Лабораторна робота 5.

Побудова логічної і фізичної моделі бази даних

CASE-засобами

Цілі роботи

1. Вивчити методологію IDEF1X.

2. Освоїти інструментарій ERwin.

3. Ознайомитися з технологією побудови логічної моделі в ERwin.

4. Вивчити методи визначення ключових атрибутів сутностей.

5. Освоїти метод перевірки адекватності логічної моделі.

6. Вивчити типи зв'язків між сутностями.

7. Вивчити види нормальних форм.

8. Освоїти роль CASE-засобу ERwin у нормалізації і денормалізації

бази даних.

9. Побудувати фізичну модель.

10. Вивчити види звітів (для самостійного вивчення).

11. Освоїти процедуру створення звітів (для самостійного опрацювання).

12. Вивчити експортування, збереження і друкування звітів (для самостійного опрацювання).

Перед виконанням роботи студент повинен знати:

поняття семантичного моделювання предметної області;

основні конструктивні елементи ER-моделі;

поняття ключа і їх види;

типи зв'язків, що застосовуються в ER-моделюванні предметної

області;

поняття залежної і незалежної сутності;

основні нотації, що використовуються для зображення ER-діаграм.

Після виконання лабораторної роботи студент повинен уміти:

аналізувати предметну область і синтезувати її модель з використанням CASE-технологій;

самостійно формувати прості і складні ER-моделі предметної області;

проводити пряме і зворотне проектування схеми бази даних;

визначати стратегії підтримки посилальної цілісності бази даних;

формувати звіти по сформованій моделі предметної області.

97

Підготовча частина

Хід роботи

5.1. Підготовчий етап.

5.2. Базовий рівень лабораторної роботи: "Побудова найпростішої

ER-моделі і проведення прямого проектування БД".

5.3. Розширений рівень лабораторної роботи: створення моделі

предметної області "ІС компанії з продажу товарів".

Форма звітності

За результатами виконання роботи слід оформити електронний звіт засобами Word. В електронному звіті за кожним завданням надати текст завдання з коментарями за формою, що подана в наведених нижче прикладах.

Критерії оцінювання

У 12-бальній системі оцінка результату захисту лабораторної роботи формується за такими правилами:

1. За підготовчий етап і тест з основних понять лабораторної роботи (Інтернет доступ: https://www.it-karkas.com.ua/bd3/indextest.php) може

бути виставлено від 0 до 2 балів.

2. За базовий рівень лабораторної роботи може бути виставлено

від 0 до 4 балів.

3. За розширений рівень лабораторної роботи може бути виставлено від 0 до 6 балів.

4. За запізнення із захистом лабораторної роботи знімається 2 бали

за кожний тиждень.

Рекомендована література: [10; 11; 20].

Основні поняття

Одним з найбільш популярних засобів інформаційного моделювання

та автоматизації є CASE-засіб (Computer-Aided Software Engineering) – ERwin

Data Modeler фірми Computer Associates. Реалізація моделювання в ERwin

базується на теорії реляційних баз даних і на методології IDEF1X.

Логічний (інфологічний) рівень проектування бази даних припускає,

що ми мислимо в поняттях реального світу та безпосередньо з нього беремо об'єкти для моделювання.

Фізичний (даталогічний) рівень проектування бази даних передбачає

інтерпретацію логічного рівня в термінах фізичної бази даних, включаючи

вибір СУБД, типів даних за замовчуванням і ефективних схем індексування.

98

Діаграма "сутність – зв'язок" ERD (Entity-Relationship Diagram)

призначена для розробки моделі даних і забезпечує графічний засіб визначення даних і відношень між ними.

Основними складовими моделі IDEF1X є: люди, предмети, явища, про

які зберігається інформація (надалі – сутності), зв'язки між цими елементами

(надалі – відношення), характеристики цих елементів (надалі – атрибути).

Сутність – це множина реальних або абстрактних об'єктів (людей,

місць, подій), яким притаманні загальні атрибути або характеристики.

Будь-який об'єкт системи може бути представлений тільки однією сутністю, яка повинна бути унікально ідентифікована.

Відношення – зв'язок між двома та більше сутностями. Іменування

відношення здійснюється за допомогою граматичної форми дієслова

(має, визначає, ...).

Таким чином: сутності – це базовий тип інформації, що зберігається

в базі даних, а відношення показують, як ці типи даних взаємопов'язані між собою. Сутність повинна мати унікальне ім'я і іменуватися іменником в однині.

Сутність має один або кілька атрибутів, які однозначно ідентифікують

кожен зразок сутності та називаються ключем (складеним ключем). Кожна сутність може мати будь-яку кількість відношень з іншими сутностями.

Якщо зовнішній ключ цілком використовується в складі первинного

ключа, то сутність є залежною від ідентифікатора.

У нотації IDEF1X сутність зображується у вигляді прямокутника.

Залежно від рівня подання даних можуть бути деякі відмінності.

Розрізняють такі рівні подання сутності: діаграма "сутність – зв'язок"

(ERD); модель даних, заснована на ключах (KB); повна атрибутивна модель (FA). Кожен атрибут кожної сутності має унікальне ім'я. Сутність

може мати будь-яку кількість атрибутів.

Розрізняють власні та успадковані атрибути. Власні атрибути є

унікальними в рамках моделі. Успадковані передаються від сутності-

"батька" шляхом визначення ідентифікованого зв'язку.

Ключовий елемент таблиці (ключ) – таке її поле (простий ключ)

або рядкове вираження, утворене зі значень декількох полів (складений

ключ), через який можна визначити значення інших полів для однієї або

кількох записів таблиці.

На практиці для використання ключів створюються індекси – службова інформація, що містить упорядковані відомості про ключові значення. У реляційній теорії і концептуальній моделі поняття "ключ" застосовується для атрибутів відношення або сутності.

99

Незалежна сутність IDEF1X – це незалежні дані, які завжди присутні в системі. Водночас відношення з іншими сутностями можуть як існувати, так і бути відсутніми.

Залежна сутність IDEF1X подає дані, залежні від інших сутностей

в системі; тому вона завжди повинна мати відносини з іншими сутностями.

У методі IDEF1X усі сутності розрізняють на залежні та незалежні

від ідентифікаторів. Сутність є незалежною від ідентифікаторів або просто

незалежною, якщо кожен її екземпляр може бути однозначно ідентифікований без визначення його відношень з іншими сутностями. Сутність називають залежною від ідентифікаторів або просто залежною, якщо однозначна ідентифікація примірника сутності залежить від його відношення

до іншої сутності. Незалежна сутність зображується у вигляді звичайного

прямокутника, залежна – у вигляді прямокутника з закругленими кутами.

У IDEF1X існують такі види потужностей зв'язків:

N потужність – кожен екземпляр сутності-батька може мати нуль,

один або більше одного пов'язаного з ним екземпляра сутності-нащадка

(за замовчуванням);

Р потужність – кожен екземпляр сутності-батька повинен мати

не менше одного пов'язаного з ним екземпляра сутності-нащадка;

Z потужність – кожен екземпляр сутності-батька повинен мати

не більше одного пов'язаного з ним екземпляра сутності-нащадка;

конкретне число – кожен екземпляр сутності-батька пов'язаний

з деяким фіксованим числом екземплярів сутності-нащадка.

За замовчуванням потужність зв'язків приймається дорівненою N.

Зв'язок "один-до-одного" означає, що екземпляр однієї сутності пов'язаний тільки з одним екземпляром іншої сутності. Зв'язок 1:М означає,

що один екземпляр сутності, розташований ліворуч лінії зв'язка, може

бути пов'язаний з декількома екземплярами сутності, розташованими

праворуч. Зв'язок "багато-до-багатьох" (М:М) означає, що один екземпляр першої сутності може бути пов'язаний з декількома екземплярами

другої сутності, і навпаки, один екземпляр другої сутності може бути пов'язаний з декількома екземплярами першої сутності.

Зв'язок зображується лінією, що проводиться між сутністю-батьком

і сутністю-нащадком, з точкою на кінці лінії у сутності-нащадка.

Якщо екземпляр сутності-нащадка однозначно визначається своїм

зв'язком з сутністю-батьком, то зв'язок називають ідентифікуючим, інакше – неідентифікуючим.

Ідентифікуючий зв'язок зображується суцільною лінією, неідентифікуючий – пунктирною лінією.

100

Атрибути зображуються у вигляді списку імен усередині блоку сутності. Атрибути, що визначають первинний ключ, розміщуються напочатку та відділяються від інших атрибутів горизонтальною рискою.

Сутності можуть мати також зовнішній ключ (Foreign Key), який використовується як або первинний ключ або неключовий атрибут. Для позначення зовнішнього ключа всередині блоку сутності поміщають імена

атрибутів, після яких записують букви FK у дужках.

ERwin – CASE-засіб для проектування та документування баз даних, який дозволяє створювати, документувати та супроводжувати бази

даних, сховища і вітрини даних (версія Erwin 4.0 Build 1298).

У ERwin'і зі встановленням ідентифікованого зв'язку атрибути первинного ключа батьківської сутності автоматично переносяться до складу первинного ключа дочірньої сутності. Цю операцію називають міграцією атрибутів. У дочірньої сутності нові атрибути позначаються як

зовнішній ключ (FK). Зі встановленням неідентифікуючого зв'язку атрибути первинного ключа батьківської сутності мігрують до складу неключових полів дочірньої сутності.

Робота з програмою починається зі створення нової моделі, для якої

потрібно вказати тип і цільову СУБД (рис. 5.1).

Рис. 5.1. Початковий вид вікна ERwin

ERwin дозволяє створювати логічну, фізичну моделі та модель,

яка поєднує логічний і фізичний рівні.

Для створення нової моделі необхідно вказати її тип: логічний, фізичний або логічно/фізичний (рис. 5.2).

101

Рис. 5.2. Вибір типу нової моделі

Після вибору режиму створення моделі або відкриття вже існуючої

моделі, палітра інструментів стає доступною.

Далі буде розглядатися робота з ERwin в нотації IDEF1X.

Примітка. У створеній моделі з налаштуванням за замовчуванням некоректно відображаються українські символи. Щоб усунути цей недолік, необхідно підкоригувати використані в моделі шрифти. Для цього необхідно зайти в меню Format -> Default Fonts & Colors,

послідовно пройтися усіма вкладками, вибравши будь-який шрифт, назва якого закінчується на CYR (наприклад, Arial CYR), і виставити перемикач Apply To в значенні All Objects.

Практична частина

5.1. Підготовчий етап

Завдання

Створіть порожню базу даних у СУБД Access 2013.

Примітка:

1) ця база даних буде автоматично заповнюватися порожніми таблицями у п. 5.2;

2) оскільки робоча версія програми ERwin, що використовується в лабораторній

роботі, створена давно, вона може співпрацювати з базою даних у форматі Access 2000.

Виконання

1. Створіть порожню базу даних у СУБД Access 2013.

2. Запустіть Access 2013, у правому кутку виберіть Нова база даних

і клацніть значок Створити.

3. Збережіть нову базу як БД Access 2000, потім укажіть ім'я і місце

розташування і клацніть Зберегти.

102

5.2. Базовий рівень лабораторної роботи:

"Побудова найпростішої ER-моделі

та проведення прямого проектування БД"

Спроектувати просту інформаційну систему, яка базується на розглянутій у лабораторній роботі 4 системі обліку продажу хлібобулочних виробів.

5.2.1. Створення моделі

Завдання

Створіть нову модель в ERwin.

Виконання

1. Запустіть Erwin.

2. Виберіть Create a new model (Створіть нову модель).

3. Виберіть тип моделі Logical/Physical. Це дозволить надалі легко

перемикатися між логічним і фізичним рівнями. У списку пропонованих

баз даних виберіть Access. Пізніше, у разі потреби, середовище реалізації можна буде змінити.

5.2.2. Створення сутності

Завдання

Створіть сутність ТОВАРИ в ERwin.

Примітка: На фізичному рівні сутність подають таблицею, атрибут – колонкою

таблиці, екземпляр – рядком таблиці. Для наведеного прикладу виділіть такі основні

логічні об'єкти: Товари та Виробники.

Виконання

1. Клацніть значок на панелі інструментів. Курсор набере вигляду .

2. Клацніть на будь-якому місці діаграми. На екрані з'явиться нова

сутність.

3. За замовчуванням, сутності надається ім'я E/x, де x – унікальний

номер сутності. Змінити ім'я можна, клацнувши на сутності лівою кнопкою

миші або вибравши пункт Entity Properties у контекстному меню сутності.

4. У властивостях сутності також можна дати опис сутності, ввести

замітки та вказати іншу інформацію.

Примітка. Обов'язково складіть описи створюваних сутностей. Це допоможе

надалі зрозуміти, що це за об'єкт. Зробіть схему читаною для інших учасників процесу розроблення. Крім того, побудувавши після створення діаграми звіт, можна

отримати готову документацію за схемою.

103

Для внесення опису та зауважень виберіть в контекстному меню

сутності пункт меню Entity Properties. У полі Definition вкажіть визначення сутності.

На фізичному рівні визначення сутності можна експортувати як частину схеми і використовувати в реальній базі даних.

5.2.3. Створення атрибутів

Завдання

Створіть атрибути для сутності "ТОВАРИ".

Виконання

1. Виділіть сутність.

2. Клацніть правою кнопкою і виберіть Attributes в контекстному

меню сутності. З'являється діалог Attributes.

3. Щоб додати новий атрибут клацніть кнопку New, введіть ім'я атрибута,

ім'я колонки до бази даних і його домен. Домен атрибута використовується

під час визначення типу колонки на фізичному рівні. Для атрибутів первинного ключа у вкладці General зробіть позначку у вікні вибору Primary Key.

Аналогічно сутностям, необхідно задавати опис атрибутам (вкладка

Definition). Далі створіть сутність ВИРОБНИКИ й атрибути для неї.

Якщо встановити рівень перегляду атрибутів (Attribute level), загальний вид екрану в цьому випадку виглядатиме як на рис 5.3.

Рис. 5.3. Вид екрану для двох створених сутностей

104

5.2.4. Створення зв'язку

Завдання

Створіть зв'язок "Виробники виробляють Товари".

Виконання

1. Клацніть на одній з кнопок на панелі інструментів.

2. Клацніть на батьківській сутності.

3. Клацніть на дочірній сутності.

Примітка. У цьому прикладі ми не вводимо ідентифікуючих зв'язків. Кожен

екземпляр усіх сутностей однозначно визначається своїм номером. У ілюстрованому прикладі зв'язок "Виробники виробляють Товари" матиме зв'язок "багато-добагатьох", який можна створити тільки на рівні логічної моделі. За замовчуванням

дієслівна фраза не відображається на екрані. Для того щоб показати на екрані дієслівну фразу, що належить до зв'язку, клацніть правою кнопкою миші на будьякому місці у вікні діаграми та виберіть пункт Relationship Display -> Verb Phrase

(рис. 5.4).

Рис. 5.4. Відображення зв'язку на діаграмі

105

5.2.5. Створення фізичної моделі даних

Завдання

Позбавтесь від зв'язків "багато-до-багатьох".

Виконання

1. Виділіть зв'язок і запустіть майстер перетворення зв'язка "багатодо-багатьох" .

2. Після запуску майстра в його початковому вікні клацніть кнопку

Далі і перейдіть у вікно Transform information, де вкажіть ім'я перетворення та його опис.

3. Клацніть кнопку Далі, а потім Готово. У результаті діаграма виглядатиме як на рис. 5.5.

Рис. 5.5. Вид створеної діаграми

після перетворення зв'язку

Примітка. З'являється нова сутність "ПРОДАЖІ" така, що забезпечує асоціативний зв'язок між виробниками та товарами. Поява такої сутності позбавляє нас від

зв'язку "багато-до-багатьох". Первинні ключі початкових сутностей стають складеним

первинним ключем нової асоціативної сутності.

4. Після цього додайте до сутності "ПРОДАЖІ" нові атрибути: Дата

і Кількість.

106

5.2.6. Генерація фізичної схеми бази даних

Завдання

Виконайте генерацію фізичної схеми бази даних (Forward Engineering).

Виконання

1. Для генерації системного каталогу бази даних перейдіть з логічної моделі на фізичну, потім виберіть пункт меню Tools -> Forward

Engineer/Schema Generation.

2. Для генерації схеми бази даних клацніть кнопку Generate та увійдіть до діалогу Connection для встановлення сеансу зв'язку з сервером

і для запуску SQL-скрипта (рис. 5.6).

3. Клацніть кнопку Connect. Відбувається запуск процесу генерації

таблиць у вибраній базі даних.

4. Відкривши створену базу даних в ACCESS, побачимо схему,

що зображена рис. 5.7.

Рис. 5.6. Вікно вибору бази даних

під час прямого проектування

107

Рис. 5.7. Схема створеної бази даних у ACCESS 2013

5.3. Розширений рівень лабораторної роботи:

створення моделі предметної області

"ІС компанії з продажу товарів"

Завдання

Спроектуйте інформаційну систему для компанії, що провадять

продажі товарів.

Приклад виконання

Постановка задачі: створення моделі предметної області

"ІС компанії з продажу товарів"

Автоматизуються такі бізнес процеси.

1. Номенклатура продукції, що продається, складає близько

10 000 найменувань. Товари об'єднуються в ієрархічно організований каталог з необмеженою кількістю рівнів ієрархії.

2. Замовлення на купівлю товару приймають по телефону й обробляють менеджери організації. Замовлення включає необмежену кількість

товарів. Покупцем може виступати фізична або юридична особа.

108

3. Замовлення доставляють кур'єри (у разі доставки по місту) і транспортні компанії (у разі доставки в інші регіони країни). Відвантаження товарів оформлюється витратною накладною.

4. Увесь товар ураховується на складі. Постачання товару здійснюється виробниками й оформлюється прибутковими накладними.

5. У системі має бути передбачено ведення бази даних, яка повинна зберігати інформацію про покупців, виробників, товари, замовлення,

служби доставки.

5.3.1. Створення моделі (аналогічно до завдання 5.2.1)

5.3.2. Створення сутності (аналогічно до завдання 5.2.2)

Виділіть такі логічні об'єкти: Замовлення, Товар, Каталог, Покупець,

Служба доставки, Виробник, Накладна (об'єкти можна назвати базовими).

5.3.3. Створення атрибутів (аналогічно до завдання 5.2.3)

Для кожної сутності визначимо перелік атрибутів і їх властивості.

Для сутності "ЗАМОВЛЕННЯ" створіть атрибути: Код\_Замовлення,

Дата\_Замовлення, Статус\_Замовлення, Спосіб\_Замовлення, Спосіб\_Оплати, Спосіб\_Доставки, Вартість\_Доставки, Покупець.

Для сутності "ПОКУПЕЦЬ" створіть атрибути: Код\_Покупця, ПІБ,

Телефон, Місто, Адреса.

Для сутності "ТОВАР" створіть атрибути: Код\_Товар, Назва, Модель, Виробник, Ціна, Од\_виміру.

Для сутності "ВИРОБНИК" створіть атрибути: Код\_виробника, Виробник, Телефон, Адреса.

Для сутності "СЛУЖБА ДОСТАВКИ" створіть атрибути: Код\_служби\_доставки, Назва\_служби.

Для сутності "НАКЛАДНА" створіть атрибути: Код\_накладної, Дата\_накладної, Тип\_накладної.

Для сутності "КАТАЛОГ" створіть атрибути: Код\_рубрики, Послідовність\_ієрархі, Назва.

Для сутності "ЗАМОВЛЕННЯ\_ТОВАР" створіть атрибути: Код\_замовлення\_товар, Кількість.

Таким чином буде створена діаграма сутностей (рис. 5.8).

109

Рис. 5.8. Діаграма сутностей без зв'язків

5.3.4. Створення зв'язків

Завдання

Створіть:

неідентифікуючий зв'язок "Покупець робить Замовлення";

зв'язок "багато-до-багатьох" між сутностями "ЗАМОВЛЕННЯ"

і "ТОВАР";

рекурсивний зв'язок для сутності "КАТАЛОГ".

Виконання

1. Клацніть на кнопці на панелі інструментів.

2. Клацніть на батьківській сутності.

3. Клацніть на дочірній сутності.

Примітка. Якщо немає інформації про покупця, то інформація про замовлення

не має сенсу. Тобто наявність атрибута Номер покупця в Замовленні обов'язкова.

Така ситуація обумовлена обов'язковістю зв'язка (Властивість Nulls).

Для того щоб установити обов'язковість зв'язка, необхідно клацнути

правою кнопкою миші на зв'язку та вибрати пункт Relationship Properties

в контекстному меню.

У блоці Nulls вибрати позицію No Nulls. В цьому випадку значок

на початку стрілки зникне (рис. 5.9).

110

Рис. 5.9. Відображення обов'язкового зв'язку

Прикладом зв'язка "багато-до-багатьох" може слугувати зв'язок між

сутностями "ЗАМОВЛЕННЯ" і "ТОВАР". Одне замовлення може включати багато товарів, а один товар може входити у багато замовлень.

Для редагування властивостей зв'язку слід клацнути правою кнопкою миші на зв'язку та вибрати пункт Relationship Properties у контекстному меню.

Наприклад, можна присвоїти зовнішньому ключу Код\_покупця

в сутності "ЗАМОВЛЕННЯ" функціональне ім'я Покупець (рис. 5.10).

Рис. 5.10. Вікно опису – функціональне ім'я Покупець

111

За замовчуванням на діаграмі відображується тільки ім'я ролі. Щоб

відобразити повне ім'я, що включає ім'я ролі та базове ім'я атрибуту, розділені точкою, необхідно в контекстному меню діаграми вибрати пункт

Entity Display -> Rolename/Attribute.

Ім'я ролі необхідно вказувати обов'язково в рекурсивних зв'язках,

коли та сама сутність є водночас і батьківською, і дочірньою.

Прикладом такого зв'язка може слугувати сутність "КАТАЛОГ". Тут

рекурсивним зв'язком відображується входження рубрики каталогу в іншу рубрику цього ж каталогу. Зовнішній ключ є номером рубрики та відіграє тут роль підрубрики (рис. 5.11).

Рис. 5.11. Створення рекурсивного зв'язка

5.3.5. Створення ключів

Завдання

Виберіть атрибут до складу ключової сутності "ТОВАР". Створіть

альтернативний ключ у сутності "ТОВАР".

Виконання

Для виконання завдання необхідно дотримуватись такої послідовності дій.

112

1. Виберіть необхідну сутність – "ТОВАР".

2. Наведіть курсор на атрибут, який треба внести до складу ключових. Курсор набере вигляду .

3. Клацніть лівою кнопкою миші й, утримуючи її, перенесіть у списку

атрибутів сутності вище за горизонтальну лінію.

4. Клацніть правою кнопкою миші на сутності та виберіть у контекстному меню пункт Key Group. Ви увійдете в редактор Key Groups.

5. Клацніть кнопку New і за необхідності змініть ім'я створюваного

ключа. Клацніть кнопку OK.

6. Виберіть у списку Available Attributes необхідний атрибут (у цьому

випадку Модель) і клацніть кнопку . Таким чином, атрибут переміститься в список Key Group Members (рис. 5.12).

7. Клацніть кнопку OK.

Рис. 5.12. Приклад створення альтернативного ключа

5.3.6. Типи сутностей та ієрархія наслідування

Завдання

Створіть категоріальний зв'язок.

113

Виконання

Для виконання завдання необхідно дотримуватись такої послідовності дій.

1. Створіть сутність – родовий предок ("НАКЛАДНА").

2. Створіть сутності – нащадки ("ВИТРАТНА НАКЛАДНА ТА ПРИБУТКОВА НАКЛАДНА").

3. Клацніть кнопку на панелі інструментів.

4. Клацніть спочатку на родовому предку, а потім на нащадку.

5. Для встановлення другого зв'язка в ієрархії категорії слід спочатку клацнути на символі категорії, потім на другому нащадку (рис. 5.13).

Примітка. Для кожного предка можна вказати атрибути, властиві цьому типу

сутності. Наприклад, для прибуткової накладної необхідно вказати, від якого постачальника був прийнятий товар. Таким чином, діаграма набере вигляду, поданого на

рис. 5.13.

Рис. 5.13. Вид створеної діаграми

5.3.7. Рівні демонстрації зображення в ERwin

Завдання

Зробіть скриншот наступного рівня – демонстрація сутностей.

114

Виконання

Для виконання завдання необхідно вибрати пункт меню: Format ->

Display -> Entity (рис. 5.14).

Рис. 5.14. Рівень демонстрації визначень сутності

5.3.8. Правила посилальної цілісності в ERwin

Завдання

Створіть правило посилальної цілісності RESTRICT для зв'язку сутностей "ПОКУПЕЦЬ" і "ЗАМОВЛЕННЯ".

Виконання

Для виконання завдання необхідно дотримуватись такої послідовності дій.

1. Зайдіть у властивості зв'язка, клацнувши правою кнопкою миші на зв'язку та вибравши пункт Relationship Properties у контекстному меню.

2. Зайдіть на вкладку RI Actions.

3. Задайте правила посилальної цілісності (рис. 5.15).

115

Рис. 5.15. Вікно встановлення правил посилальної цілісності

5.3.9. Створення фізичної моделі даних в ERwin

Завдання

Створіть подання "ЗАМОВЛЕННЯ\_ПОКУПЦІ".

Виконання

Для виконання завдання необхідно дотримуватись такої послідовності дій.

1. Клацніть значок на панелі інструментів. Курсор набере вигляду .

2. Клацніть на будь-якому місці діаграми. На екрані з'явиться нова

сутність (рис. 5.16).

3. Клацніть правою кнопкою миші на поданні та виберіть у меню

пункт Database View Properties.

116

Рис. 5.16. Вікно створення подання "ЗАМОВЛЕННЯ\_ПОКУПЦІ"

5.3.10. Позбавлення від категоріальних зв'язків в ERwin

Завдання

Позбутися від категоріального зв'язку (атрибут Тип\_накладної).

Виконання

Для виконання завдання необхідно дотримуватись такої послідовності дій.

1. Виділіть створений категоріальний зв'язок для накладних і на панелі інструментів клацніть на кнопці "Supertype-Subtype Identity".

2. У вікні Supertype/Subtype Indentity Transform Wizard натисніть

кнопку "Далі" і перейдіть у вікно опису перетворення категоріального

зв'язку.

3. Введіть ім'я перетворення і його визначення. Клацніть кнопку

"Далі". Потрапляємо у вікно "Summary".

4. Клацніть кнопку "Готово".

У результаті отримуємо модель, показану рис. 5.17.

117

Рис. 5.17. Фрагмент моделі з перетвореним

категоріальним зв'язком

Надалі, позбавившись від усіх зв'язків "багато-до-багатьох", можна приступати до формування схеми бази даних у рамках конкретної СКБД.

5.3.11. Зворотне проектування в ERwin

Завдання

Виконайте зворотне проектування.

Виконання

Для виконання завдання необхідно дотримуватись такої послідовності дій.

1. Виберіть пункт меню Tools -> Reverse Engineer. Після цього виникає діалог Select Template, в якому треба вибрати шаблон діаграми,

потім ‒ діалог вибору СКБД.

2. Задайте опції зворотного проектування.

3. Виберіть базу даних і виконайте підключення (Connect).

У результаті отримана згенерована модель (рис. 5.18).

118

Рис. 5.18. Згенерована модель за зворотного

проектування

Завдання для самостійного виконання

1. Зробіть скриншоти таких рівнів: демонстрація атрибутів сутностей, демонстрація первинних ключів, демонстрація визначень сутностей,

демонстрація піктограм.

119

2. Створіть область для ілюстрації бізнес-функцій відділу продаж,

яка включатиме сутності, що належать до оформлення замовлення: Покупець, Замовлення, Товар, Служба доставки.

3. Створіть новий домен ЦІНА та новий атрибут ЦінаЗамовлення

в моделі, використовуючи опис доменів.

4. Створіть правило валідації для колонки Тип таблиці "СЛУЖБА

ДОСТАВКИ". У описі предметної області задано, що замовлення можуть

доставлятися або кур'єрами, або транспортними компаніями.

5. Виконайте пряме проектування логічної моделі даних.

6. Створіть звіт "СУТНІСТЬ" щодо лабораторної роботи за допомогою інструменту – Data Browser, в якому подайте логічну та фізичну схеми БД з CASE-інструменту, а також схему БД, отриману

в Access:

6.1. Відформатуйте звіт: змініть сортування даних, черговість колонок, зробіть колонку невидимою, задайте її стиль.

6.2. Виконайте експорт набору даних "СУТНІСТЬ" у форматі DDE.

6.3. Перевірте помилки у створеній моделі.

6.4. Створіть подання звіту.

7. Створіть звіт з лабораторної роботи за допомогою Report

Template Builder.

8. Побудуйте інформаційну модель для індивідуально обраної предметної області (див. табл. 4.36) – це базовий рівень лабораторної роботи.

9. Побудуйте інформаційну модель для індивідуально обраної предметної області (див. табл. 4.36) – це розширений рівень лабораторної

роботи.