації зв'язків між таблицями, що є принциповим для розуміння роботи реляційної бази даних. Проте, у методичних рекомендаціях та підручниках практично не розглядається питання використання різних типів об'єднання записів при побудові запитів. Між тим, подібні навички необхідні учням при розв'язуванні олімпіадних завдань з інформаційних технологій (ІТ).

Аналіз актуальних досліджень. У методичних джерелах традиційно приділяють увагу вивченню типів зв'язків у реляційних БД. Найбільш відомим та популярним є підручник Завадського І. «Основи баз даних» [3]. У навчальному посібнику автор на поглибленому рівні розглядає теорію зв'язків у БД. Значна увага приділяється семантичному моделюванню БД, побудові моделей «сутність-зв'язок». Крим того, чи не вперше у шкільній навчальній літературі пропонується вивчення основ мови SQL. Адже в основу всіх реляційних СУБД покладено насамперед SQL, а майстри та конструктори в СУБД Access – це вторинні візуальні засоби [3, с.4].

Пошук відомостей щодо типів об'єднання в СУБД Access, як правило, приводить до веб-сайтів, які присвячені вивченню структурованій мові запитів SQL та її синтаксису. Загальною є думка, що без опанування мови SQL, не можливо зрозуміти та навчитися застосовувати на практиці різні типи об'єднань у запитах Access. Мова SQL більш універсальна, та дозволяє створити такі типи об'єднань записів, які не підтримуються у Конструкторі запитів.

Базові концепції про типи об'єднань, що можна використовувати в режимі Конструктора запитів, викладено на сторінках сайту «Office Support-Office 365». Довідка на сайті розробника програмного продукту є автоматичним перекладом англомовного джерела. Досвід користування довідкою показує, що в україномовному перекладі помилок ще більше, ніж у російськомовному. Краще користуватися англомовним оригіналом статті [1]. Отже, маємо практично відсутність методичних розробок для вивчення типів об'єднання при побудові запитів у режимі Конструктора.

Мета статті – надати методичні рекомендації по вивченню зв'язків між таблицями у реляційній базі даних, описати різні типи об'єднання записів, які підтримуються у Конструкторі запитів, та навести приклади їх використання для розв'язування практичних завдань.

Виклад основного матеріалу. Питання створення зв'язків між таблицями часто сприймається учнями поверхнево. Не розуміючи окремих моментів: доцільності зв'язку, механізму зв'язку, перевірки типу зв'язку, типу об'єднання, забезпечення цілісності даних, вони можуть на практиці зв'язати одне поле з іншим, не задумуючись про тип поля и про дані, які воно зберігає. Щоб завадити цьому потрібно:

по-перше, обґрунтувати необхідність створення декількох таблиць та зв'язків між ними для різних типів відношень;

по-друге, роз'яснити механізм зв'язку, який може здійснювати програма: за однаковими значеннями полів зв'язку; вочевидь, що тип та підтип полів має бути однаковим;

по-третє, довести, що від того, як таблиця зв'язана з іншими, залежить яке з її полів слід призначити первинним ключем.

У теорії СУБД відомі чотири варіанти зв'язків, які традиційно називають відношеннями (англ. *relationship*) [2, с. 117]. Вони встановлюють правила відповідності записів в двох таблицях, причому «розглядаємо» як би з боку першої таблиці, яка є «головною», а друга – «підлеглою», «зв'язаною». Записи в головну таблицю додають раніше ніж у підлеглу. Дамо коротку характеристику цім відношенням з погляду практичного застосування.

Відношення «один-до-одного»: одному запису першої таблиці може відповідати не більше одного запису другої таблиці. Це означає, що в підлеглій таблиці може і не бути відповідного запису. В обох полях не повинно бути значень, що повторюються. Поле – первинний ключ першої таблиці зв'язано з таким самим полем другої таблиці, яке, найчастіше, також є первинним ключем. Дві таблиці, які зв'язані «один-до-одного», це як би одна загальна таблиця, яка розділена надвоє вертикальною межею, так що частина полів опинилася в першій таблиці, а частина в другій. Розділення йде за функціональною ознакою, з метою введення даних на окремих робочих місцях.

Відношення «один-до-багатьох»: в цьому випадку одному запису першої таблиці може відповідати декілька записів другої таблиці. Це означає, що в підлеглій таблиці також може бути один відповідний запис, а може й ні одного. Поле – первинний ключ першої таблиці зв'язано з таким самим полем в другій таблиці, яке є зовнішнім ключем. Значення по зовнішньому ключу можуть повторюватися. Цей найбільш поширений тип зв'язку використовується задля усунення багатьох повторень у таблиці.

Відношення «багато-до-одного»: декільком записам першої таблиці може відповідати не більше одного запису з другої таблиці. Це відношення «один-до-багатьох» навпаки, головною тут є таблиця з боку багатьох записів.

Відношення «багато-до-багатьох»: декільком записам першої таблиці може відповідати декілька записів другої таблиці. В цьому випадку зв'язки між таблицями непрямі, організовані за допомогою ще однієї таблиці з використанням складного ключа, тобто набору ключових полів.

Аналіз теоретичних положень та порівняння їх з практичною реалізацією в Access призводить до висновку, що двох перших типів відношення достатньо для побудови структури будь якої БД в Access. Треба лише звертати увагу на те, яка таблиця має бути головною, а яка підлеглою, зв'язаною.

Від типу відношення зв'язку таблиці з іншими залежить яке з її полів слід призначити первинним ключем. Для поля-первинного ключа унікальний індекс встановлюється автоматично. Якщо обидва поля зв'язку мають унікальний індекс буде створено зв'язок «один-до-одного». Щоб встановити унікальний індекс потрібно для властивості *Индексированное поле* вказати значення *Да (Совпадения не допускаются).* Коли індекс одного поля унікальний, а іншого – ні, буде створено зв'язок «один-до-багатьох». У поля на стороні зв'язку «один» (якщо це не первинний ключ) треба встановити унікальний індекс. Поле на стороні «багато» повинно мати індекс, який може повторюватися. Тобто для властивості *Индексированное поле* має бути задано значення *Нет* або *Да (Совпадения допускаются).*

Зв'язкі між таблицями можуть бути практично реалізовані завдяки наявності в них загальних полів з однаковими значеннями. Необхідно, щоб поля двох таблиць, по яких вони зв'язуються між собою, мали однакові значення, однаковий тип і підтип даних, імена можуть і не співпадати. Зверніть увагу на те, що поле типу *«Счетчик»* зберігає дані так саме, як поле типу «*Число*», якщо його властивість *Размер поля* має значення *Длинное целое*. Значення цих полів можна порівнювати та по ним зв'язувати таблиці. Коли загальних полів в таблицях немає, потрібно зробити наступне:

- якщо між реляційними таблицями існує відношення «один-до-одного» або «один-до-багатьох», то слід скопіювати поле, по якому встановлюється зв'язок, з головної таблиці в підлеглу;
- якщо між таблицями існує відношення «багато-до-багатьох», то слід створити нову таблицю (таблицю асоціацій, таблицю зв'язку) і включити в неї ключові поля обох таблиць.

Зв'язки створюються та редагуються у вікні *РАБОТА С БАЗАМИ ДАННЫХ / Схема данных*. Для виконання операцій користуються контекстними меню області вікна, таблиць і ліній зв'язку, а також командами на стрічці *РОБОТА СО СВЯЗЯМИ / КОНСТРУКТОР*. На кожну операцію, як правило, існує 2-3 способи виконання. Наприклад, додати таблиці на схему можна перетягуванням з бокової панелі об'єктів Access, використанням контекстного меню схеми даних, кнопкою на панелі інструментів. Головне зрозуміти, що означає та чи інша операція. Команда *Очистить макет* не видаляє зв'язки, а лише знімає їх відображення у вікні схеми даних. Команда *Все связи* відображає схему даних знову. Команда *Отчет по схеме данных* дозволяє роздрукувати схему. Найбільш зручним способом створення зв'язку вважається перетягування поля зв'язку з однієї таблиці в іншу на відповідне поле, у напрямку з головної таблиці у підлеглу. Типовою помилкою є перетягування поля зліва-направо для будь якої пари таблиць. Це призводить до неправильного визначення головної та підлеглої таблиці й неможливості додавати записи у потрібну таблицю.

При створенні зв’язків у вікні Схеми даних користувач іноді стикається з неприємними «сюрпризами»: блокування створення або редагування зв’язків, неможливість вибору типу зв'язку, неможливість встановити зв'язок із забезпеченням цілісності даних. Необхідно надати учням простий алгоритм створення зв’язків, та пояснити можливі помилки: блокування трапляється для відкритої у БД таблиці; тип зв’язку визначається програмою за властивостями полів зв’язку – ключові, індексовані, можливі чи ні повторення значень; забезпечення цілісності даних можливо при однакових типах, підтипах, та узгодженості даних полів зв’язку.

Таким чином, перед початком роботи зі Схемою даних бажано закрити усі відкриті таблиці та інші об’єкти БД, тому що форми, звіти, запити базуються на таблицях. При створенні зв’язку слід ретельно перетягувати ключове поле з головної таблиці на поле зв’язку підлеглої таблиці. Бажано, щоб короткі імена полів у таблицях відповідали даним, які в них зберігаються, наприклад, КодКлиєнта, КодКниги, КодЗапису. Це додає зручності при зв’язуванні відповідних полів. У вікні зв'язку треба переконатися в правильності створеного відношення, якщо воно *НЕ ОПРЕДЕЛЕНО*, відредагувати властивості полів зв’язку. Модифікація структури заповненої таблиці, зміна властивостей полів часто потребує видалення зв’язків таблиці з іншими. Забезпечення цілісності даних не вдається підключити, якщо цілісність даних порушена, дані не узгоджено. В цьому випадку необхідно змінити дані.

Здатність Access підтримувати цілісність даних надзвичайно важлива. Підключення параметру забезпечення цілісності даних є обов’язковим для робочих БД. Це дозволяє уникнути записів-«сиріт», які не мають з’язків з головною таблицею, та можуть привести до неправильної інформації. Розглянемо приклад: припустимо, що у підлеглої таблиці є запис-«сирота» з кодом 10; якщо у головну таблицю буде занесений запис з кодом 10, одразу ця запис-«сирота» автоматично буде прив’язана до неї; як наслідок, в БД буде зберігатися неправильна інформація. При наявності записів-«сиріт» створення зв’язку з забезпеченням цілісності даних неможливе.

При підключеному параметрі цілісності даних операції з даними у зв’язаних таблицях контролюються. Якщо спробувати в підлеглу таблицю додати дані про об'єкт, який не має відповідного запису у головній таблиці, в збереженні даних буде відмовлено. При спробі видалення запису з головної таблиці за наявності зв'язаних з нею записів буде відмовлено. Не можна змінювати значення первинного ключа в головній таблиці, якщо це призведе до появи відірваних записів, в операції буде відмовлено. Для уточнення та для підтримки цілісності даних можна за необхідності підключити ще два параметра: каскадне оновлення зв'язаних полів та каскадне видалення зв'язаних записів. Перший автоматизує зміни значень зв’язаних полів для підлеглої таблиці, другий дозволяє автоматично вилучити зв’язані записи в підлеглій таблиці.

Слід зазначити, що цілісність бази даних є формальною властивістю. Access «не розуміє» сенс і логіку даних, що зберігаються. Програма лише враховує увесь набір обмежень цілісності. Якщо усі обмеження виконані, вважається, що дані коректні [4]. Так, наприклад, для числових даних, можна помилково створити зв'язок для різних за сенсом даних, якщо даних мало, або взагалі немає, тобто вони неначе «узгоджені».

Зв’язки, які встановлено на Схемі даних між базовими таблицями або запитами, та їх властивості, діють за замовчанням у наступних операціях опрацювання та виведення даних: при створенні форм, звітів, запитів. Важливу роль при цьому грає ще один параметр – тип об'єднання. Об’єднанням (англ. *join*) називається операція, під час якої виконується зіставлення та поєднання значень у спільних полях зв’язаних таблиць, які є джерелом даних у формі, звіті, або запиті. Тип об’єднання можна вказати через контекстне меню лінії зв’язку.

Тип об'єднання задає спосіб перегляду зв'язаних записів на основі заданого відношення. Для перегляду даних одночасно двох таблиць у формі, звіті, запиті, вибирають один з трьох типів об'єднання:

- рівне або внутрішнє об'єднання – об'єднання тільки тих записів, в яких зв'язані поля обох таблиць співпадають;
- зовнішнє ліве об'єднання – об'єднання ВСІХ записів з першої таблиці і лише тих записів з другої таблиці, в яких зв'язані поля співпадають;
- зовнішнє праве об'єднання – об'єднання ВСІХ записів з другої таблиці і лише тих записів з першої таблиці, в яких зв'язані поля співпадають.

Внутрішні об'єднання відображають записи в об’єднаних полях як один запис. Якщо не знайдено відповідного значення у полі зв’язку зв’язаної таблиці, дані не відображаються взагалі. Таким чином, кількість записів, які може бачити користувач у запиті, формі, звіті, дорівнює кількості записів у зв’язаній таблиці. Для таблиць, які зв’язані відношенням «один-до-багатьох», буде видно повторювання частини запису з боку таблиці «один». Це найбільш поширений тип об'єднання, встановлений за замовчанням.

Зовнішні об'єднання дозволяють відображати поля усіх записів таблиці незалежно від існування зв'язаних записів в об'єднаній таблиці. Таким чином кількість записів більше, ніж при рівному внутрішньому об'єднанні. Застосовують ліві і праві зовнішні об'єднання. Назви «ліве» та «праве» виникли відповідно до традиції розміщувати головну таблицю на Схемі даних зліва, а зв’язану, об’єднану – праворуч [2, с. 266]. Тому при зовнішньому лівому об’єднанні відображаються всі записи головної таблиці та поєднані з ними: записи-пари и записи без пари. Зовнішнє ліве об’єднання використовують для зручного додавання записів у підлеглій таблиці. При зовнішньому правому об’єднанні відображаються всі записи підлеглої таблиці та поєднані з ними з головної таблиці: пари записів і «сироти». Зовнішнє праве об'єднання використовують для пошуку записів-«сиріт» та виправлення помилок.

Треба відмітити, що можливість вибору типу об’єднання записів двох зв’язаних таблиць у вікні зв’язку на Схемі даних використовується мало, найчастіше потрібність змінити тип об’єднання виникає при конструюванні запиту. При конструюванні запиту, слід уважно вказувати джерела даних для результуючого набору записів та враховувати наступне:

– якщо додати у Конструктор запитів таблиці, які безпосередньо зв’язані на Схемі даних, автоматично діє існуючий зв'язок; якщо таблиці не зв’язано безпосередньо, потрібно додати весь ланцюжок зв’язаних таблиць в якості джерела даних;

– якщо додати у Конструктор запитів не зв’язані таблиці – отримаємо декартів добуток, або перехресне об’єднання, при якому кожному запису першої таблиці з N1 записами підставляється кожний запис другої таблиці з N2 записами, загальна кількість отриманих записів у запиті дорівнює N1xN2.

У режимі Конструктору запитів підтримується 5 видів об’єднання, які поділяються на типи:

внутрішні об'єднання – одно-стовпцеві та багато-стовпцеві;

зовнішні об'єднання – ліве та праве;
само-об’єднання;
тета-об'єднання за умовою нерівності
перехресне об’єднання.

Об'єднання двох таблиць, яке ґрунтується на зв'язку одного стовпця в кожній таблиці, називають одно-стовпцевім внутрішнім об'єднанням. Між двома таблицями можна створити декілька об’єднань, перетягнувши поле з однієї таблиці на поле іншої у Конструкторі запитів, отримаємо багато-стовпцевє внутрішнє об’єднання. Використовується для відбору лише тих записів, значення об’єднаних полів в яких співпадають (рис. 1).

Приклад на рис. 1 показує запит у Конструкторі з багато-стовпцевим внутрішнім об’єднанням. Запит відображає записи тих клієнтів готелю, які займали один й той же номер кімнати протягом двох сезонів. Дані витягуються з таблиці «Гості» та «Гості-Прош» за попередній сезон. Точки на кінцях лінії зв'язку означають, що об'єднання створене між полями з невизначеним типом відношення.

Приклад на рис. 2 показує запит у Конструкторі, що відображає назви електростанцій з таблиці «Електростанції», для яких відсутні дані про виготовлення ними електроенергії у таблиці «Обсяги виготовлення». Використане зовнішнє ліве об'єднання, щоб відобразити всі записи з головної таблиці «Електростанції», та вказана умова Is Null (немає даних) для поля «Обсяги» з підлеглої таблиці. Вигляд лінії зв’язку відповідає типу відношення «один-до-багатьох» та підключеному параметру «забезпечення цілісності даних», «стрілка» вказує на зовнішнє об’єднання.

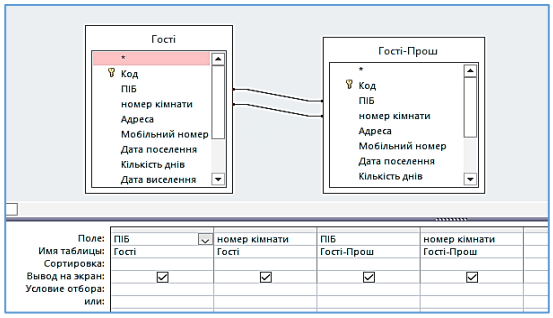

Рис. 1. Багато-стовпцевє внутрішнє об’єднання

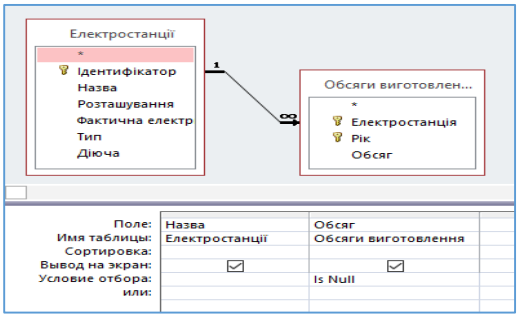

Рис. 2. Зовнішнє ліве об'єднання

Само-об’єднання зв'язують поля однієї таблиці (як правило різні поля) та відображають лише ти записи, в яких значення цих полів співпадають. Для створення само-об’єднання в Конструкторі запитів таблицю додають двічі, дублікату автоматично призначається псевдонім. Потім вказують об'єднання полів, які зв’язують шляхом перетягування, тип полів має бути однаковим. Само-об’єднання застосовують частіше у запитах на створення нової таблиці. У запитах на вибірку більш надійно використовувати умову на значення.

Підґрунтям більшості об’єднань (внутрішніх, зовнішніх, само-об’єднань) є рівні значення полів зв’язку, їх називають екві-об’єднання. Проте, іноді необхідно створити об'єднання за умовою нерівності значень полів, які називають тета-об'єднання. Тета-об'єднання зв'язують дані за допомогою операторів порівняння, відмінних від оператора рівності. Тета-об'єднання не відображаються лініями і застосовуються у запитах для відбору записів, яким потрібно відношення особливого типу. Якщо в стовпець бланка запиту включити умову нерівності значень полів – одержимо тета-об'єднання.

Перехресне об’єднання теж не відображається лініями, застосовується у запитах для генерації комбінацій записів двох таблиць. Взагалі, якщо немає однозначності у співставленні записів (не визначено поля зв’язку або зустрічаються повторення значень в обох полях звязку) – результуючий набір запиту має всі можливі комбінації. Цей факт використовують при розв’язуванні завдань, в яких треба на основі даних таблиці сформувати нову таблицю з більшою кількістю записів.

Розглянемо приклади в яких потрібно сконструювати за допомогою запиту набір записів, який є комбінуванням значень одного поля заданої таблиці. На практиці це може бути назви спортивних команд які змагаються між собою, або ж назви країн, які торгують, тощо.

Приклад на рис. 3 показує запит у Конструкторі, де використане перехресне об’єднання та тета-об'єднання за умовою нерівності. Запит генерує набір комбінацій назв країн з таблиці «Країни», яка має 7 записів. Спочатку додаємо таблицю «Країни» двічі у вікно Конструктору запитів, псевдонім копії таблиці «Країни_1». Далі у бланку запиту вказуємо поля для виведення: *Країни.назва*, *Країни_1.назва.* Отримаємо декартів добуток 7х7=49 записів. Серед комбінацій є 7 зайвих записів, коли одна и та же назва комбінується (наприклад: «Україна-Україна», «Білорусь-Білорусь»). Зайвими є також, у даному випадку, записі-повтори комбінацій, в яких назви країн переставлено (наприклад: «Україна-Білорусь», «Білорусь-Україна»). Необхідно позбавитися зайвих записів, для чого у стовпці поля *Країни.назва* треба вказати вираз для умови: *>[Країни_1]![назва].* Отримаємо 21 запис комбінацій назв країн які не повторюються.

На рис. 4 приведений приклад використання Само-об’єднання при конструюванні запиту з набором комбінацій назв спортивних команд, який є розв’язанням наступного завдання олімпіади з ІТ: «створіть запит, який на основі поданої у таблиці «Команди» інформації про футбольні команди-учасниці генерує усі матчі групового етапу; в груповому етапі кожна команда повинна двічі зіграти з кожною іншою командою своєї групи, причому один раз вдома, а один – на виїзді; кожна група складається з чотирьох команд». В таблиці «Команди» маємо 48 назв команд з 12 груп від A до L з різних країн, по 4 команди у групі. Створюємо само-об’єднання по полю *Група*, отримаємо 16 (4х4) комбінацій назв команд у межах кожної групи, разом 16х12=192. За допомогою умови нерівності назв позбавляємось комбінацій з однакових назв (їх по 4 у кожній групі, 4х12=48). Отримаємо 144 (192-48) комбінації результуючого набору запиту по 12 матчів групового етапу для 12 груп. Цей приклад ілюструє той факт, що при наявності повторень у значеннях полів зв’язку в обох таблицях результуючий набор записів у запиті буде містити всі можливі комбінації їх значень.

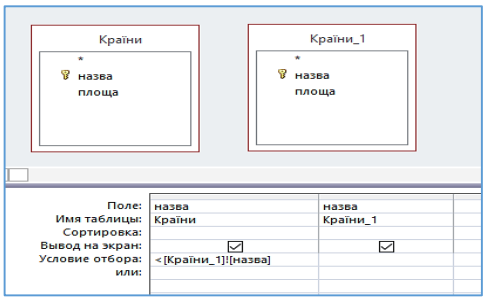

Рис. 3. Тета-об'єднання за умовою нерівності

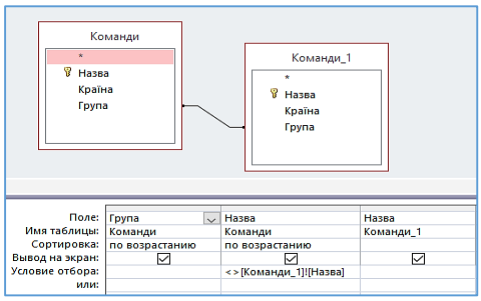

Рис. 4. Само-об'єднання

Висновки. Розгляд різних типів об'єднання та їх особливостей дає підстави зробить наступні висновки відносно їх використання у запитах. У Конструкторі запитів можна використовувати внутрішні і зовнішні об'єднання, само-об'єднання, перехресне об'єднання, тета об'єднання за умовою нерівності:

- внутрішні та зовнішні об'єднання, як правило, використовують у запитах на вибірку
- само-об'єднання застосовують частіше у запитах на створення нової таблиці
- перехресне об'єднання використовують для генерації комбінацій записів
- тета-об'єднання застосовують для особливих типів відношень за умови нерівності
- об'єднання, яке створене між полями з невизначеним типом відношення, може привести до помилкових записів
- у більшості випадків потрібний результуючий набір у запитах на вибірку можна отримати за допомогою умови на значення полів.

Наведений опис і приклади різних типів об'єднань при конструюванні запитів ілюструють підвищений рівень складності матеріалу при вивченні зв'язків та типів об'єднання у базах даних Access. Лише на практиці можливе засвоєння на поглибленому рівні відомостей, які необхідні при виконанні практичних завдань олімпіадного рівня. Вивчення типів об'єднання розвиває логічне мислення, допомагає засвоїти та закріпити навички конструювання запитів, привчає до контролювання можливих помилок у результуючому наборі записів, мотивує до нешаблонного креативного підходу при розв'язуванні практичних завдань.

Список використаних джерел

1. Join tables and queries. *Access* : веб-сайт. URL: https://support.office.com/en-us/article/join-tables-and-queries-3f5838bd-24a0-4832-9bc1-07061a1478f6?ui=en-US&rs=en-US&ad=US (дата звернення: 6.02.2018).
2. Дженнингс Р. Использование Microsoft Access 97 : спец. изд. Пер. с англ. Изд. 2-е, К., М., СПб : Издат. дом «Вильямс», 1998. 944 с.
3. Завадський І.О. Основи баз даних : навч. посіб. К. : Видавець І.О. Завадський, 2011. 192 с.
4. Шамшина Н.В. Об особенностях сохранения информации в базах данных. *Фізико-математична освіта : науковий журнал*. 2016. Вип. 4(10). С. 148-151.

References

1. Join tables and queries. *Access*: website. URL: https://support.office.com/en-us/article/join-tables-and-queries-3f5838bd-24a0-4832-9bc1-07061a1478f6?ui=en-US&rs=en-US&ad=US (date use: 6.02.2018) (in English)
2. Jennings R. Using Microsoft Access 97: spec. ed. Trans. with English. Ed. 2-d, – K., M., St. Petersburg: Publication house "Williams", 1998. 944 p. (in Russian)
3. Zavadsky I.O. Fundamentals of the databases: textbook – K. : Publ. I.O. Zavadsky, 2011. 192 p. (in Ukrainian)
4. Shamshina N.V. On the features of storing information in databases. *Physical and Mathematical Education: scientific journal*. 2016. Issue 4 (10). P. 148-151. (in Russian)

METHODICAL FEATURES OF STUDYING RELATIONSHIPS AND TYPES OF JOINS IN DATABASES MICROSOFT ACCESS

Natalia Shamshina

Makarenko Sumy State Pedagogical University

Abstract. *The article is devoted to the study relationships between tables and types of joins in a relational database. The author considers difficulties in understanding certain concepts and theoretical principles with which the beginners faces when studying database management system Access in practice. The question of creating relationships between tables is fundamental to understanding the operation of a relational database. Skill to create relationships, specify their properties, ensure integrity of data formed when solving practical problems. Author accessibly explains the mechanism of implementation of various type of relationships describes the typical errors of students and gives rules and algorithms for creation of different types of relationships. Attention focused to the types of joins that supported in the mode of QBE: internal joins – single-column and multi-column; external joins – left and right; self-joins theta-joins with the condition of inequality, cross-joins. A join is an operation that compares and combines values in the common fields of linked tables that are the source of data in a form, report, or query. A type of joins sets the method of viewing of the linked records based on the set relationship. The number of records displayed depends on the type of join. The basis of most joins (internal, external, self-joins) are equal values of the fields of join. Theta-join or unequal join created on the condition of the inequality of field values. A cross-join or Cartesian product used to generate combinations of records in a query. A join created between fields with an indefinite type of relation may lead to erroneous records, because the repetition of values in both fields of join leads to the appearance of all possible combinations of matching records. Examples of their use when constructing queries for solving practical problems are given. Material of the article contains elucidations and methodical recommendations for the study of theme of "System Management Databases" from discipline Information technologies.*

Key words: *method of study, Access, database, inter-table relationships, types of joins.*