

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**Методичні рекомендації
до виконання лабораторних робіт
з навчальної дисципліни**

**"ПРИНЦИПИ ПРОЕКТУВАННЯ БАЗ ДАНИХ
ТА БАЗ ЗНАНЬ"**

**для студентів напряму підготовки
6.051501 "Видавничо-поліграфічна справа"
всіх форм навчання**

Харків. ХНЕУ ім. С. Кузнеця, 2015

Затверджено на засіданні кафедри комп'ютерних систем і технологій.
Протокол № 7 від 13.02.2015 р.

Самостійне електронне текстове мережне видання

Укладач Пандорін О. К.

М 54 Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни "Принципи проектування баз даних та баз знань" для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" всіх форм навчання : [Електронне видання] / уклад. О. К. Пандорін. – Х. : ХНЕУ ім. С. Кузнеця, 2015. – 44 с. (Укр. мов.)

Подано методичні рекомендації до виконання лабораторних робіт за темами першого модуля навчальної дисципліни з метою формування практичних навичок та закріплення теоретичних знань щодо використання сучасних підходів до етапів проектування баз даних мультимедійного контенту на всіх стадіях життєвого циклу друкованих та електронних інтерактивних мультимедійних видань.

Рекомендовано для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" всіх форм навчання.

Вступ

Дисципліна "Принципи проектування баз даних та баз знань" вивчають студенти напряму підготовки 6.051501 "Видавничо-поліграфічна справа" всіх форм навчання протягом VIII семестру і є методологічною та методичною основою для подальшого опанування студентами технологій і методів розроблення програмної частини мультимедійних електронних видань; набуття практичних навичок у самостійному опрацюванні мультимедійної інформації та подання її як складової друкованих та мультимедійних електронних видань.

Дисципліна є методологічною, методичною та інструментальною основою для виконання аналітичної та практичної частин дисциплін, а також курсових і дипломних робіт.

Особливість лабораторних робіт – орієнтація на різноманітні програмні технології, вивчення яких допомагає сформуванню різнобічно підготовленого фахівця. У цьому практикумі розглядаються методи використання різних програмних технологій розповсюдження та опрацювання контенту мультимедійних середовищ, які презентовані різноманітними інформаційними технологіями на основі наскрізного розвитку одного додатка за індивідуального варіантом завдання.

У результаті виконання лабораторних робіт студенти мають набути навичок у проектуванні, розробленні, впровадженні та експлуатації програмних систем, що ґрунтуються на компонентному та об'єктно-орієнтованому підході.

Тексти програм СМД визначати за допомогою літер однакової ширини. Перехід до іншого рядка у фрагментах коду, там, де він вимовлений лише можливостями цього видання, позначено символом ↵. У процесі використання сучасних середовищ розроблення значна частина коду має будуватися автоматично. Частини коду, які не додаються автоматично, за допомогою графічних засобів розроблення, зображуються на сірому фоні.

Модуль 1

Базові концепції систем управління базами даних

Лабораторна робота 1. Визначення сфери застосування, збирання інформації про стан предметної сфери, складання функціональної схеми предметної сфери, аналіз вимог оброблення

Мета роботи – ознайомити студентів із основними прийомами і методиками процесу проектування першого етапу процесу розроблення інформаційної системи.

Проектування баз даних містить такі етапи:

1. Виявлення вимог, що містить системний аналіз цієї предметної області поставленого завдання і складання її словесного опису.

2. Концептуальне проектування, визначення функцій, що має виконувати інформаційна система, та інформації, необхідної для їх виконання (сутностей, їх атрибутів і зв'язків (відносин)). На цьому етапі СМД створити інформаційну концептуальну модель у вигляді ER-діаграми.

3. Логічне проектування. На цьому етапі слід здійснити вибір СУБД (системи управління базами даних), за допомогою якої базу даних буде реалізовано, відображення моделі сутність – зв'язок на об'єкти, що підтримуються середовищем СУБД. У лабораторному курсі передбачено використання реляційної СУБД *MS SQL Server*, таким чином необхідно відобразити об'єкти моделі сутність – зв'язок на відношення реляційної моделі.

Основне завдання першого етапу – оцінювання проекту, його цілей і завдань, а також визначення сутностей і функцій на високому рівні. Мета – отримати якомога повнішу інформацію про систему (повне та однозначне розуміння вимог замовника) і передати цю інформацію у формалізованому вигляді для такого здійснення етапу аналізу.

Наприклад, створимо значно спрощену модель системи видавничого обліку роботи над виданнями.

Слід визначити сферу застосування. Існує картотека співробітників видавництва (авторів, редакторів, коректорів, художників) і список видань, що має бути видано. Потрібно побудувати систему для

підтримки контролю за виконанням доручень під час опрацювання видань.

Співробітник одночасно працює над декількома книгами, і над однією книгою можуть працювати декілька співробітників.

Контрольні запитання

1. Розкрийте зміст понять "інформація" і "дані".
2. Розкрийте зміст поняття "інформаційна система".
3. Наведіть основні класифікаційні ознаки інформаційних систем.
4. Розкрийте зміст поняття "життєвий цикл інформаційної системи".
5. Наведіть відмінні риси процесу опрацювання економічної інформації.
6. Охарактеризувати види забезпечення 1С.
7. Охарактеризуйте роль і місце інформаційного забезпечення 1С.
8. Розкрийте зміст понять "позамашинна інформаційна база" і "внутрішньомашинна інформаційна база".
9. Наведіть місце й роль СКБД в пакеті 1С.
10. Наведіть визначення предметної сфери СКБД.

Лабораторна робота 2. Концептуальне проектування

Мета роботи – ознайомити студентів із основними методиками, прийомами й технологічними інструментами для стадії концептуального проектування у процесі розроблення інформаційної системи.

Концептуальну модель системи побудуємо у формі ER-моделі (модель "сутність – зв'язок") – широко використовуваного інструменту розроблення баз даних, що є формалізованим графічним зображенням предметної області у вигляді прямокутників, ліній і спеціальних символів.

Перевага ER-моделі полягає в тому, що вона зрозуміла і розробникам баз даних, і замовникам, тобто майбутнім її користувачам, і тому всі можливі зміни у структурі даних погоджено між ними саме на рівні цієї моделі. ER-модель ніяк не пов'язана ні з якою конкретною СУБД, цей вибір слід здійснювати згодом. У основі ER-моделі покладено поняття "сутність" (*entity*) і "зв'язок" (*relation*). Не існує задовільного формального опису поняття "сутність".

Можна сказати, що це – сукупність значущих об'єктів, що існують у межах предметної області. Наприклад, у процесі побудови системи управління контактами видавництва та авторів доведеться оперувати сутностями "Співробітник", "Видання", а в разі створення банківської системи – сутностями "Клієнт" і "Рахунок".

Сутності прийнято називати іменником в однині.

Кожна сутність містить набір атрибутів, що є названими властивостями сутності.

Наприклад, сутність "Співробітник" може мати атрибути:

- 1) прізвище;
- 2) ім'я;
- 3) дата народження;
- 4) табельний номер;
- 5) *ISBN* видань.

Сутність – сукупність об'єктів, а кожен об'єкт цієї сукупності – екземпляр сутності. Із точки зору технології об'єктно-орієнтованого програмування, сутність – це клас об'єктів, а кожен об'єкт цієї сукупності – екземпляр класу.

Екземпляри мають бути помітні, тобто не мають бути двох екземплярів із однаковими наборами атрибутів. Кожна сутність має набір атрибутів, унікальний для кожного екземпляра сутності, названого ключем.

Наприклад, для сутності "Автор" ключ складається з одного атрибута "Табельний номер", оскільки точно не існує двох і більше студентів із однаковим значенням цього атрибута.

Між сутностями може бути встановлено зв'язки, що визначають, як сутності співвідносяться або взаємодіють між собою. Зв'язок може існувати між двома сутностями (бінарний зв'язок) або між сутністю і нею ж самою (рекурсивний зв'язок). Наявність зв'язку означає, що одна з них має атрибут або набір атрибутів складовий ключ іншої сутності. Наприклад, сутність "Автор" і сутність "Видання" пов'язані між собою атрибутом "*ISBN* видань групи", який є ключем сутності "Видання" і наявний у наборі атрибутів сутності "Автор".

Зв'язки бувають трьох типів:

1. Зв'язок "*один до одного*". За такого зв'язку один елемент першої сутності пов'язаний не більш ніж з одним екземпляром іншої сутності. Цей тип зв'язку зустрічається порівняно рідко.

2. Зв'язок "один до багатьох". У цьому разі один екземпляр першої сутності пов'язаний із декількома екземплярами іншої сутності, але один екземпляр іншої сутності пов'язаний не більш ніж з одним екземпляром іншої сутності. Першу сутність слід вважати в цьому зв'язку основною, іншу – підпорядкованою. У банківській системі кожен клієнт може мати багато рахунків, але рахунки тільки один хазяїн – клієнт.

3. Зв'язок "багато до багатьох". У цьому разі кожен екземпляр першої сутності може бути пов'язаний із декількома екземплярами іншої сутності, а кожен екземпляр іншої сутності – із декількома екземплярами першої. Як приклад можна навести зв'язок між сутностями "Співробітник" і "Видання" видавничої системи. Цей зв'язок має тип "багато до багатьох", оскільки один співробітник може працювати над декількома виданнями належати книг, а над одним виданням можуть працювати одночасно декілька співробітників.

Отже, для побудови ER-моделі необхідно спочатку виявити сутності, характерні для заданої предметної області та визначити для кожної сутності набір атрибутів, що мають значення виходячи з поставленого завдання. Потім необхідно визначити всі наявні зв'язки між сутностями характеристики цих зв'язків. Після цього слід збудувати ER-діаграму, тобто власне графічне зображення моделі з використанням однієї з наявних нотацій. У цій роботі ми дотримуватимемося нотації *Microsoft Entity Framework*, і в ER-діаграмі зображено у формі прямокутників, усередині яких перераховано імена атрибутів. Атрибути слід підкреслювати. Ім'я сутності вказано у прямокутнику першим і відділено від списку атрибутів рисою. Зв'язки між сутностями зображено у вигляді ліній, що сполучають прямокутники відповідних сутностей. Для сутності, що перебуває з боку "багато", лінія зв'язку має закінчуватися значком із трьох ліній, що розходяться. Необов'язковість зв'язку позначено значком, що має на кінці невелике коло.

Для кожної сутності предметної області бази даних необхідно:

дістати список атрибутів сутності;

визначити функціональні залежності;

визначити можливі ключі, зокрема, розглянувши унікальний ідентифікатор сутності;

виконати нормалізацію сутності;

призначити первинні ключі нових, отриманих у результаті нормалізації сутностей;

сформувати бізнес-правила підтримки цілісності сутності.

Для кожного зв'язку між сутностями необхідно:

визначити потужність зв'язку;

визначити обов'язковість входження сутності у зв'язок;

дозволити зв'язки "багато до багатьох";

призначити первинні ключі асоціативних сутностей виходячи з унікального ідентифікатора зв'язку і процедури міграції ключів за умови нормалізації;

визначити не ключові атрибути асоціативних сутностей, якщо вони потрібні;

сформувати бізнес-правила підтримки цілісності зв'язків;

документувати логічну модель реляційної бази даних;

перевірити логічну модель реляційної бази даних;

У процесі проектування БД зручно користуватися різними угодами про найменування.

Microsoft використовує такі:

імена сутностей завжди є іменниками в однині;

первинний ключ завжди має вигляд *Id*.

Усі їх властивості розподілено на дві групи:

пов'язані безпосередньо з атрибутами сутності: *Id* тощо;

навігаційні – службові, покажчики на інші сутності.

Останні створено, виходячи з аналізу зв'язків сутностей. Вони дозволяють зручно і просто запрошувати пов'язану з даного зв'язку, але і їх тип. До того ж символ "зірочка" означає необмежену кількість елементів.

Наприклад, як розглянуто в першій лабораторній роботі, можна виділити такі сутності:

1) видання;

2) співробітник.

Атрибутами сутності "Видання" будуть:

1) *ISBN*;

2) назва;

3) рік видання;

4) співробітники.

Ключем є *ISBN*.

Сутність "Співробітник" матиме атрибути:

1) табельний номер;

2) прізвище;

- 3) ім'я;
- 4) по батькові;
- 5) спеціальність;
- 6) видання.

Ключем сутності є табельний номер співробітника.

Нагадаємо, що ключем сутності є атрибут або набір атрибутів, значення яких є унікальними для кожного екземпляра сутності. Тепер розглянемо зв'язки, наявні між сутностями.

1. Оскільки над кожним виданням можуть працювати декілька співробітників, а співробітник може працювати над декількома виданнями, то між сутностями "Співробітник" і "Видання" є зв'язок "багато до багатьох". Оскільки відразу ж після прийняття на роботу співробітник може ще не бути пов'язаним роботою ні над одним виданням, а під час формування плану видань може бути не призначено виконавців, то зв'язок необов'язковий з обох боків.

На рис. 1 показано ER-діаграму, побудовану для цього завдання, урахуваючи наявні зв'язки. Ім'я кожної сутності відокремлено від імен атрибутів рисою, визначальні атрибути кожної сутності позначено відповідним символом ключа.

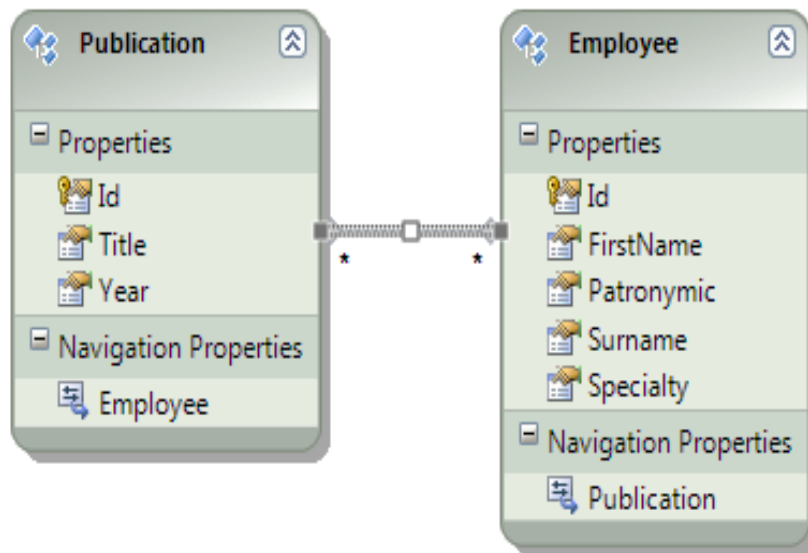


Рис. 1. Модель сутність – зв'язок

У принципі, уже на рівні ER-моделі можна зробити її нормалізацію. У цій роботі вимагається, щоб модель задовольняла умовам третьої нормальної форми.

Нагадаємо, що відношення перебуває в першій нормальній формі, якщо значення всіх його атрибутів атомарні, тобто є прості, а не складені типи даних. У нашому прикладі цю умову виконано для всіх сутностей. Значеннями атрибутів усіх сутностей є тільки числа, терміни або дати.

Масивів, структур або підтаблиць серед них немає.

Сутність задовольняє умовам другої нормальної форми, якщо кожен неключовий атрибут залежить від усього первинного ключа, але не залежить від якої-небудь частини первинного ключа. Із визначення випливає, що умову другої нормальної форми може бути порушено тільки у складеного первинного ключа. Сутність перебуває у третій нормальній формі, якщо вона перебуває у другій нормальній формі та кожен неключовий атрибут залежить тільки від первинного ключа, але не ключового атрибута. Усі сутності нашого прикладу задовольняють умови третьої нормальної форми.

Побудову ER-діаграми здійснюємо в середовищі *MS Visual Studio*. Створюємо новий проект (рис. 2).

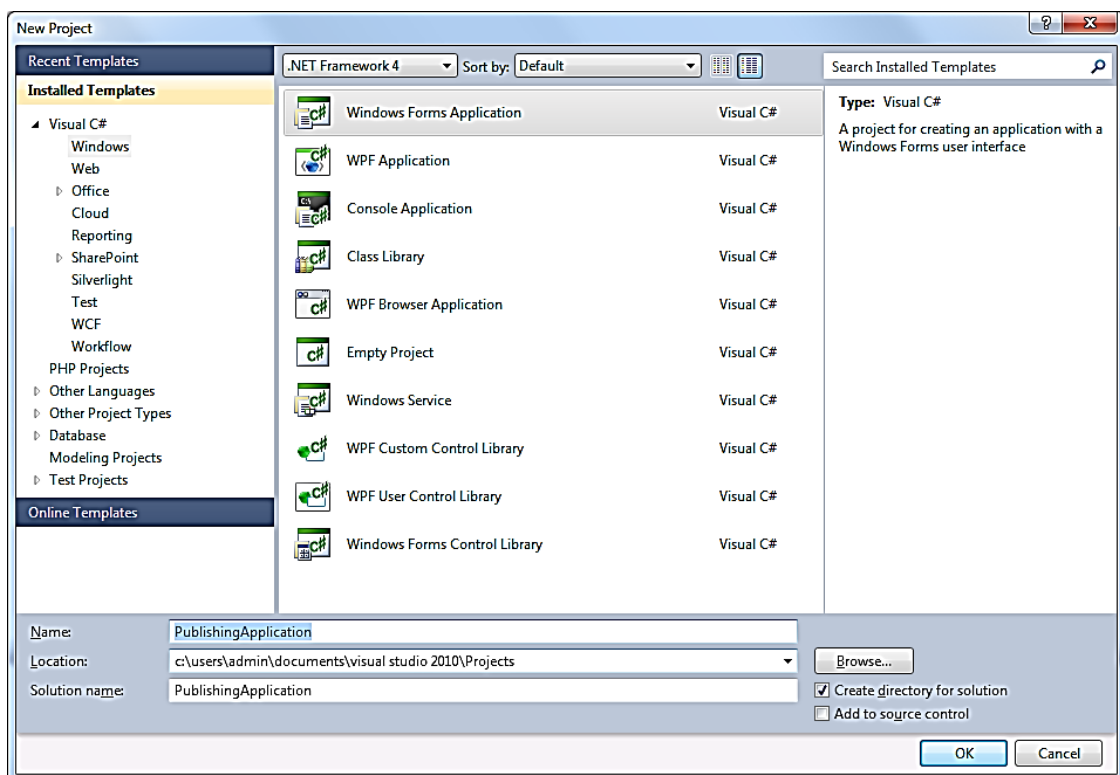


Рис. 2. Діалог щодо створення нового проекту

Додаємо модель сутність – зв'язок (рис. 3, 4).

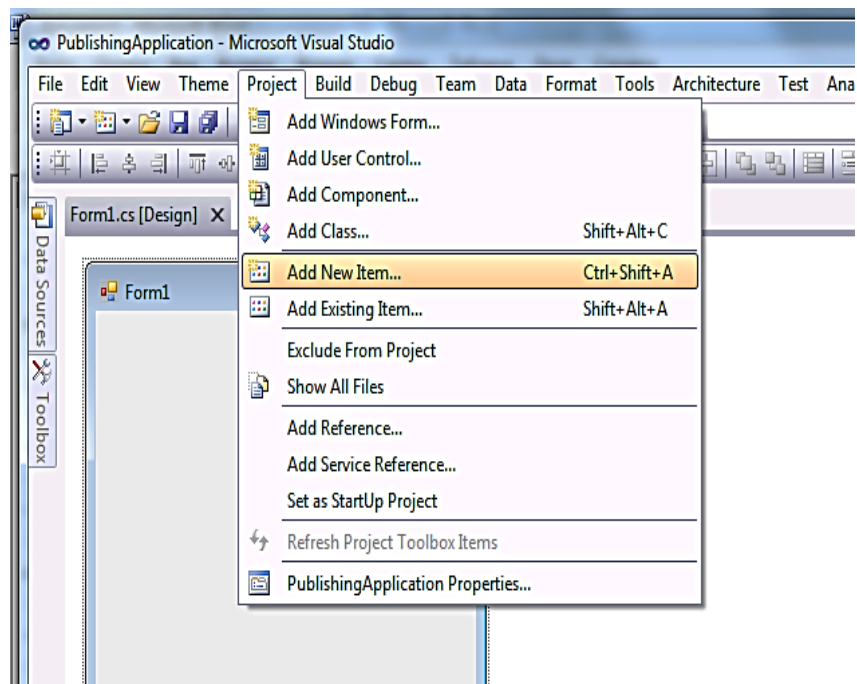


Рис. 3. Діалог щодо створення нового елемента проекту

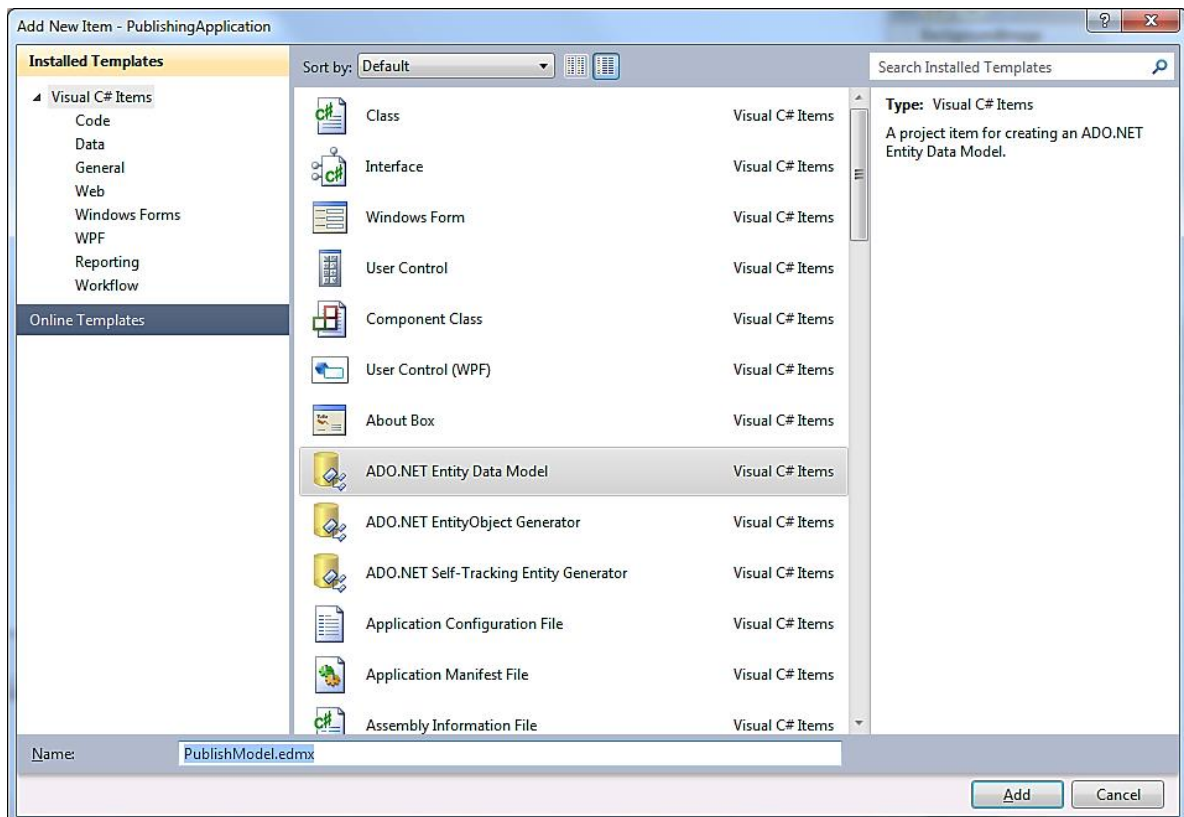


Рис. 4. Діалог щодо додання моделі

Якщо спроба не вдалася, закінчилася повідомленням $HRESULT = 0$, спробуйте спробувати ще раз через контекстне меню. Перегляд рішення (рис. 5).

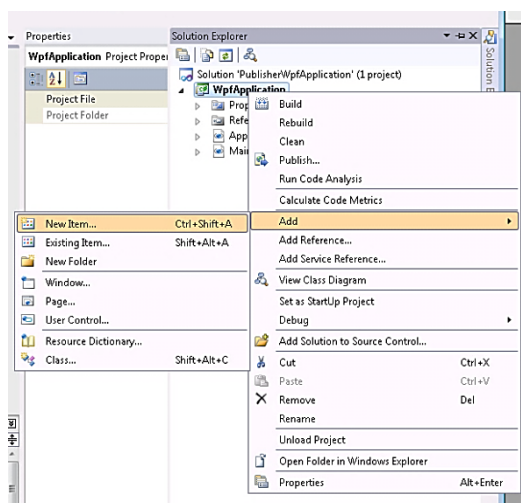


Рис. 5. Контекстне меню додавання нового елемента проекту в перегляд рішення

Модель має бути порожньою, саме в цьому середовищі ми і будемо її проектувати (рис. 6).

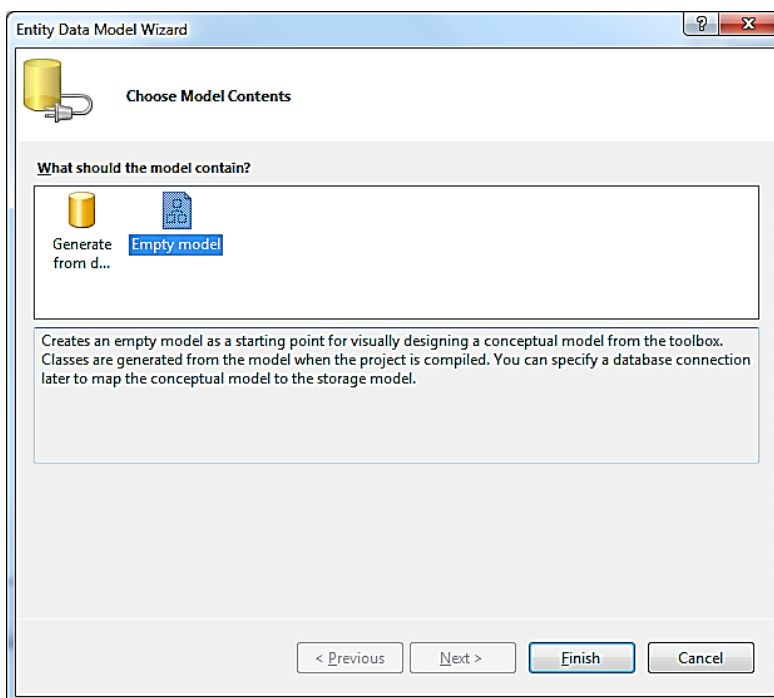


Рис. 6. Діалог щодо обрання джерела моделі

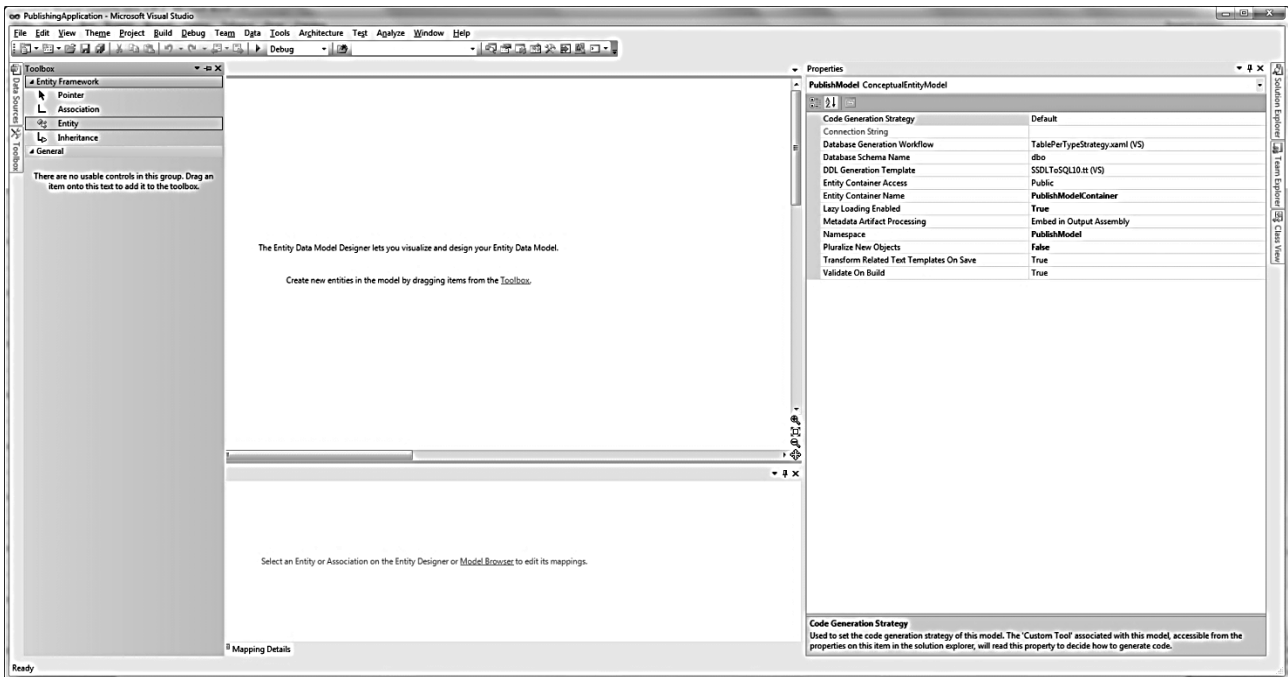


Рис. 7. Панель проектування моделі, панель інструментів моделі, властивості моделі

Перетягуємо компоненти-сутності з панелі інструментів (рис. 8) на панель моделі (рис. 9).

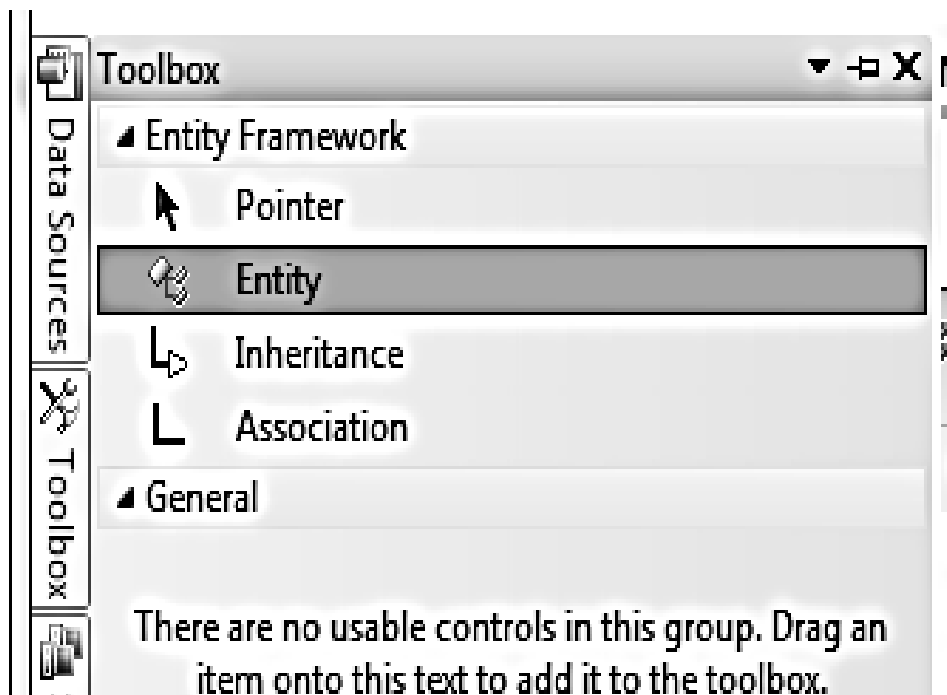


Рис. 8. Панель інструментів моделі. Нова сутність

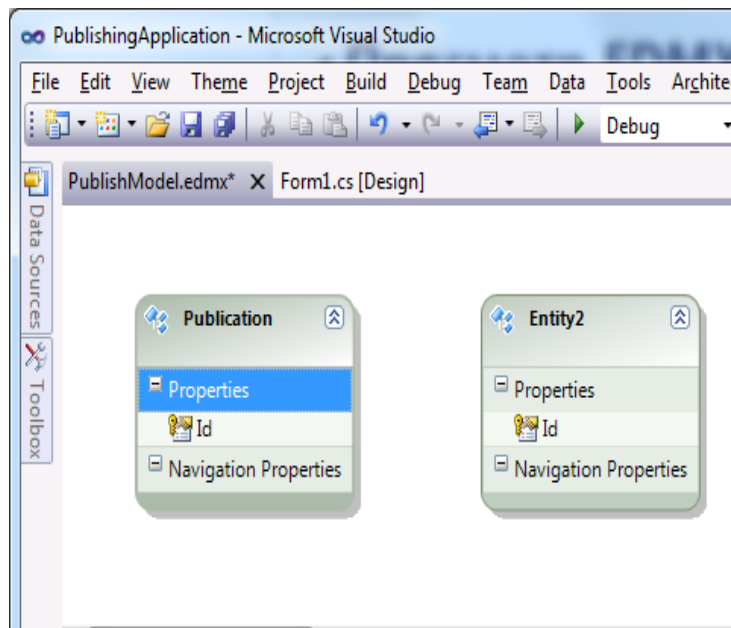


Рис. 9. Панель моделі. Сутності після створення

Додаємо властивості-атрибути (рис. 10).

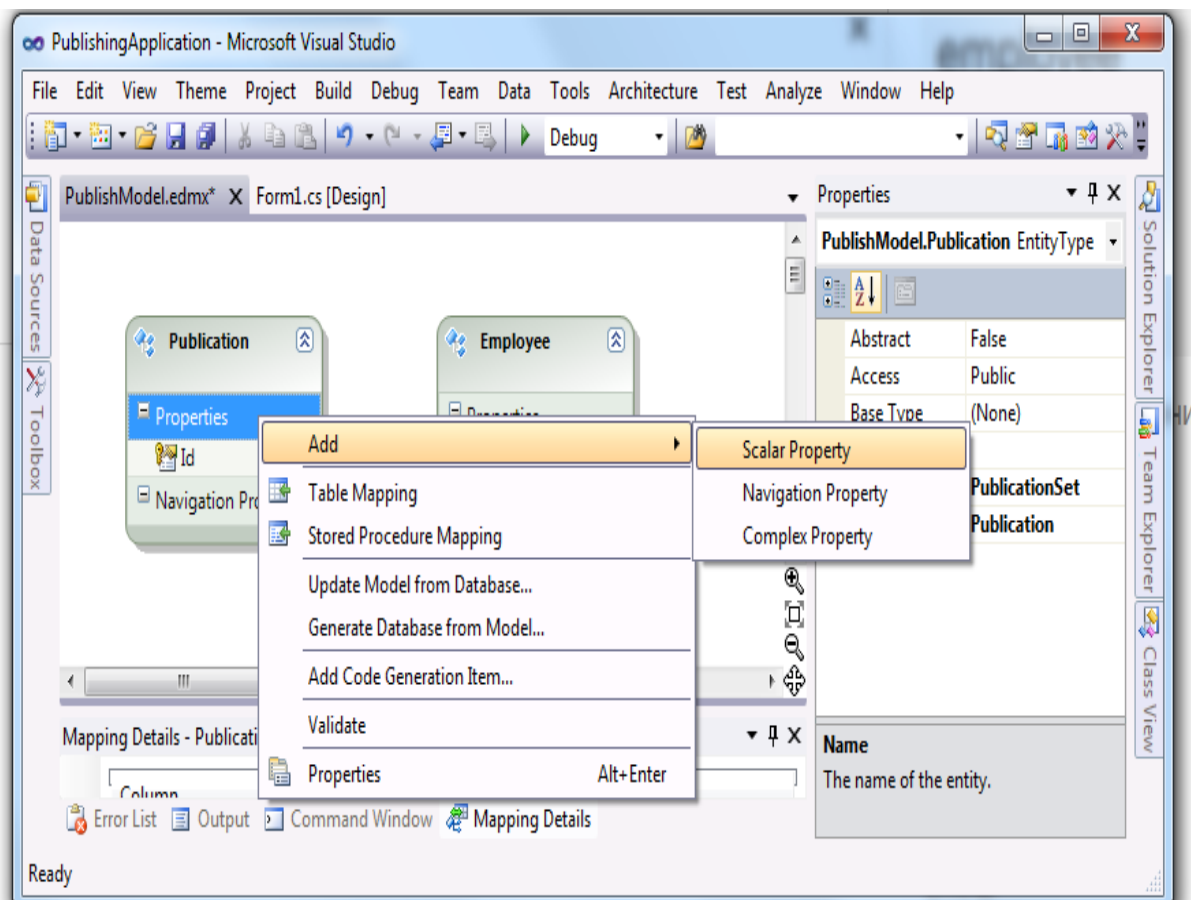


Рис. 10. Контекстне меню сутності. Додання властивості

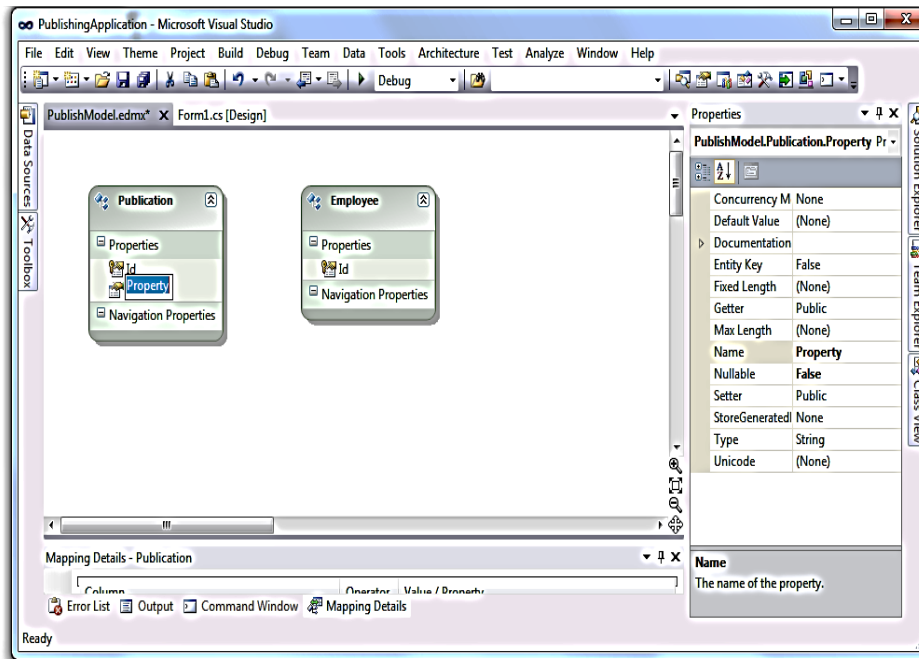


Рис. 11. Зміна імені властивості

Надаємо властивості значуще ім'я (див. рис. 10) та тип (див. рис. 11). Для *ISBN* необхідно використовувати 64-бітове ціле (рис. 12).

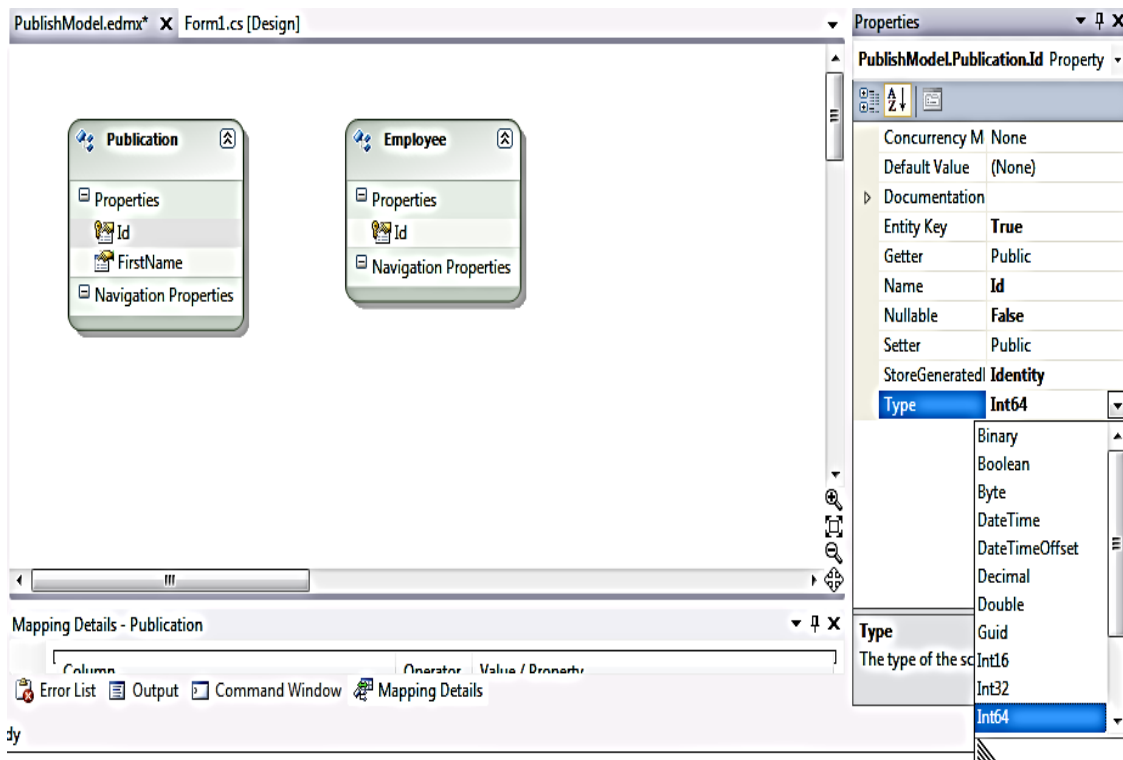


Рис. 12. Редагування типу властивості

Після створення скалярних атрибутів (рис. 13) додамо зв'язок між сутностями (рис. 14).

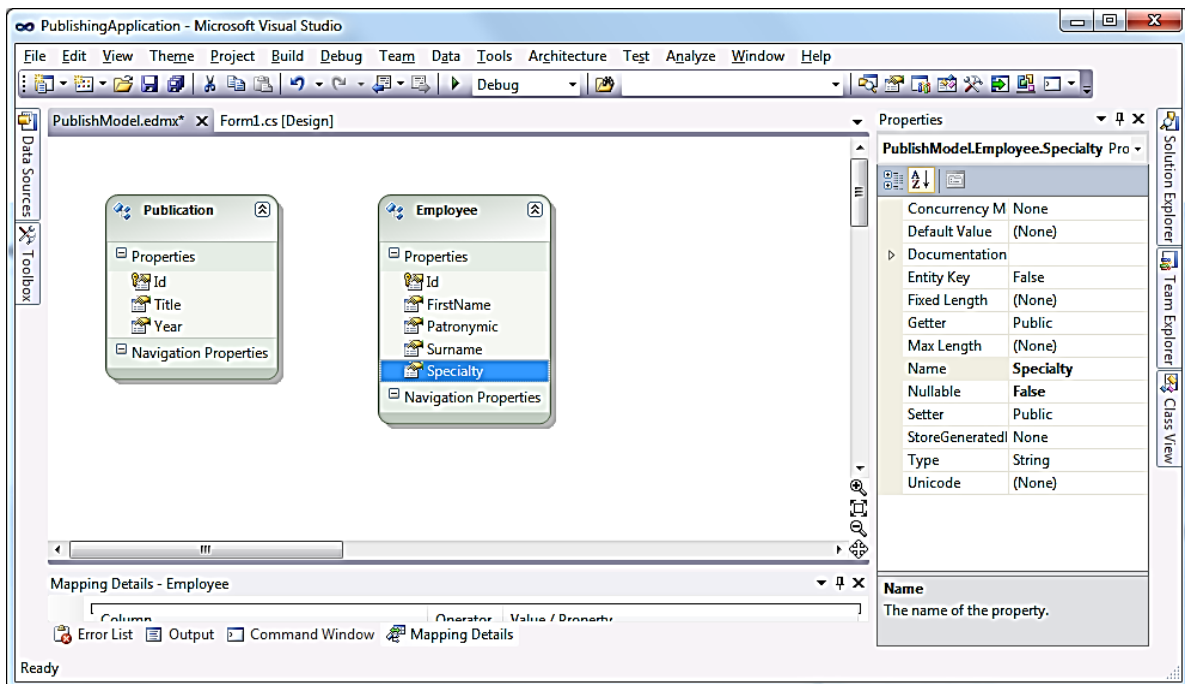


Рис. 13. Панель сутностей після створення скалярних атрибутів

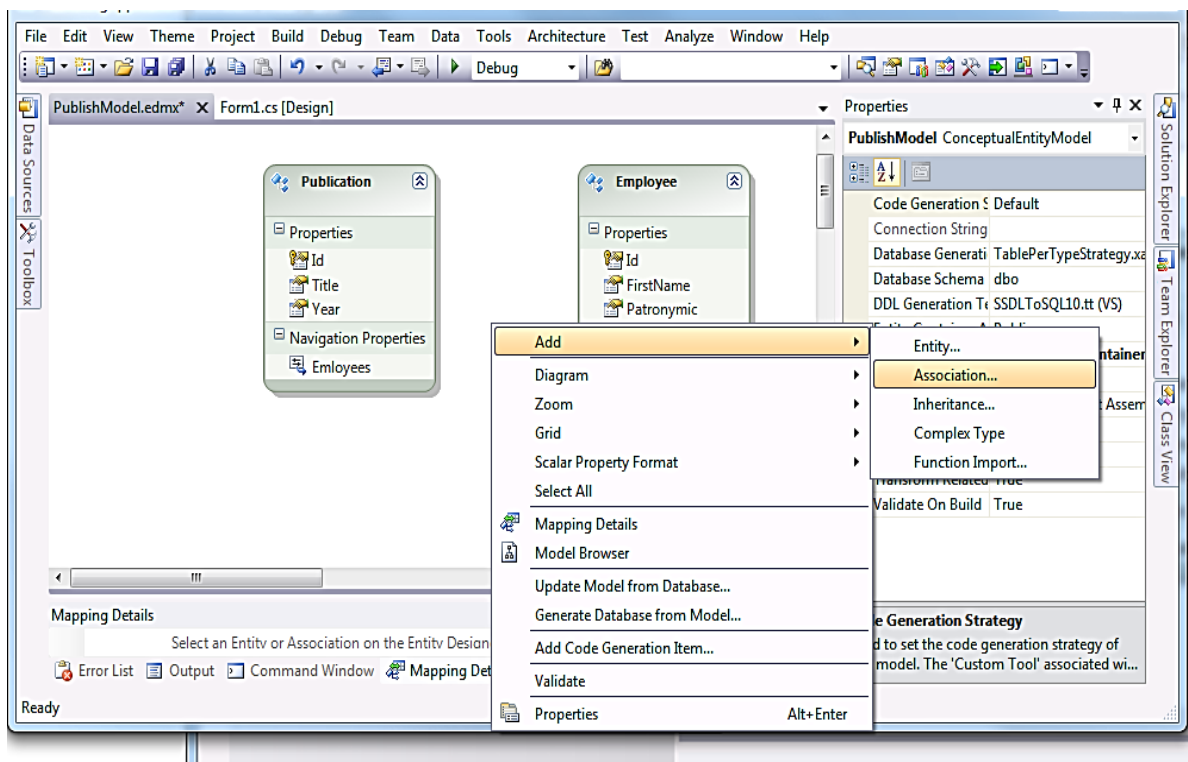


Рис. 14. Контекстне меню сутності. Додання зв'язку

Описуємо властивості зв'язку "багато до багатьох" (рис. 15).

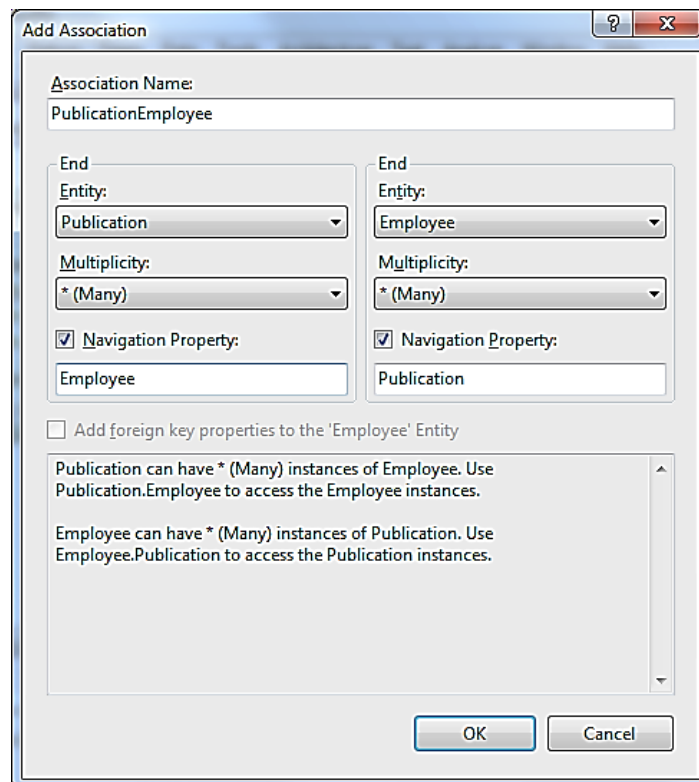


Рис. 15. Діалог щодо властивості зв'язку "багато до багатьох"

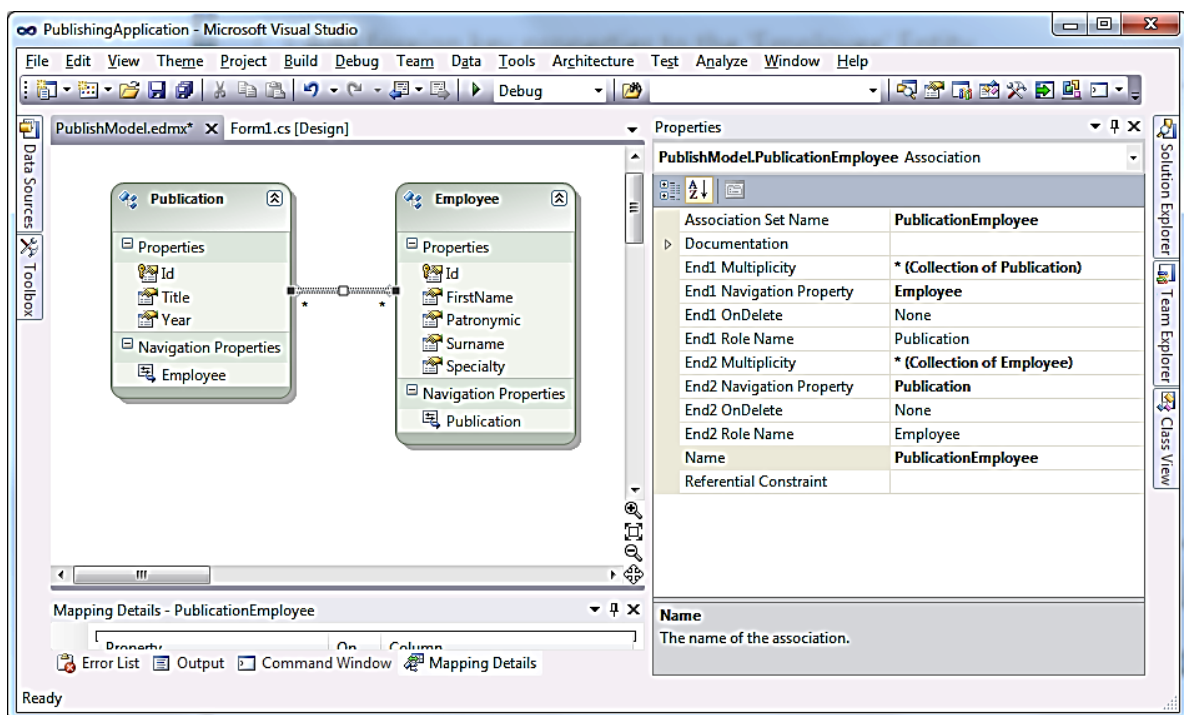


Рис. 16. Сутності, яких з'єднано, та властивості зв'язку "багато до багатьох"

Для зручності аналізу моделі відобразимо типи атрибутів (рис. 17).

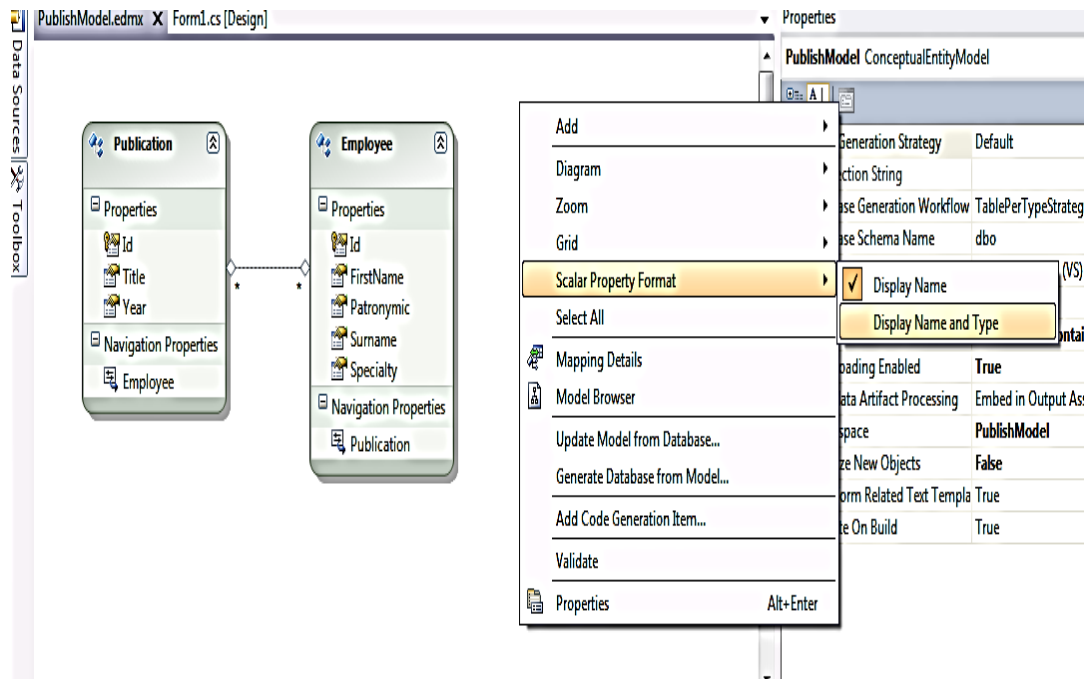


Рис. 17. Контекстне меню моделі. Призначення режимів відображення властивостей

Модель побудовано (рис. 18).

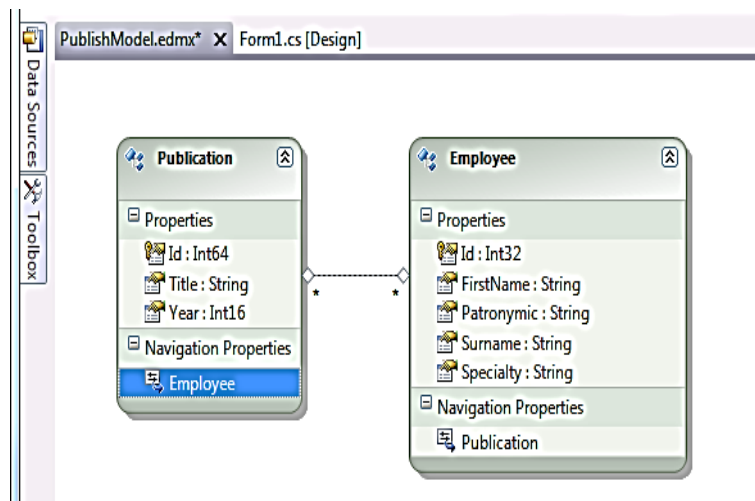


Рис. 18. Модель *PublishModel* із сутностями *Publications Employee*, які з'єднано зв'язком "багато до багатьох"

Таким чином використання графічних засобів дозволяє зробити побудову моделі більш зрозумілою.

Контрольні запитання

1. Наведіть основні терміни та визначення графічних мовних засобів ER-моделі.
2. Наведіть основні терміни та визначення текстових мовних засобів моделі даних.
3. Наведіть основні й похідні від них операції над даними у вигляді зв'язаних об'єктів.
4. Поясніть використання операцій, які найчастіше використовуються в процесі проектування, ведення й розвитку ІС. Привести приклади.

Лабораторна робота 3. Фізичне проектування

Мета роботи – ознайомити студентів із основними прийомами і технологічними інструментами для організації відображення об'єктної моделі на схему бази даних.

На такому етапі проектування необхідно перетворити ER-модель на реляційну, саме таку за набором типів стовпців (атрибутів), які підтримуються обраним сервером СУБД.

Основною структурою реляційної моделі є відношення (*relation*), графічною інтерпретацією якого є таблиця.

Кожне відношення складено з деякої обмеженої кількості кортежів, а зміст кожного кортежу визначено набором атрибутів відношення. Кожен атрибут має певний тип, значення якого взято з певного домену.

Кортежам відношення відповідають рядки таблиці, кількість стовпчиків таблиці дорівнює кількості атрибутів відношення, а тип величини, що перебуває у відповідному стовпці визначено типом відповідного атрибута. Зв'язки можуть бути пов'язані між собою за допомогою набору атрибутів, що однаково містяться в обох відношеннях. Зв'язки між відношеннями в реляційній моделі, на відміну від ER-моделі, можуть мати тільки тип "один до багатьох", тобто одне відношення завжди буде основним, а інші – підлеглим і одному кортежу основного відношення можуть відповідати декілька кортежів підлеглого відношення.

Ця відповідність означає, що в обох кортежів значення набору атрибутів, за яким пов'язано стосунки, збігаються. Зазвичай в основного відношення цей набір атрибутів є первинним ключем і, отже, унікальним

для кожного кортежу. В іншого відношення цей набір атрибутів названо зовнішнім ключем для цього зв'язку. Кортежів із однаковим значенням зовнішнього ключа може бути скільки завгодно, але не може існувати кортежів зі значенням зовнішнього ключа, якому не відповідав який-небудь первинний ключ основного відношення. Зв'язок "один до одного" слід розглядати, як окремий випадок зв'язку "один до багатьох".

Зв'язків "багато до багатьох" у реляційній моделі бути не може.

Щоб перетворити ER-модель на реляційну необхідно виконати такі дії:

1. Поставити кожній сутності ER-моделі у відповідність відношення реляційної моделі, до того ж кожному атрибуту сутності поставлено у відповідність атрибут відношення реляційної моделі. Ключ сутності стає первинним ключем відповідного відношення (*PRIMARY KEY*). Водночас імена сутностей і зв'язків, відповідно, як і атрибутів, можуть не збігатися. Бажано в разі зазначення імен зв'язків і атрибутів реляційної моделі слід використовувати латиницю, оскільки ці імена найчастіше мають бути ідентифікаторами в деякій мові програмування.

2. У кожне відношення, відповідне підлеглий сутності, додано набір атрибутів, відповідний ключу основної сутності, якщо, звичайно, він там вже не був наявний. У будь-якому разі цей набір атрибутів стає зовнішнім ключем у підлеглому відношенні (*FOREIGN KEY*).

3. За умови обов'язкового характеру зв'язку в атрибутів, відповідних зовнішньому ключу, встановлено властивість відсутності невизначених значень (*NOT NULL*).

4. Якщо в ER-моделі є зв'язок "багато до багатьох", то їх потрібно перетворити на зв'язок "один до багатьох", оскільки зв'язки "багато до багатьох" у реляційній моделі не допускають. Для цього в реляційну модель додано єднальне відношення, атрибути якого відповідають атрибутам первинних ключів обох відношень, що беруть участь у зв'язку "багато до багатьох". Єднальне відношення перебуватиме у зв'язку "один до багатьох" із кожним із цих відношень.

У даному прикладі зв'язок "один до багатьом" мають сутності "Співробітник" і "Видання".

За умови графічного проектування в середовищі *Visual Studio* доцільно і всі елементи моделей називати латиницею. Таким чином, сутності назвемо, відповідно, *Employee* і *Publication*. У реляційній моделі введено єднальне відношення *PublicationEmployee*, атрибутами якого будуть *Publication_Id* і *Employee_Id*. Перший атрибут відповідає

первинному ключу сутності *Publication*, а інший – первинному ключу сутності *Employee*. Це відношення матиме зв'язок "один до багатьох" до відношень, відповідних сутностей *Publication*, *Employee* і *Publication*.

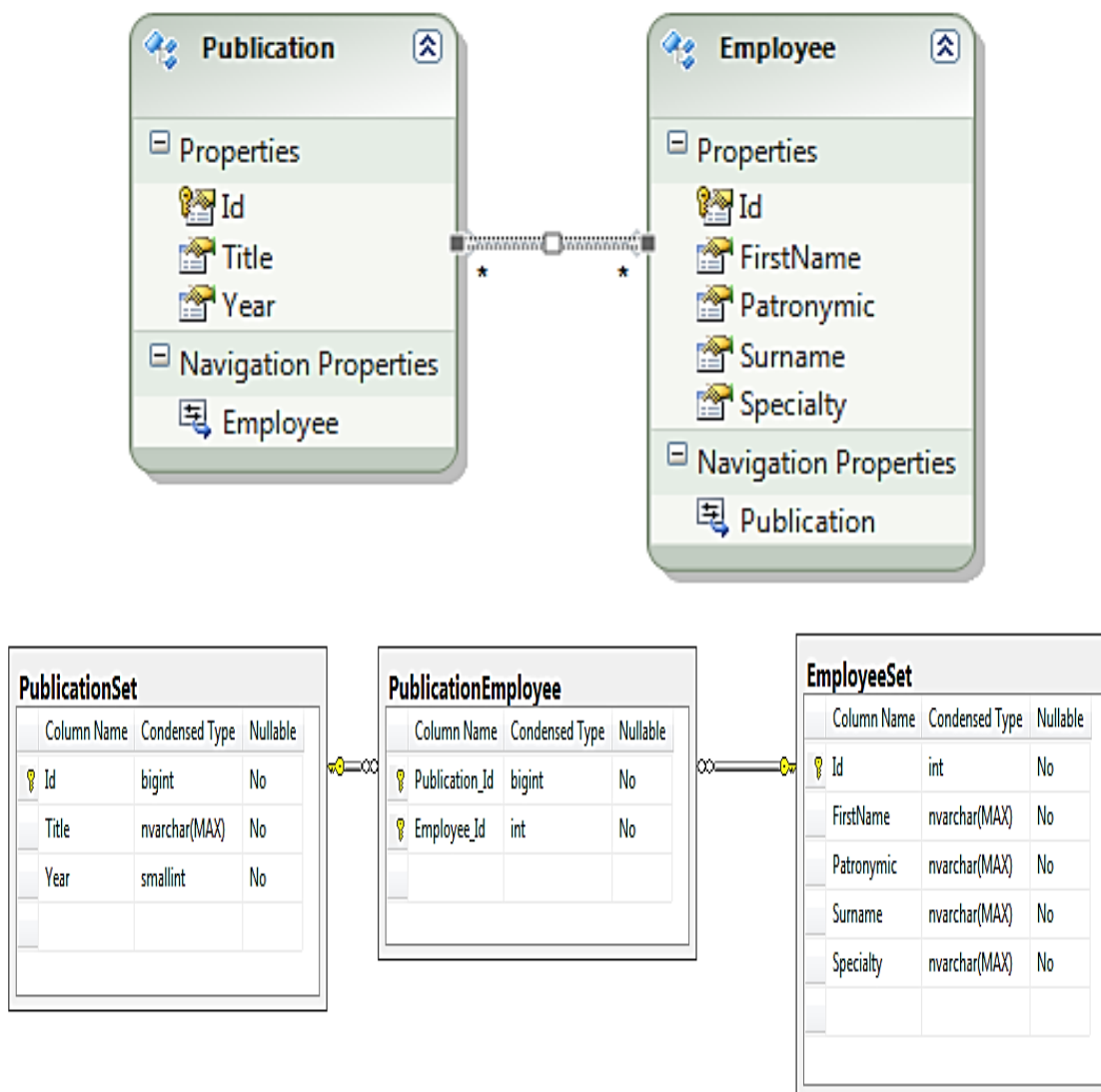


Рис. 19. ER-модель та відповідна реляційна модель

Визначену реляційну модель також зображуватимемо у графічній формі. Кожному відношенню відповідатиме прямокутник, у який буде вписано імена атрибутів відношення і їх типи. Набір можливих типів визначено офіційним стандартом мови SQL, проте в конкретних СУБД зазвичай є розбіжності зі стандартом. У нашій моделі атрибути можуть набувати текстових, числових значень, або бути якою-небудь датою. Розглянемо ці типи детальніше.

1. Тип дати. У цій роботі для створення таблиць слід використовувати СУБД *MS SQL Server*, у якій для вказання дати є тип *Date*.

2. Для презентації рядкових атрибутів слід використовувати стандартний тип *nvarchar(n)*. Тут *n* – максимальна довжина атрибута в символах. Довжину кожного рядкового атрибута необхідно погоджувати із замовником, але в цій роботі її можна вибирати довільно, на розсуд розробника.

3. Для числових атрибутів можна використовувати тип *integer*, якщо атрибут може набувати тільки цілочислових значень, або тип *decimal(m, n)*, якщо атрибут може мати дробову частину.

У реляційної моделі для нашого прикладу є три відношення, причому два з них відповідають двом сутностям ER-моделі, а третє відношення *PublicationEmployee* виникло в результаті перетворення зв'язку "багато до багатьох" між сутностями *PublicationEmployee* і *Employee* до двох зв'язків "один до багатьох" між відношеннями *Publication* і *PublicationEmployee*, а також *Employee* і *PublicationEmployee* (див. рис. 18).

Відношення, що бере участь у зв'язку збоку "багато", є підлеглим, а це означає, що атрибут підлеглого відношення може брати значення тільки зі списку значень відповідного йому за зв'язком атрибута головного відношення.

Створюємо відношення та зв'язки між ними в базі даних за допомогою *MS Visual Studio*.

Відкриваємо вікно *Server Explorer* (рис. 20).

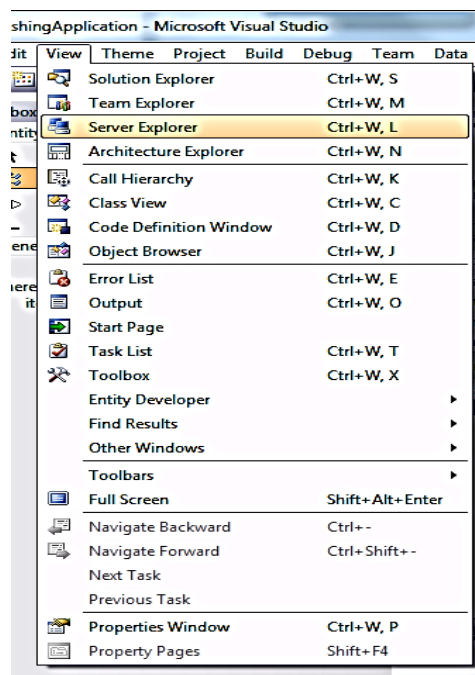


Рис. 20. Відкриття вікна *Server Explorer* за допомогою меню *View*

Створюємо нову базу в середовищі *MS SQL Server* (рис. 21). Для цього спочатку виберемо сервер БД.

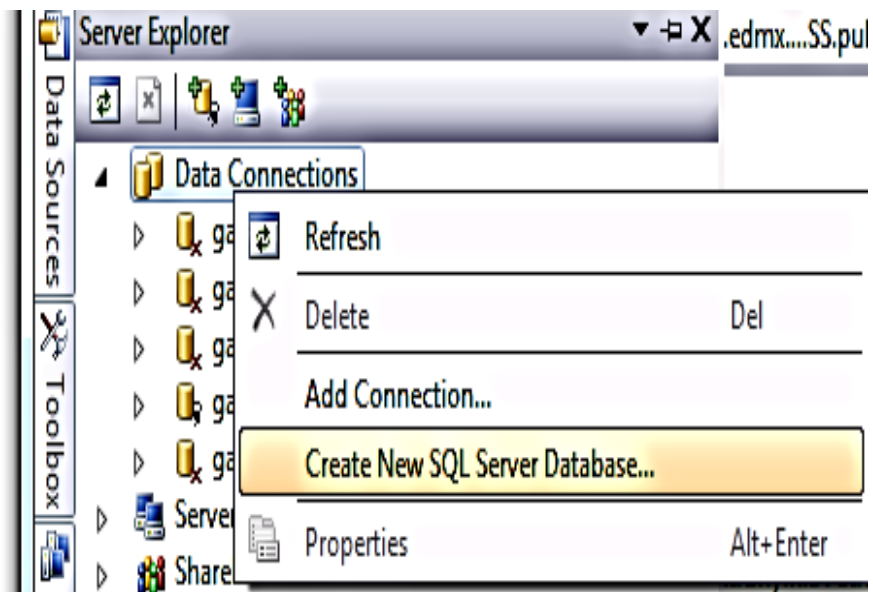


Рис. 21. Створення нової бази *SQL Server* за допомогою контекстного меню вікна *Server Explorer*

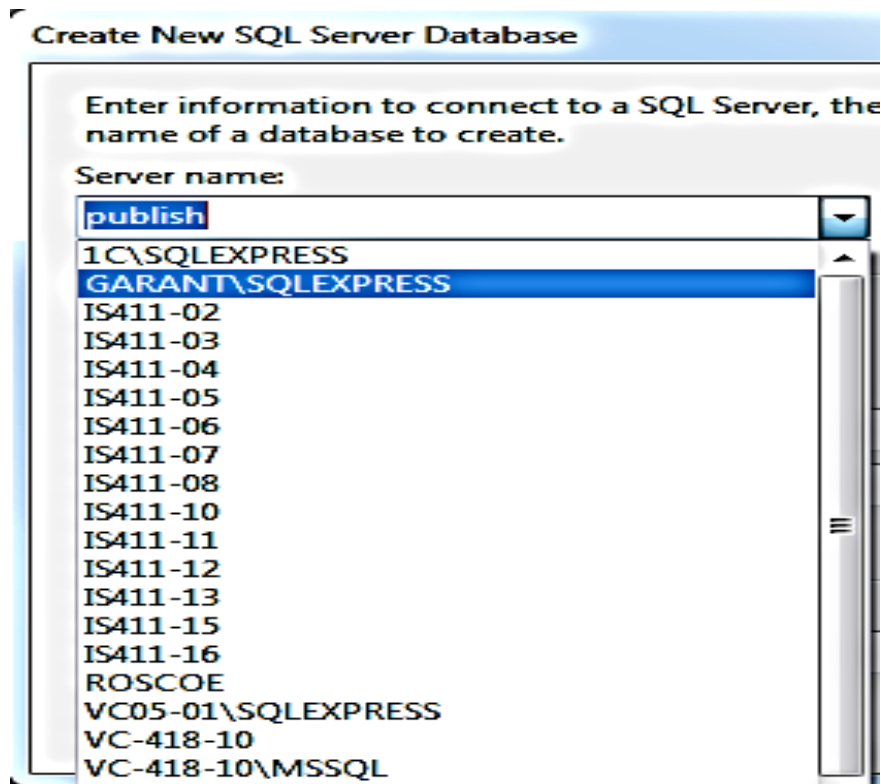


Рис. 22. **Діалогове вікно створення нової бази *SQL Server*.
Список доступних серверів**

Уводимо мнемонічне ім'я створюваної бази (рис. 23).

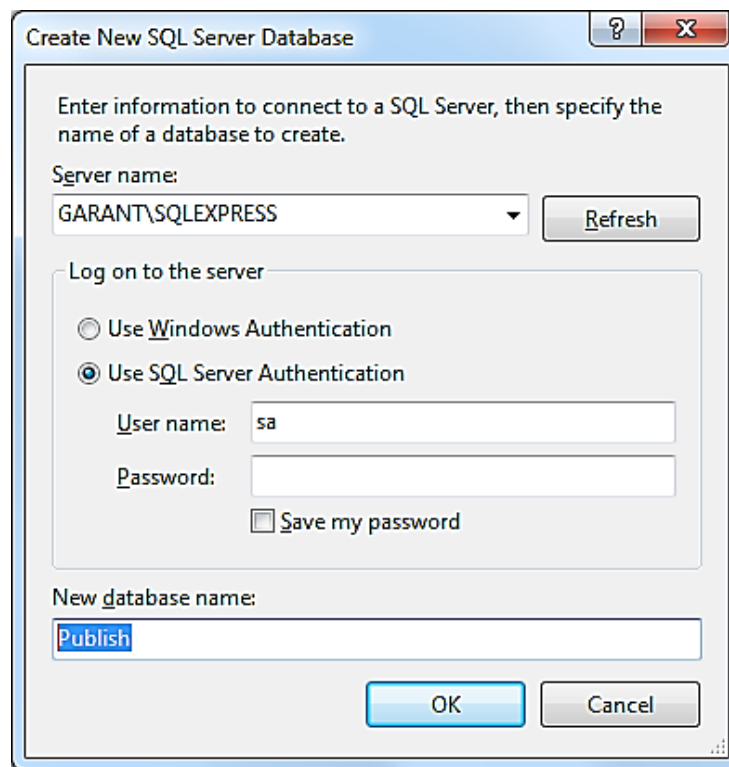


Рис. 23. **Діалог щодо створення бази нової бази SQL Server**

Створену базу видно за станом вікна *Server Explorer* (рис. 24).

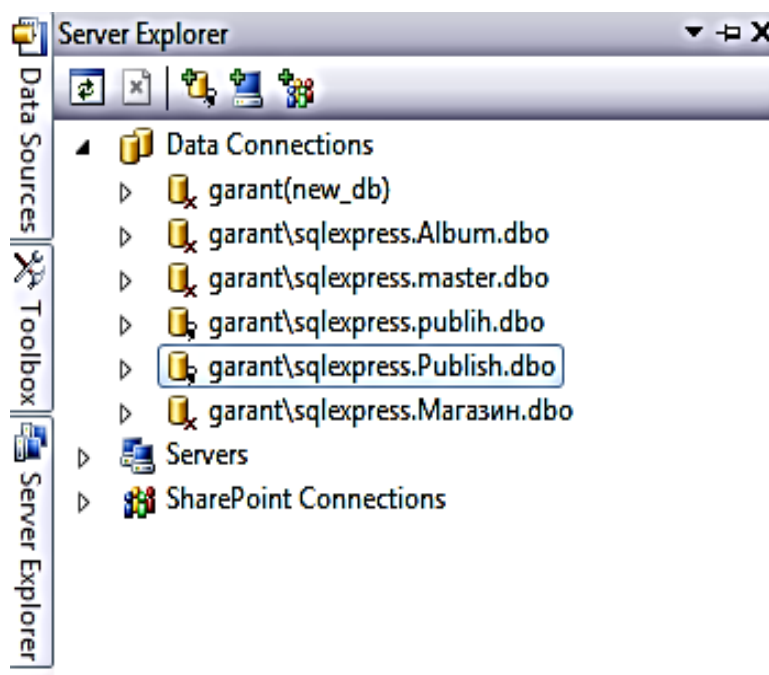


Рис. 24. **Вікно Server Explorer із посиланням на нову базу даних**

Створюємо таблиці-відношення за допомогою контекстного меню вікна *Server Explorer* (рис. 25).

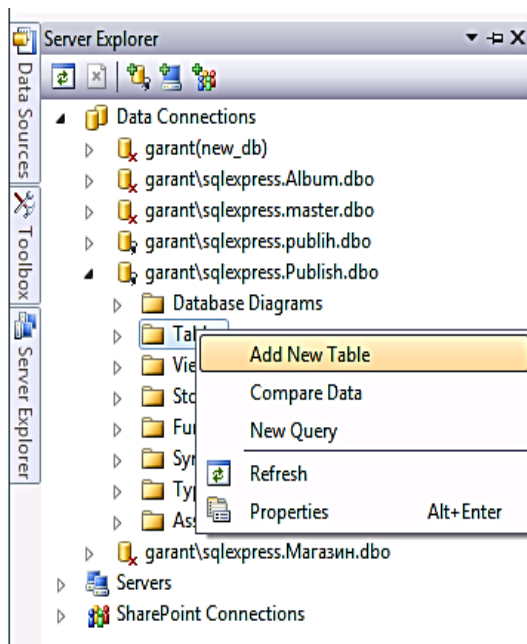


Рис. 25. Контекстне меню вікна *Server Explorer* "Створення таблиці"

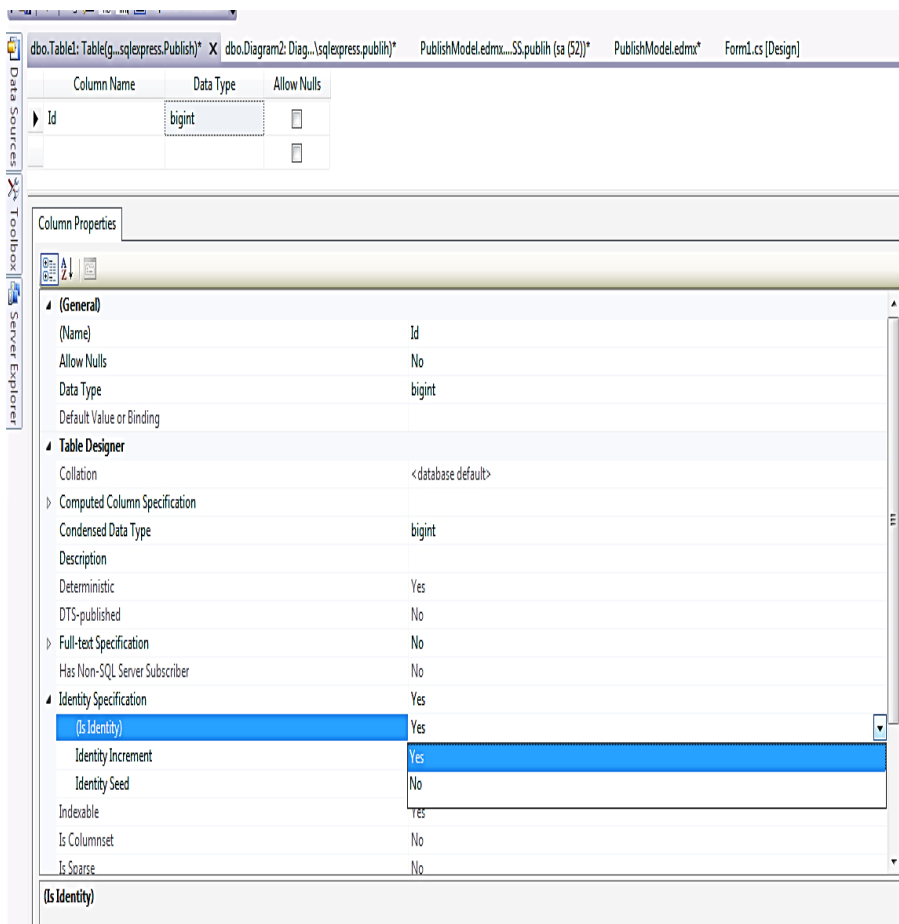


Рис. 26. Панель редагування стовпців

Створюємо стовпці.

Ключ зазвичай має ім'я *Id* і атрибут *Identity=Yes*.

Крім того, необхідно вказати, що це поле – ключове – за допомогою контекстного меню редагування стовпця (рис. 27).

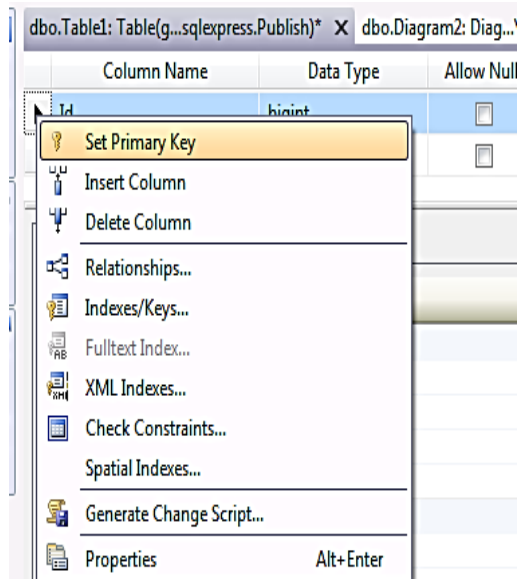


Рис. 27. Контекстне меню редагування стовпця.
Призначення первинного ключа

З'явився символ ключа біля імені стовпця (рис. 28).

Під час створення таблиці, яка відповідає зв'язку сутностей, перед заданням складеного первинного ключа необхідно виділити обидва ключові поля – за допомогою *Ctrl* або *Shift*. Якщо цього не зробити, знак ключа перестрибує з одного поля на інше.

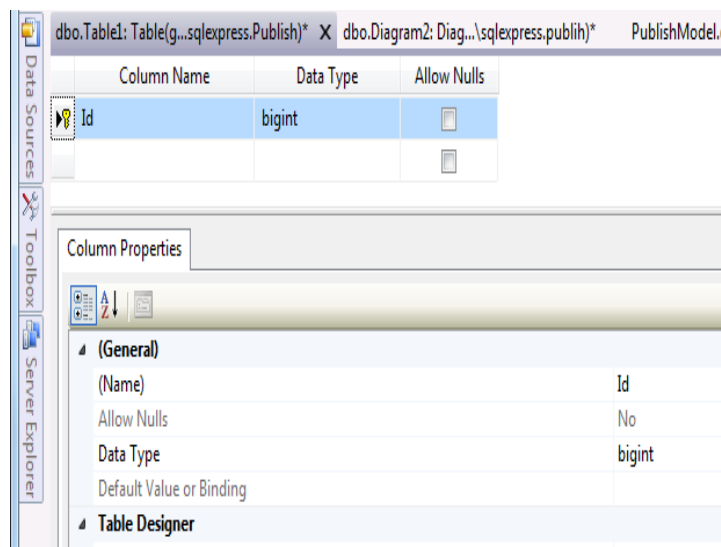


Рис. 28. Властивості стовпчику *Id*

Аналогічно створюють інші стовпці (рис. 29).

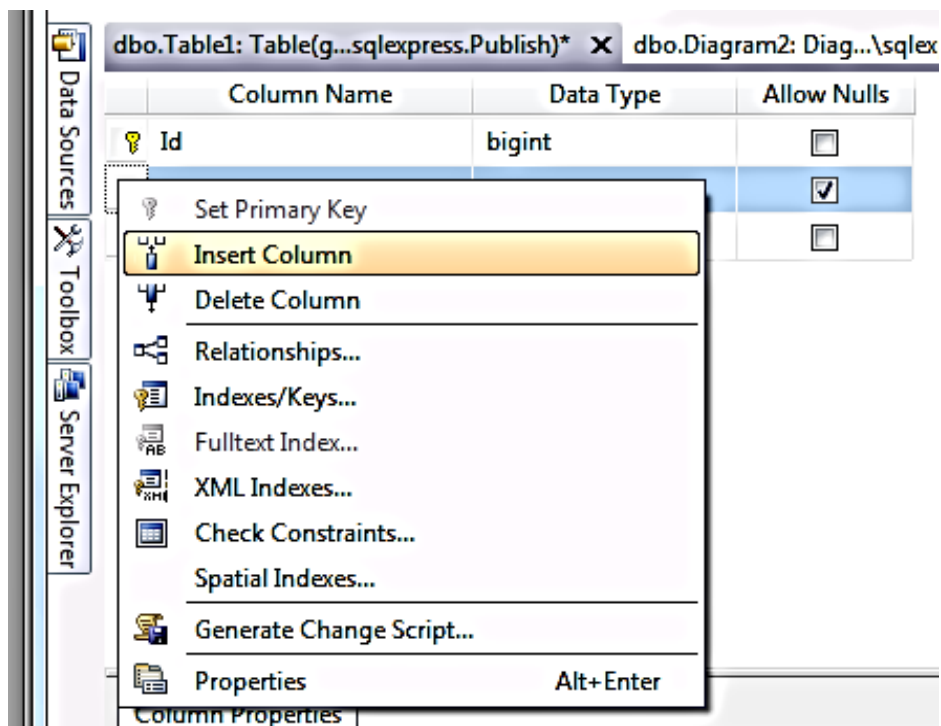


Рис. 29. Додання нового стовпця

Описи таблиць слід зберігати в разі закриття відповідного вікна редактора таблиці (рис. 30).

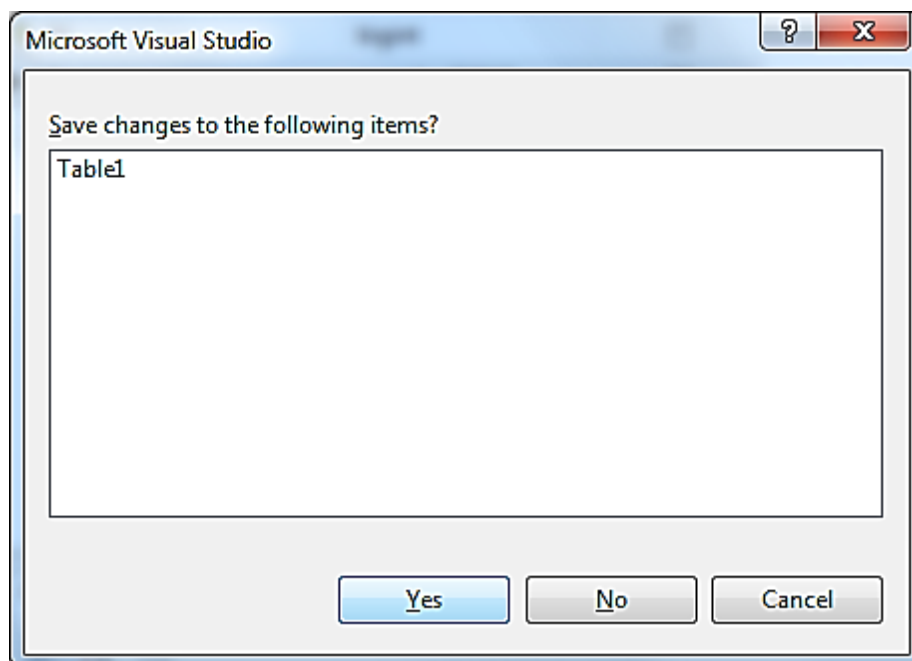


Рис. 30. Діалог про збереження таблиці

Таблиці слід надати мнемонічне ім'я (рис. 31).

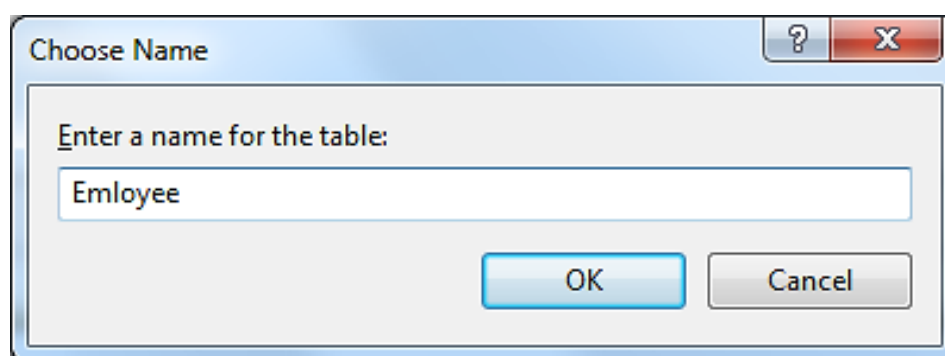


Рис. 31. **Діалог щодо зміни ім'я таблиці**

Створені таблиці-стосунки необхідно пов'язати.

Найпростіше це зробити в редакторі діаграм-схеми бази даних.

Створюємо нову діаграму (рис. 32).

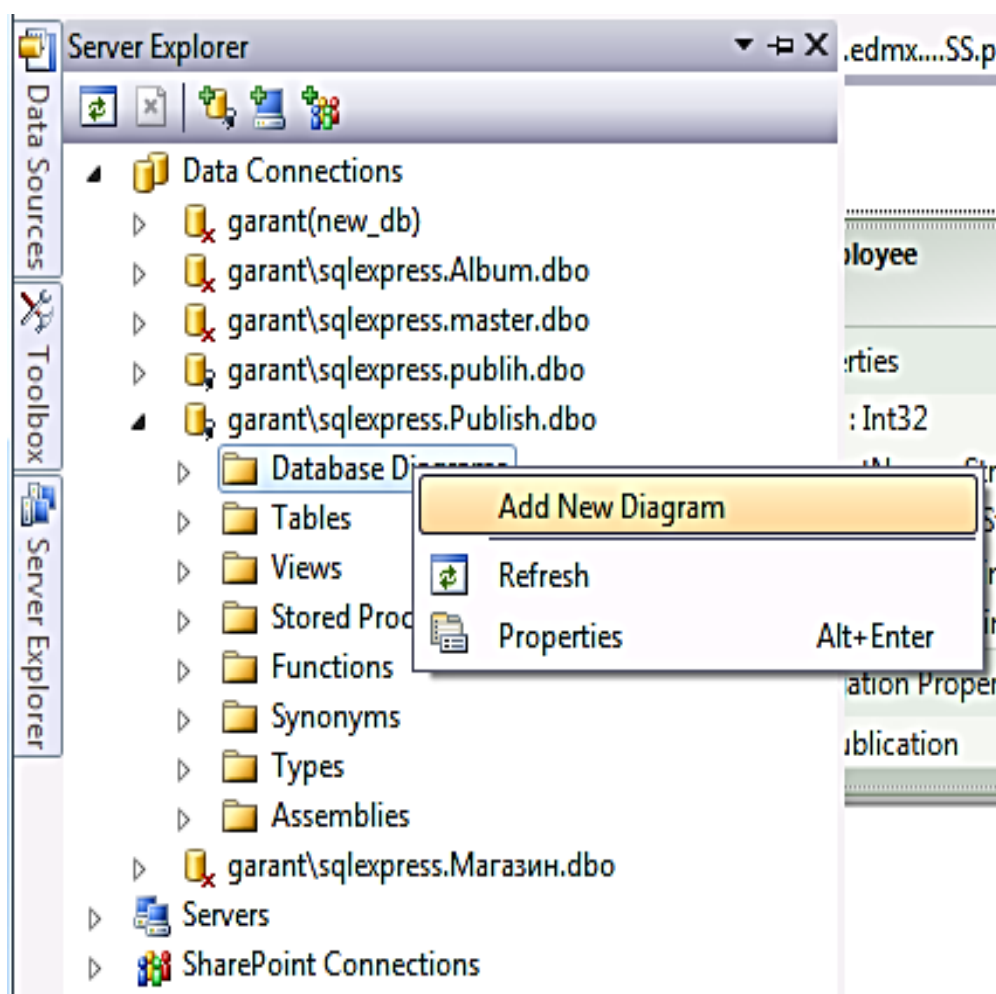


Рис. 32. **Контекстне меню вікна Server Explorer "Створення діаграм"**

Погоджуємось створювати об'єкти для діаграми (рис. 33).

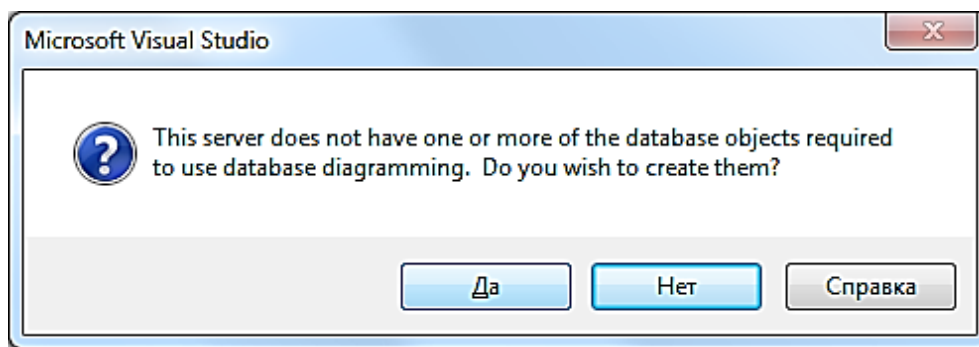


Рис. 33. Діалог щодо створення діаграм

Додаємо на діаграму всі таблиці (рис. 34).

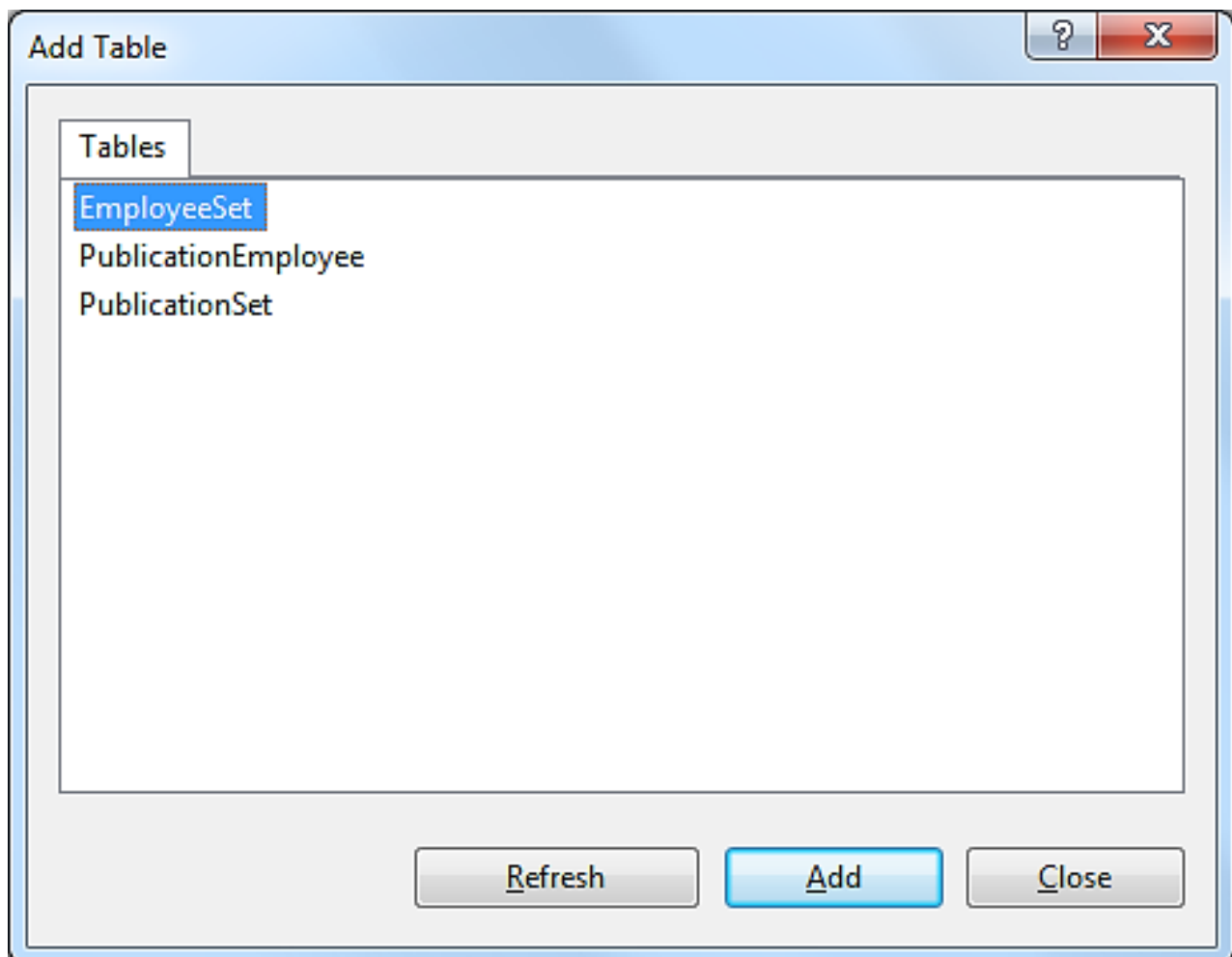


Рис. 34. Діалог щодо додання таблиць до діаграми

Пов'язуємо таблиці, перетягуючи посилання на ключові поля (рис. 35, 36).

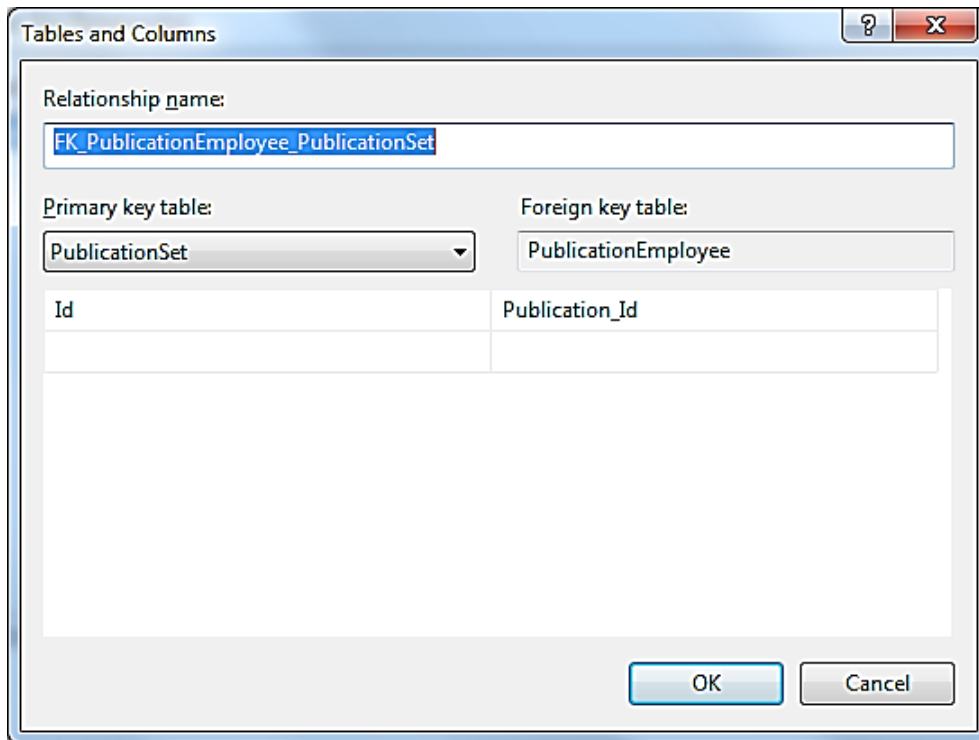


Рис. 35. Діалог щодо зв'язування таблиць

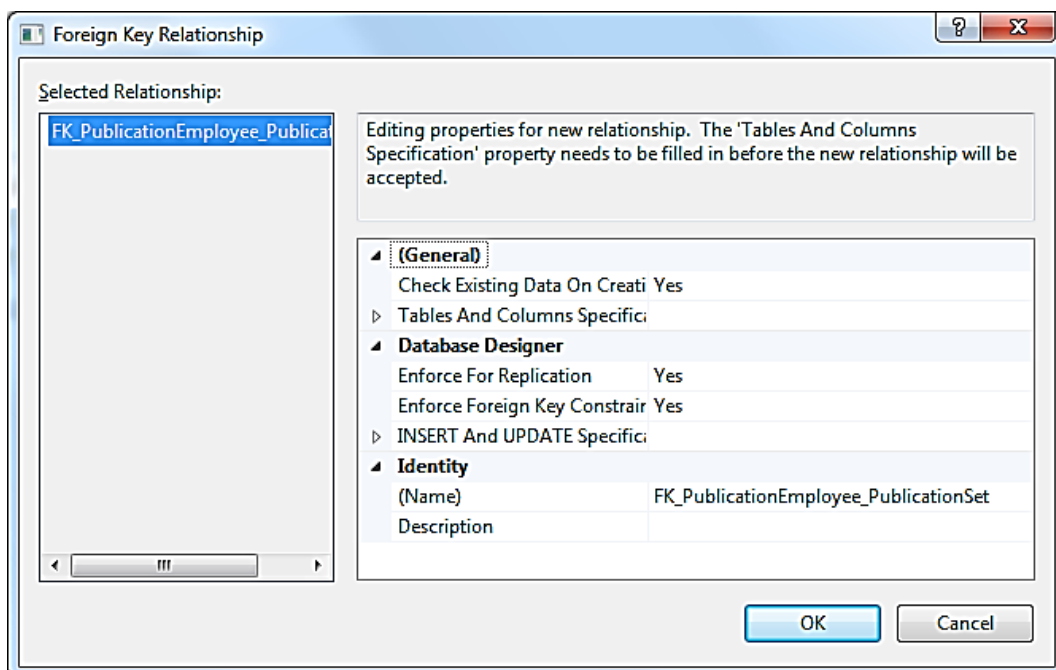


Рис. 36. Діалог щодо властивості зв'язку

Пов'язуємо всі таблиці (рис. 37).

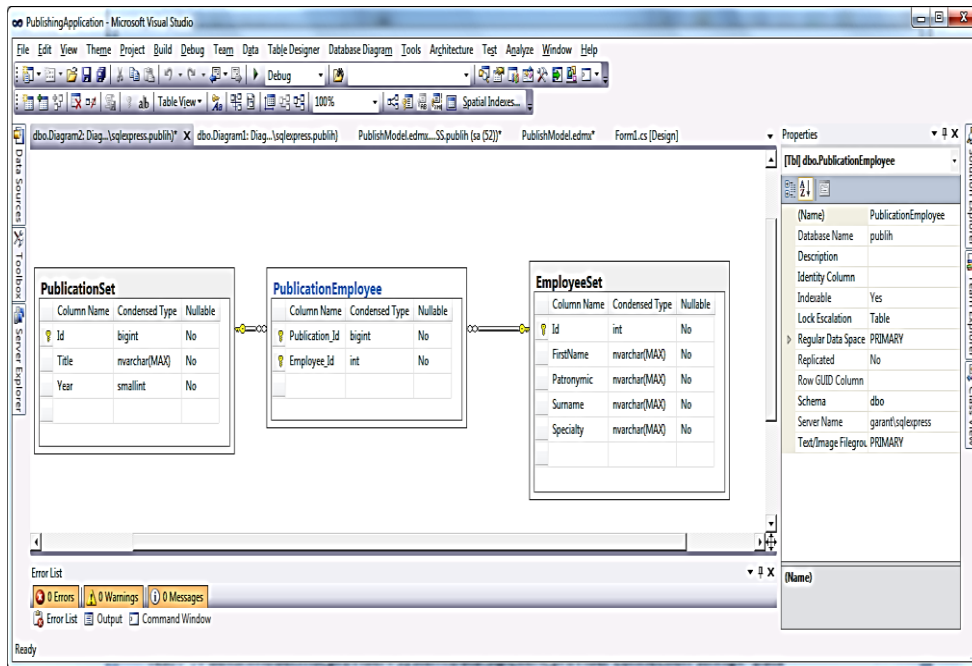


Рис. 37. Вікно діаграми після втілення зв'язку "багато до багатьох" засобами реляційної моделі

Під час створення реляційної моделі виникають деякі проблеми:

1. Не вдається пов'язати поля перетяганням.

Можливі причини. Пов'язувані поля мають різні типи. Серед пов'язуваних полів немає *Primary key*.

Вирішення. виправити помилки у структурі таблиць.

2. Не вдається редагувати створену таблицю (рис. 38).

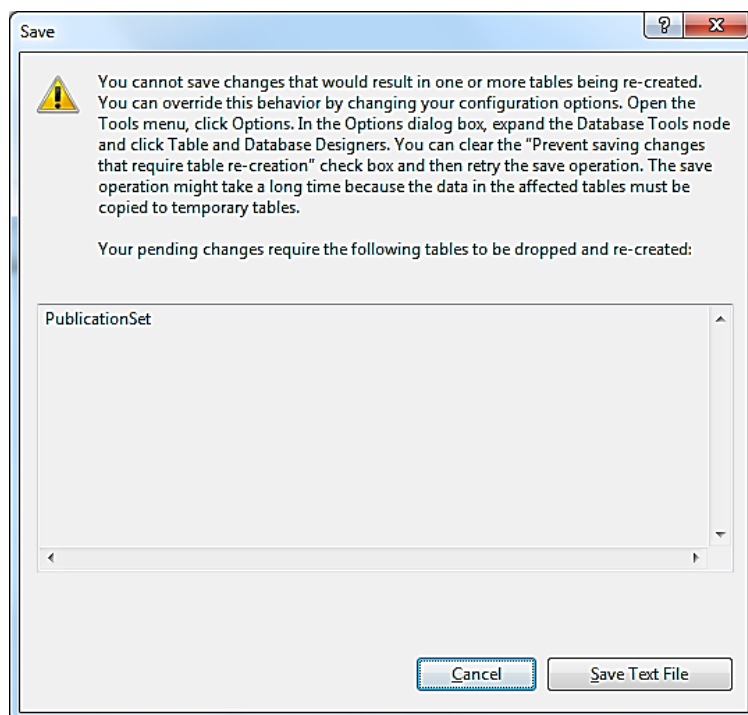


Рис. 38. Повідомлення про неможливість редагування опису стовпців

Можливі причини. Заборонено відтворення таблиці.
Вирішення. Дозволити відтворення (рис. 39, 40).

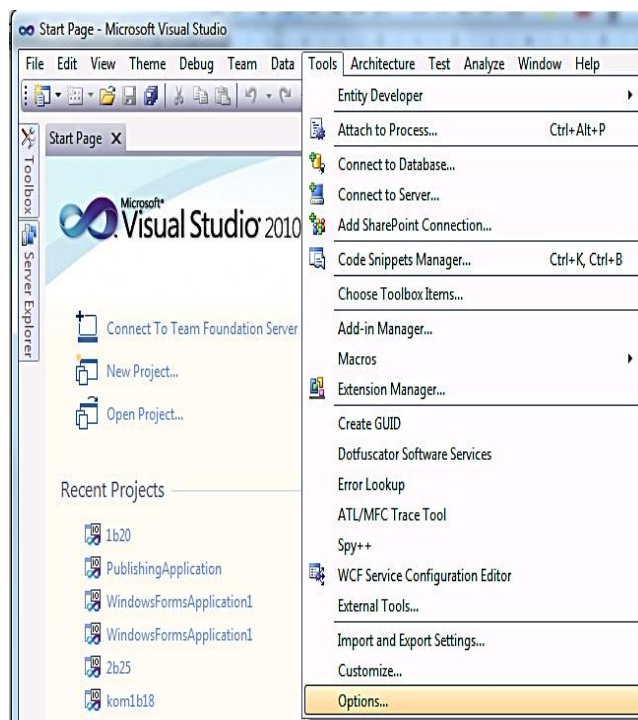


Рис. 39. Меню опцій

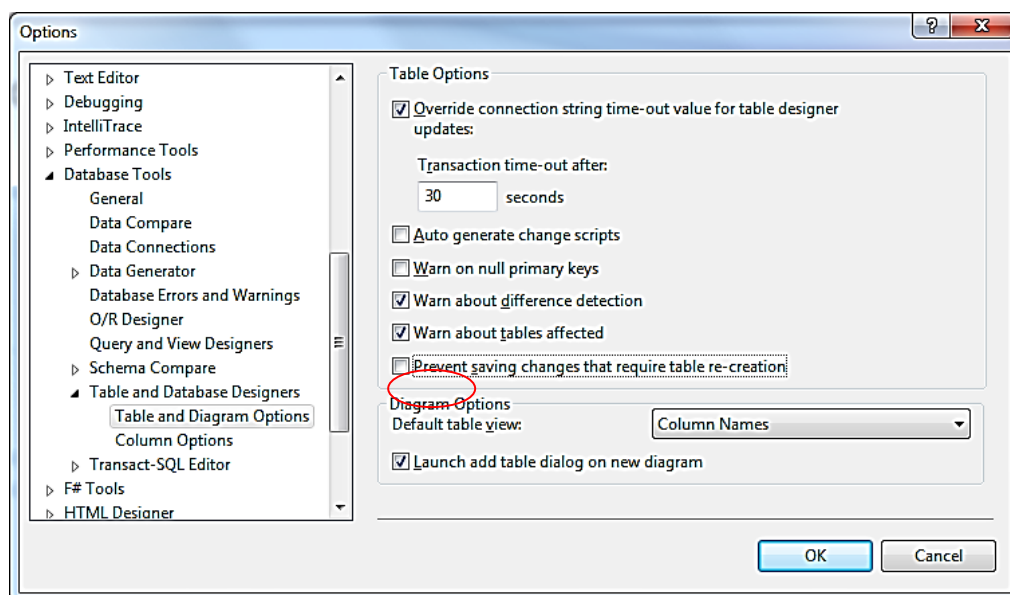


Рис. 40. Діалог про опції редагування таблиць

3. Таблицю, яку створено, не вдається зберегти.

Можливі причини. Відсутній первинний ключ *Primary key*.

Вирішення. Вибрати поля, із яких мав створитися первинний ключ *Primary key* і визначити їх ключовими.

Після створення та пов'язування таблиць потрібно заповнити їх даними.

Заповнення БД виконують у такому порядку.

Спочатку заповнюються довідкові, батьківські таблиці. Потім – ті, які містять посилання на них, потім таблиці, що містять посилання на вже заповнені і так далі.

Водночас необхідно стежити за тим, щоб було заповнено поля *NOT NULL*, а поля зовнішнього ключа (*FOREIGN KEY*) містили правильні посилання на первинні ключі (*PRIMERY KEY*). Поля *PRIMERY KEY Identity* заповнювати не потрібно. А *PRIMERY KEY* без *Identity* необхідно обов'язково заповнити – вони не можуть бути *NOT NULL*.

З'єднання із сервером – це стан оболонки *Visual Studio*, а не частина проекту.

Відкрийте вікно *Server Explorer*.

Якщо ви працюєте на іншій робочій станції, з'єднання необхідно створити наново (рис. 41, 42).

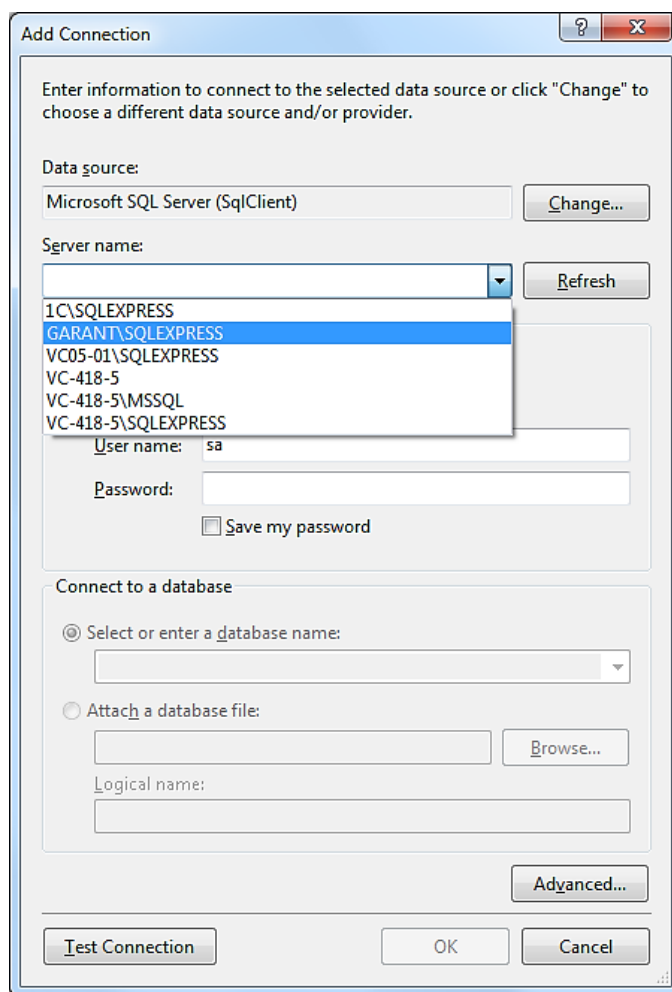


Рис. 41. Діалог щодо створення нового з'єднання. Вибір сервера СУБД

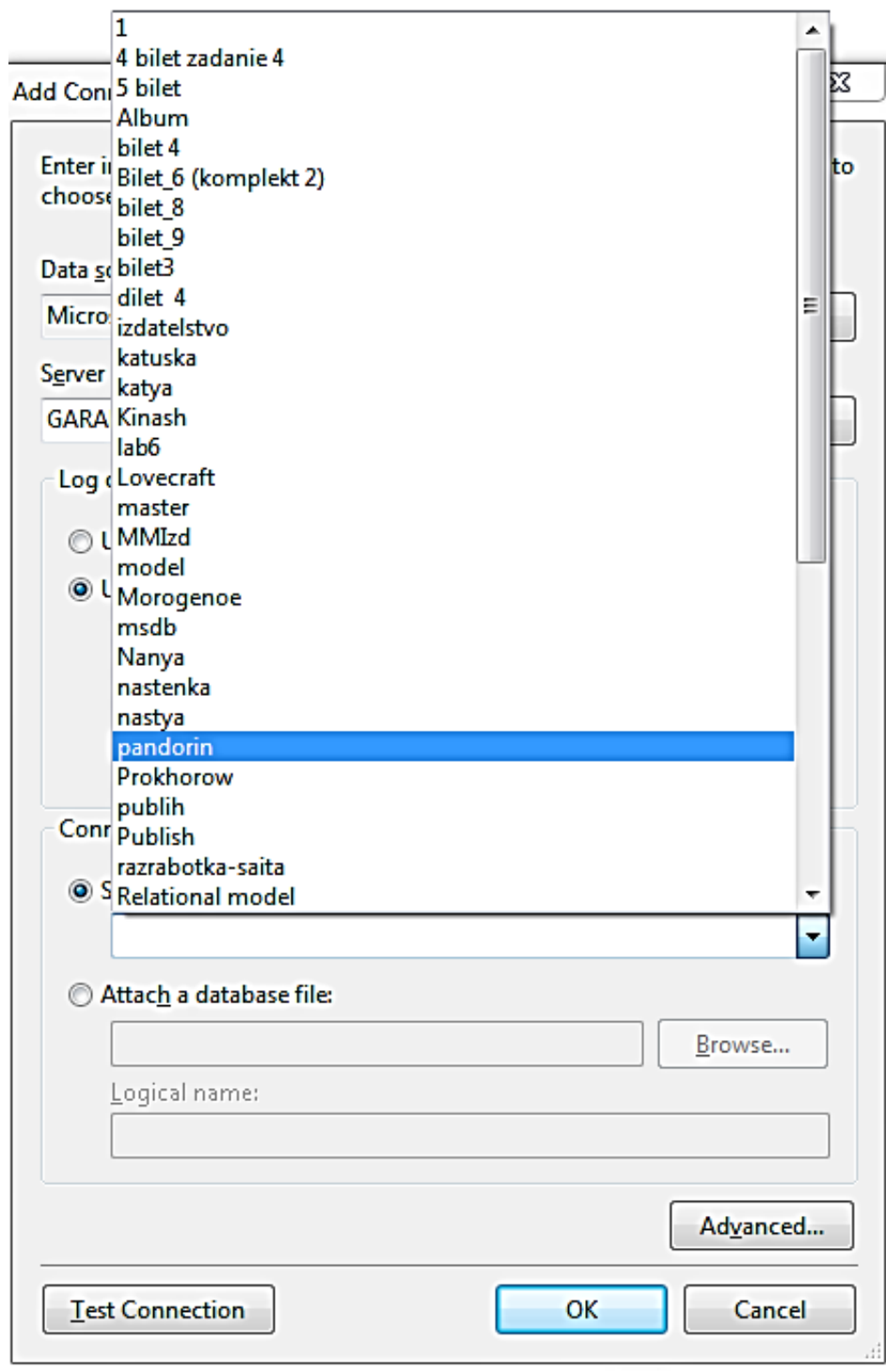


Рис. 42. Діалог щодо створення нового з'єднання.
Вибір сервера бази даних

Звичайно, цього робити не треба, якщо ви працюєте на тій самій робочій станції.

Розкрийте вузол із вашою базою даних.

Розкрийте вузол таблиці.

За допомогою контекстного меню перейдіть до редагування даних у таблиці (рис. 43, 44).

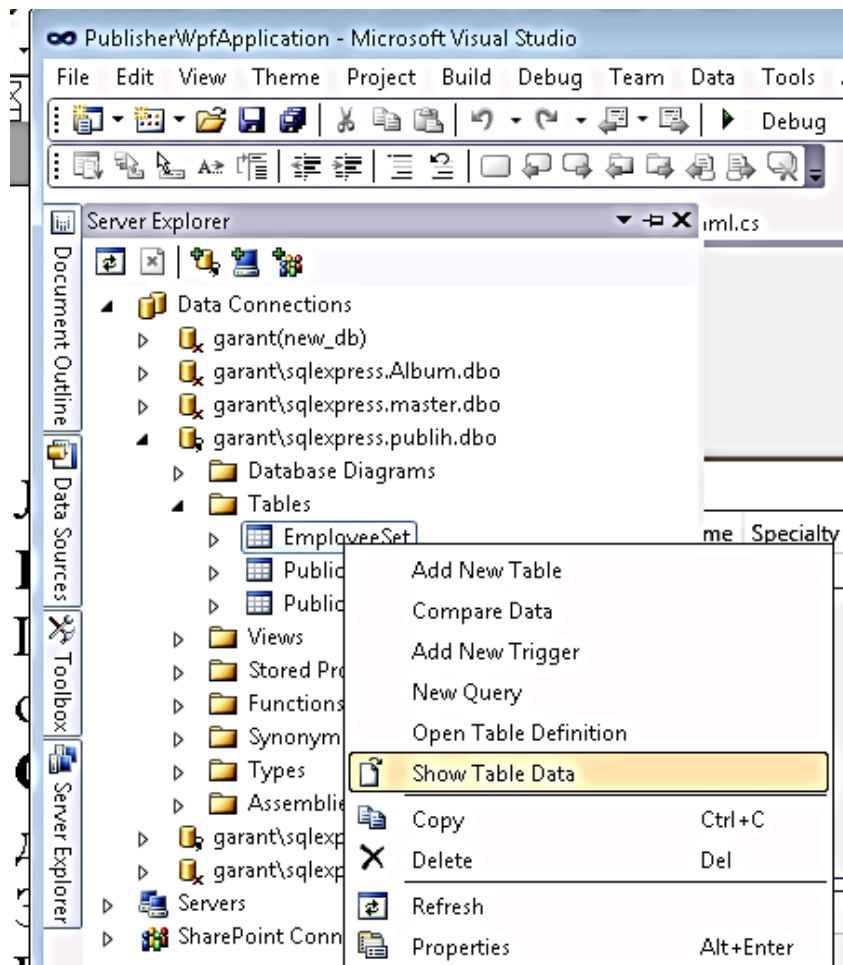


Рис. 43. Контекстне меню редагування даних у таблицях

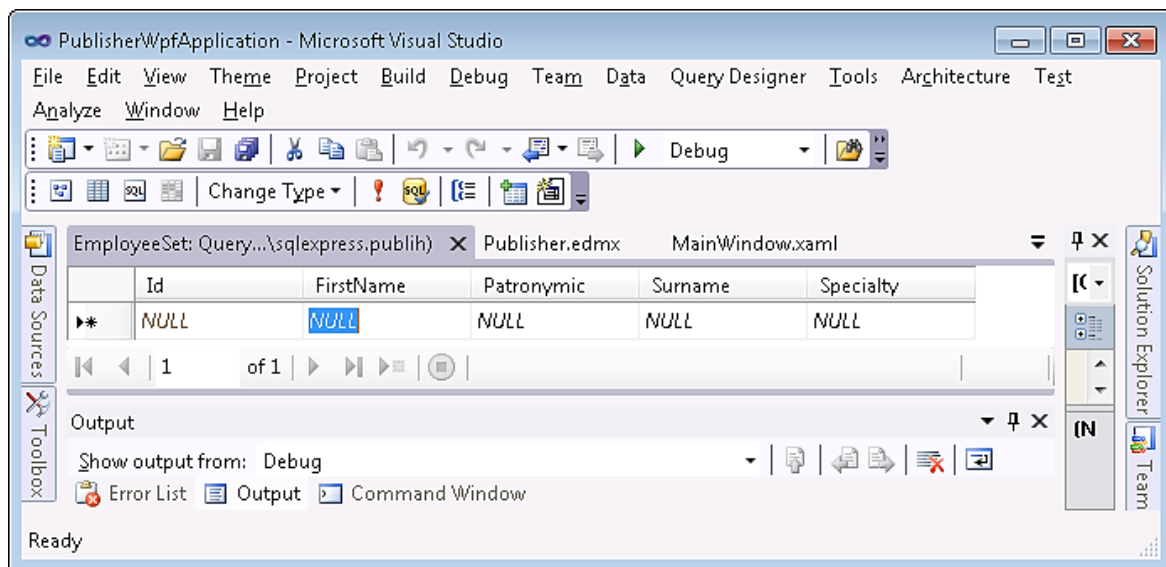


Рис. 44. Панель редагування даних у таблицях

Заповнюємо рядки. Не намагайтеся заповнити значення *Id* – його заповнить сервер БД.

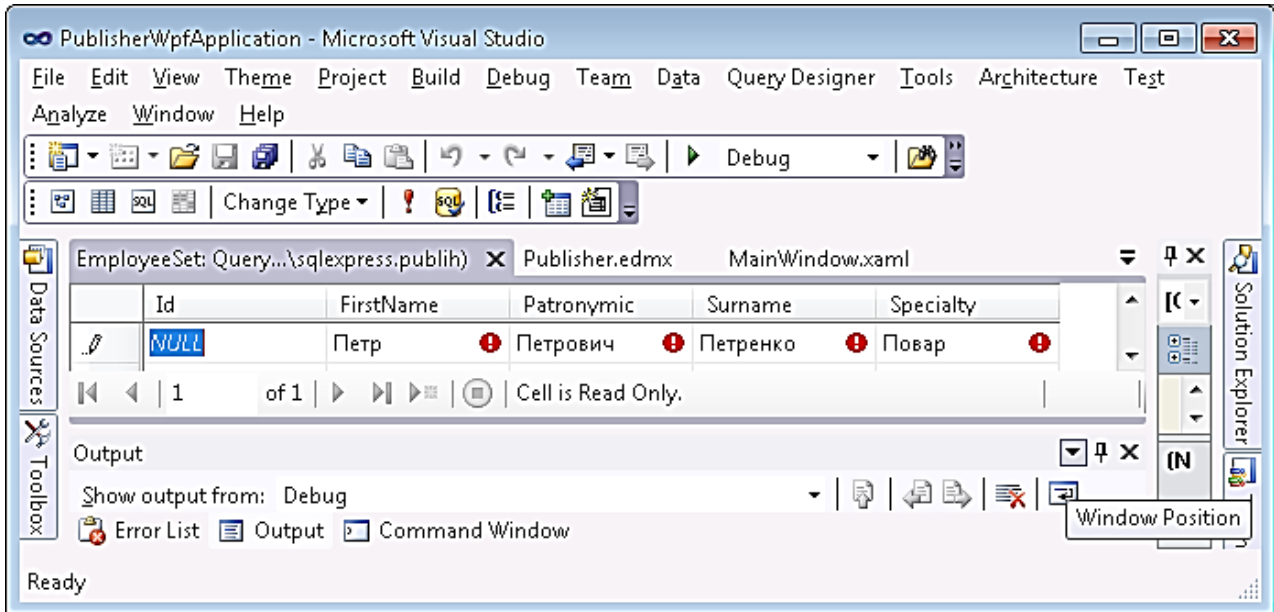


Рис. 45. Заповнений рядок таблиці "Співробітники"

Червоні знаки окликів означають, що кортеж-рядок не сформований правильно – доки відсутнє значення *Id* (рис. 45).

Після переходу до наступного рядка сервер заповнить його без вашої участі (рис. 46).

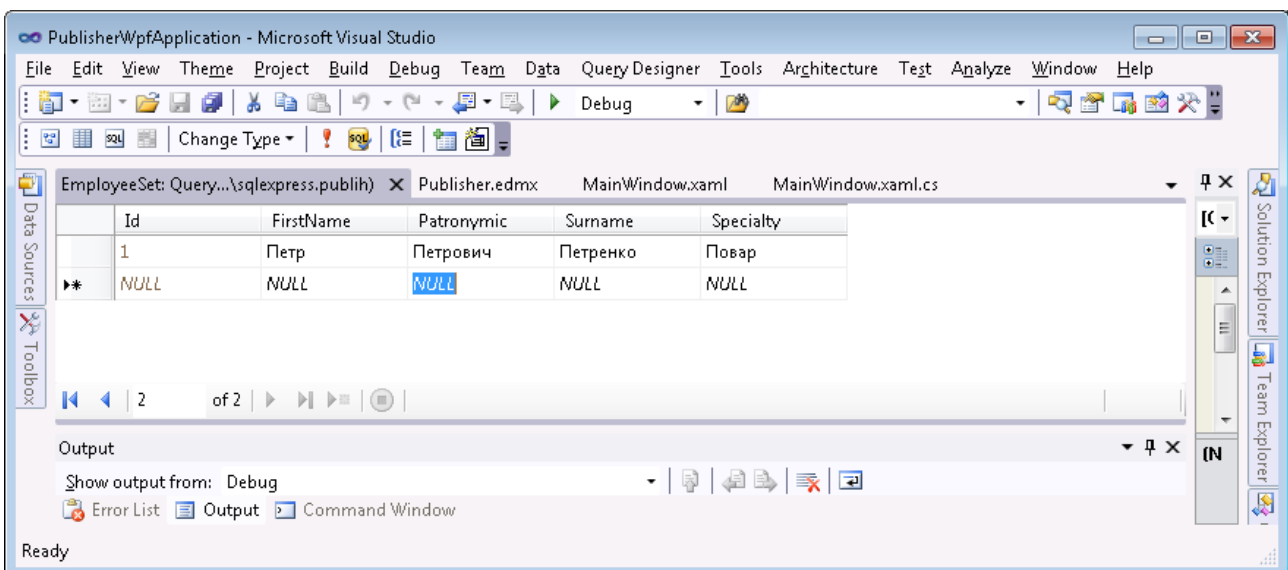
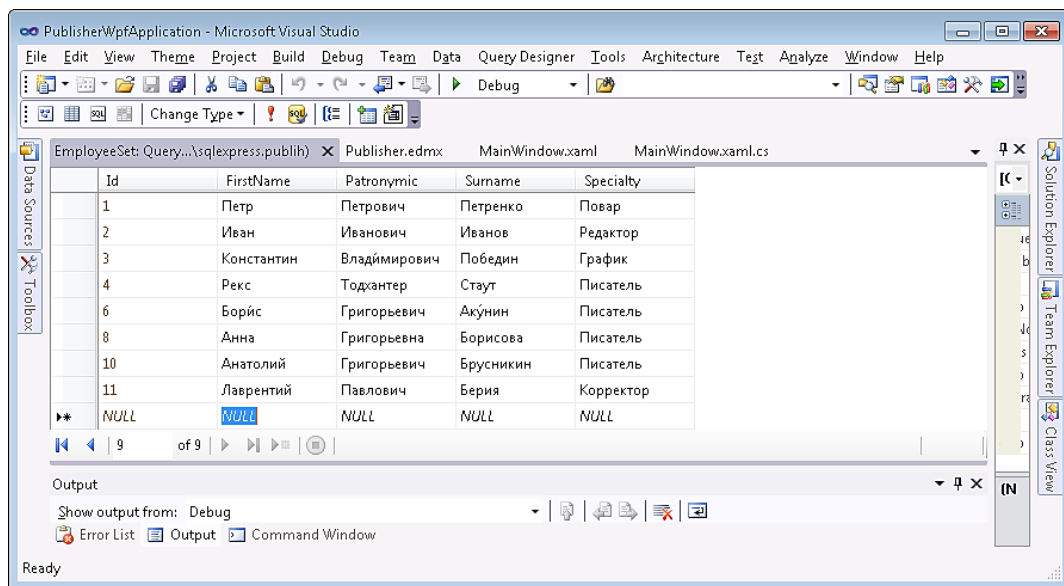


Рис. 46. Автоматичне заповнення значень у стовпці *Identity*

Заповнена таблиця має приблизно такий вигляд (рис. 47).

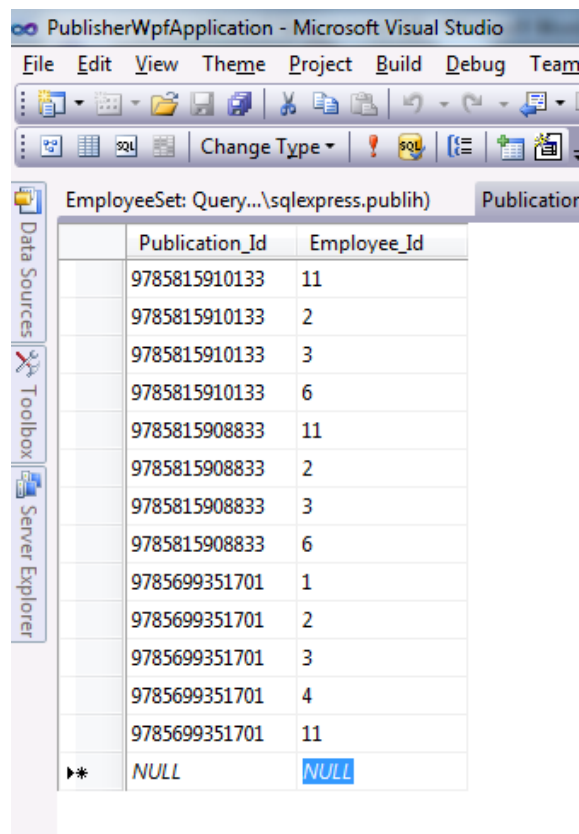


Id	FirstName	Patronymic	Surname	Specialty
1	Петр	Петрович	Петренко	Повар
2	Иван	Иванович	Иванов	Редактор
3	Константин	Владимирович	Победин	График
4	Рекс	Тодхантер	Стаут	Писатель
6	Борис	Григорьевич	Ахунин	Писатель
8	Анна	Григорьевна	Борисова	Писатель
10	Анатолий	Григорьевич	Брусникин	Писатель
11	Лаврентий	Павлович	Берия	Корректор
▶*	NULL	NULL	NULL	NULL

Рис. 47. Заповнена таблиця Співробітники

Заповнимо таблицю "Видання".

У ній стовпець *Id*, що презентує *ISBN PRIMARY KEY* без *Identity* необхідно обов'язково заповнити – він не може бути *NOT NULL* (рис. 48).



Publication_Id	Employee_Id
9785815910133	11
9785815910133	2
9785815910133	3
9785815910133	6
9785815908833	11
9785815908833	2
9785815908833	3
9785815908833	6
9785699351701	1
9785699351701	2
9785699351701	3
9785699351701	4
9785699351701	11
▶*	NULL

Рис. 48. Заповнена таблиця "Видання"

У разі спроби ввести значення, яке вже є ключем деякого запису, виникає повідомлення (рис. 49).

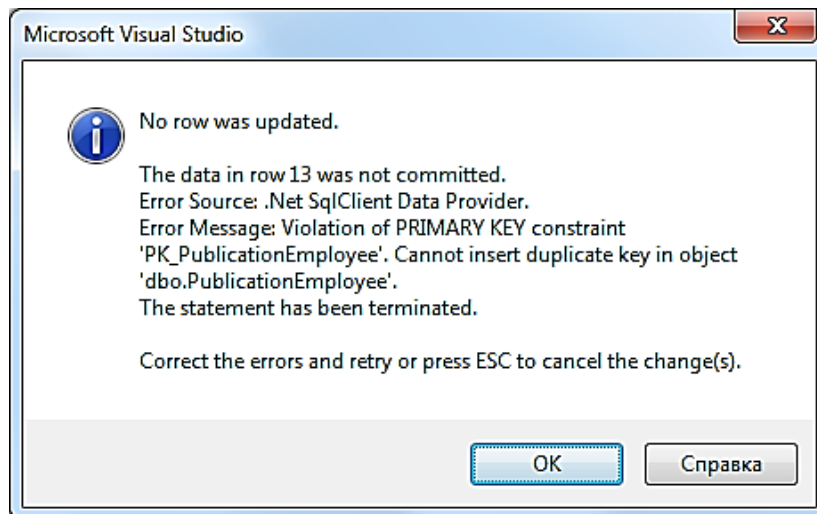
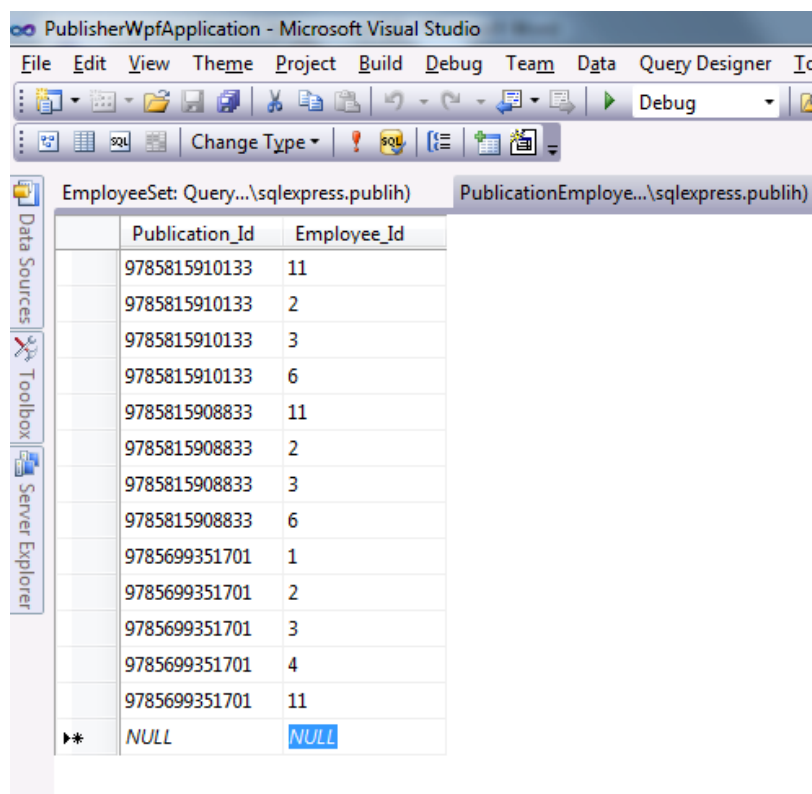


Рис. 49. Повідомлення про наявність запису із ключем, що вводиться

Заповнимо таблицю зв'язків, пам'ятаючи про те, що первинний ключ до неї складено зі значень двох стовпців (рис. 50).



	Publication_Id	Employee_Id
	9785815910133	11
	9785815910133	2
	9785815910133	3
	9785815910133	6
	9785815908833	11
	9785815908833	2
	9785815908833	3
	9785815908833	6
	9785699351701	1
	9785699351701	2
	9785699351701	3
	9785699351701	4
	9785699351701	11
**	NULL	NULL

Рис. 50. Заповнена таблиця зв'язків

Таким чином, оболонка *Visual Studio* надає потужні інструменти проектування і завантаження реляційних баз даних, що дозволяє помітно підвищити надійність інформаційних систем.

Контрольні запитання

1. Дайте визначення поняттям "елемент даних", "атрибут", "домен", "сегмент", "кортеж".
2. Дайте визначення поняттям "первинний атрибут", "описовий атрибут", "можливий ключ", "первинний ключ", "ключова детермінанта", "вторинний ключ", "складний ключ", "зовнішній ключ".
3. Дайте визначення поняттям "логічний і фізичний записи".
4. Наведіть відмінності в поняттях "логічний запис" і "екземпляр запису".

Рекомендована література

Основна

1. Автоматизовані системи. Терміни та визначення. ДСТУ 2226–93. – К. : Держстандарт України, 1994. – 32 с.
2. Бази даних. Терміни та визначення : ДСТУ 2874–94. – К. : Держстандарт України, 1995. – 32 с.
3. Дейт К. Дж. Введение в системы баз данных / К. Дж. Дейт. – 8-е изд. – К. : Диалектика, 2005. – 1328 с.
4. Ергономічне проектування робочих систем. Основні принципи : ДСТУ EN ISO 6385:2005. – К. : Держстандарт України, 2006. – 32 с.
5. Інформаційні системи та технології в економіці : посібник для студентів вищих навчальних закладів / за ред. В. С. Пономаренка. – К. : ВЦ "Академія", 2002. – 544 с.
6. Інформаційні технології. Основні напрямки оцінювання та відбору CASE-інструментів. ДСТУ 3919–99 (ISO/IEC 14102:1995). – К. : Держстандарт України, 2000. – 32 с.
7. Інформаційні технології. Процеси життєвого циклу програмного забезпечення : ДСТУ 3918–99 (ISO/IEC 12207:1995). – К. : Держстандарт України, 1999. – 32 с.

8. Інформаційні технології. Супровід програмного забезпечення : ДСТУ ISO/IEC 14764–2002. – К. : Держстандарт України, 2002. – 32 с.
9. Інформаційні технології. Словник термінів. ДСТУ ISO/IEC 2382–17:2005. Ч. 17. Бази даних. – К. : Держстандарт України, 2006. – 32 с.
10. Інформаційні технології. Словник термінів. ДСТУ ISO/IEC 2382–18:2005. Ч. 18. Розподілене оброблення даних. – К. : Держстандарт України, 2006. – 32 с.
11. Інформаційні технології. Система стандартів з баз даних. Еталонна модель керування даними : ДСТУ 3330–96 (ГОСТ 34.321–96). – К. : Держстандарт України, 1997. – 32 с.
12. Інформаційні технології. Система стандартів з баз даних. Концепція та термінологія для концептуальної схеми й інформаційної бази : ДСТУ 3329–96 (ГОСТ 34.320–96). – К. : Держстандарт України, 1997. – 32 с.
13. Кулямин В. В. Технологии программирования. Компонентный подход / В. В. Кулямин. – М. : МГУ, 2003. – 312 с.
14. Купцевич Ю. Microsoft ASP.NET, Web-сервисы, Web-приложения / Ю. Купцевич. – К. : Диалектика, 2003. – 400 с.
15. Папа Дж. Приложения для привязки данных с помощью ADO.NET и настраиваемые объекты / Дж. Папа. – MSDN Magazine, February 2007.
16. Оньон Ф. Основы ASP.NET с примерами на C# / Ф. Оньон. – К. : Диалектика, 2004. – 304 с.
17. Нортон А. C# 3.0_ Эволюция LINQ и его влияние на проект C# / А. Нортон. – MSDN Magazine, June 2005.
18. Пономаренко В. С. Проектування баз даних : навч. посіб. / В. С. Пономаренко, Л. А. Павленко, І. О. Максименко. – К. : ІЗМН, 1997. – 172 с.
19. Системи оброблення інформації. Основні поняття. Терміни та визначення. ДСТУ 2938–94. – К. : Держстандарт України, 1995. – 32 с.
20. Системи стандартів з баз даних. Структура системи словників інформаційних ресурсів. ДСТУ 3302–96. – К. : Держстандарт України, 1997. – 32 с.
21. Системи оброблення інформації. Керування процесами оброблення даних. Терміни та визначення. ДСТУ 2940–94. – К. : Держстандарт України, 1995. – 28 с.
22. Системи оброблення інформації. Розроблення систем. Терміни та визначення. ДСТУ 2941–94. – К. : Держстандарт України, 1995. – 20 с.

23. Системи оброблення інформації. Керування процесами. Оброблення даних. Терміни та визначення. ДСТУ 2940–94. – К. : Держстандарт України, 1995. – 32 с.

24. Система стандартів з баз даних. Мова баз даних SQL з розширенням цілісності. ДСТУ 3149–95. – К. : Держстандарт України, 1995. – 32 с.

25. Системи управління якістю. Вимоги. ДСТУ ISO 9001–2001. – К. : Держстандарт України, 2001. – 32 с.

26. Системи управління якістю. Настанови щодо поліпшення діяльності. ДСТУ ISO 9004–2001. – К. : Держстандарт України, 2001. – 32 с.

27. Системи управління якістю. Основні положення та словник. ДСТУ ISO 9000–2001. – К. : Держстандарт України, 2001. – 32 с.

Додаткова

28. Microsoft Corporation. Разработка Web-приложений на Microsoft Visual Basic .NET и Microsoft Visual C# .NET : учебный курс MCAD/MCSD. – М. : Издательско-торговый дом "Русская Редакция", 2003. – 704 с.

29. Microsoft Corporation. Разработка Web-сервисов XML и серверных компонентов на Microsoft Visual Basic .NET и Microsoft Visual C# .NET : учебный курс MCAD/MCSD. – М. : Издательско-торговый дом "Русская Редакция", 2004. – 576 с.

Інформаційні ресурси

30. Вендров А. М. Современные методы и средства проектирования информационных систем [Электронный ресурс]. – Режим доступа : www.citforum.ru/database/case/index.shtml CASE–технологии.

31. Керування даними [Электронный ресурс]. – Режим доступа : uk.wikipedia.org/wiki/Категорія:Керування_даними.

32. Разработка XNA [Электронный ресурс]. – Режим доступа : www.xnadev.ru

33. Садоян Р. ASP на блюдечке [Электронный ресурс] / Р. Садоян. – Режим доступа : www.fububy.na.by/index.php?elif=asp/aaa07d7ebeac7309678fbca4b7b69a78.htm.

34. Стандарт онтологического исследования IDEF5 [Электронный ресурс]. – Режим доступа : www.citforum.ru/cfin/idef/idef5.shtml.

35. Статьи по ASP [Электронный ресурс] – Режим доступа : www.fububy.na.by/index.php?tsrid=asp.

36. Хоффман Кевин LINQ to Entities vs. LINQ to SQL – что когда использовать? [Электронный ресурс] / Кевин Хоффман. – Режим доступа : www.optim.su/issues.asp.

37. Vista для разработчика [Электронный ресурс]. – Режим доступа : www.thevista.ru/.

Зміст

Вступ.....	3
Модуль 1. Базові концепції систем управління базами даних.....	4
Лабораторна робота 1. Визначення сфери застосування, збирання інформації про стан предметної сфери, складання функціональної схеми предметної сфери, аналіз вимог оброблення	4
Лабораторна робота 2. Концептуальне проектування.....	5
Лабораторна робота 3. Фізичне проектування	19
Рекомендована література.....	39
Основна.....	39
Додаткова.....	41
Інформаційні ресурси	41

НАВЧАЛЬНЕ ВИДАННЯ

**Методичні рекомендації
до виконання лабораторних робіт
з навчальної дисципліни
"ПРИНЦИПИ ПРОЕКТУВАННЯ БАЗ ДАНИХ
ТА БАЗ ЗНАНЬ"**

**для студентів напряму підготовки
6.051501 "Видавничо-поліграфічна справа"
всіх форм навчання**

Самостійне електронне текстове мережне видання

Укладач **Пандорін** Олександр Костянтинович

Відповідальний за випуск *Пушкар О. І.*

Редактор *Доценко О. Г.*

Коректор *Міхно В. В.*

План 2015 р. Поз. № 28 ЕВ. Обсяг 44 с.

Видавець і виготівник – ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Леніна, 9-А

*Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру
ДК № 4853 від 20.02.2015 р.*