

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

**МЕТОДИЧНІ ВКАЗІВКИ**

до виконання лабораторних робіт

**«Моделювання даних з використанням нотації IDEF1X»**

для студентів спеціальностей

121 «Інженерія програмного забезпечення»,

122 «Комп'ютерні науки»,

126 «Інформаційні системи та технології»

Затверджено  
редакційно-видавничою  
радою університету,  
протокол № 3 від 26.10.2022 р.

Харків  
НТУ «ХПІ»  
2022

Методичні вказівки до виконання лабораторних робіт за темою «Моделювання даних з використанням нотації IDEF1X» для студентів спеціальностей 121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки» та 126 «Інформаційні системи та технології» / уклад. Д.Л. Орловський, А.М. Копп. – Харків : НТУ «ХП», 2022. – 33 с.

Укладачі: Д.Л. Орловський

А.М. Копп

Рецензент: В.В. Москаленко

Кафедра програмної інженерії та інтелектуальних технологій управління

## ЗМІСТ

Вступ .....	4
Розробка моделей IDEF1X .....	7
Створення бази даних на основі моделі IDEF1X .....	27
Список джерел інформації .....	33

## ВСТУП

Мета: Постановка завдання лабораторного практикуму. Вивчення і аналіз предметної області. Виділення на основі аналізу основних об'єктів бази даних.

### **Загальний (спрощений) опис предметної області**

Деяке підприємство купує продукцію у різних постачальників. Постачальниками можуть бути як юридичні особи, так і фізичні особи – суб'єкти підприємницької діяльності. Придбання продукції здійснюється партіями і оформляється у вигляді договорів на поставку. Кожен договір на поставку продукції має унікальний номер і може бути укладений тільки з одним постачальником. Підставою для поставки є який-небудь документ (попереднє замовлення, рахунок-фактура і т.п.). У документах за кожним договором для кожного виду продукції зазначаються: найменування, розмір поставленої партії і ціна (в грн.). Поставлена продукція оприбутковується на склад з метою подальшої реалізації шляхом продажу, відпустки на реалізацію і т.п.

Аналіз предметної області дозволив виділити і деталізувати основні бізнес-процеси, пов'язані з поставками продукції. Передбачається, що в процесі роботи підприємства доведеться зберігати досить великі обсяги даних, пов'язані з поставками продукції. Крім того, інформація про поставки продукції повинна бути організована таким чином, щоб персонал і керівництво підприємства мали можливість здійснювати її аналітичну обробку. У зв'язку з цим для зберігання і обробки інформації, пов'язаної з поставками продукції, необхідно створити базу даних. Аналіз бізнес-процесів дозволив виділити наступні інформаційні масиви, які можуть входити до складу такої бази даних.

#### **1. Відомості про продукцію**

Включають в себе інформацію про назву продукції, ідентифікатор, який однозначно визначає кожну конкретну продукцію (код, артикул і т.п.), одиницю виміру продукції (штука, ящик, кг тощо) і т.д. Продукція об'єднана в деякі групи (товарні групи) – наприклад, продукти харчування, парфумерія, побутова хімія і т.д. Передбачається, що кожна продукція може входити тільки в одну товарну групу.

#### **2. Відомості про постачальників продукції**

Включають в себе інформацію про суб'єктів підприємницької діяльності, які працюють на ринку і пропонують продукцію, в придбанні якої зацікавлене розглядається підприємства. Постачальниками можуть бути як юридичні особи, так і фізичні особи.

До відомостей про постачальників відносяться такі дані, як назва суб'єкта підприємницької діяльності, індивідуальний податковий номер, номер свідоцтва платника ПДВ (для юридичної особи); прізвище, ім'я, по батькові, номер свідоцтва про реєстрацію (для фізичної особи); адреса місцезнаходження, контактний телефон (для фізичної та юридичної особи) і т.д.

3. Відомості про ціни на продукцію, пропоновану різними постачальниками

Одна і та сама продукція може пропонуватися на ринку різними постачальниками, причому кожен постачальник може пропонувати продукцію за різними цінами (роздрібною, дрібнооптовою, оптовою і т.д.) в залежності від обсягу закупівлі, умов договору і т.д.

4. Відомості про постачання продукції

Кожна поставка здійснюється на підставі договору на поставку, який укладається між постачальником і підприємством. Для кожної поставки відома наступна інформація: постачальник, дата поставки, загальна сума поставки, дані про поставлену продукцію. Дані про поставлену продукцію включають в себе (по кожній продукції): найменування продукції, кількість одиниць, ціну за одиницю. Ціна, за якою постачається продукція, може відрізнятися від стандартного прайс-листа постачальника (для конкретного покупця можуть діяти спеціальні знижки, ціна на окремі види продукції може призначатися індивідуально і т.д.).

На основі аналізу перерахованих інформаційних масивів можна виділити наступні таблиці (сутності), які будуть входити до складу бази даних, що розробляється. Для кожної таблиці (сутності) також наводяться поля (атрибути), що конкретизують збережену в таблицях інформацію:

1. Товарні групи
  - 1.1. Код товарної групи
  - 1.2. Назва товарної групи
2. Одиниці виміру продукції
  - 2.1. Код одиниці виміру
  - 2.2. Назва одиниці виміру
3. Продукція
  - 3.1. Код продукції
  - 3.2. Назва продукції
  - 3.3. Товарна група
  - 3.4. Одиниця виміру
4. Види цін на продукцію
  - 4.1. Код виду ціни
  - 4.2. Назва виду ціни

## 5. Постачальники

- 5.1. Код постачальника
- 5.2. Назва постачальника (для юридичної особи)
- 5.3. Індивідуальний податковий номер (для юридичної особи)
- 5.4. Номер свідоцтва платника ПДВ (для юридичної особи)
- 5.5. Прізвище, ім'я, по батькові (для фізичної особи)
- 5.6. Номер свідоцтва про реєстрацію (для фізичної особи)
- 5.7. Адреса
- 5.8. Контактний телефон
- 6. Ціни на ринку
  - 6.1. Продукція
  - 6.2. Постачальник
  - 6.3. Вид ціни
  - 6.4. Значення ціни
  - 6.5. Умова надання ціни
- 7. Договори на постачання продукції
  - 7.1. Код договору
  - 7.2. Дата постачання
  - 7.3. Постачальник
  - 7.4. Коментар (деяка додаткова інформація про поставку)
- 8. Поставлена продукція
  - 8.1. Код договору
  - 8.2. Продукція
  - 8.3. Поставлена кількість
  - 8.4. Ціна за одиницю

Наведена вище інформація може бути використана як вихідна при розробці моделі даних IDEF1X.

## РОЗРОБКА МОДЕЛЕЙ IDEF1X

Мета: розробити моделі IDEF1X.

Кроки:

1. Створення логічної моделі:

- 1) запустити ERWin Data Modeler і створити нову модель за допомогою New (File -> New);
- 2) у вікні «Create Model – Select Template» вибрати «Logical/Physical»;
- 3) у розділі «Target Database» виберіть «Database» – Access і «Version» – 2000/2002/2003. Натисніть «OK»;
- 4) у результаті буде показано інтерфейс ERWin (рисунок 1).

2. Створення сутностей та атрибутів.

Необхідну послідовність кроків розглянемо на прикладі сутності «Product groups »:

- 1) натисніть «Entity»;
- 2) перемістіть вказівник на місце, де можна розмістити сутність, і натисніть ліву кнопку миші (рисунок 2);
- 3) змініть ім'я сутності, для якої клацніть по ній правою кнопкою миші та виберіть «Entity Properties...», потім у вікні «Entities», що з'явилося, введіть нове ім'я «Product\_groups» і натисніть «Ok» (рисунок 3);
- 4) створіть атрибути сутності, для чого клацніть на ній правою кнопкою миші та виберіть «Attributes...», у вікні «Attributes» (рисунок 4) натисніть кнопку «New» і у вікні «New Attribute», що з'явилося, введіть «product\_group\_ID», виберіть домен «Number» (рисунок 5) і натисніть «Ok». Створіть другий атрибут «product\_group\_name» типу «String»;
- 5) встановіть первинний ключ для цієї сутності, для якого виберіть атрибут «product\_group\_ID» і натисніть «Primary Key» (рисунок 6);
- 6) натисніть «Ok» у вікні «Attributes» (рисунок 7), і тепер буде створено сутність «Product\_groups».

Решта об'єктів, які можуть бути створені на цьому кроці, мають бути створені таким же чином. Опис цих сутностей і атрибутів наведено в таблиці 1.

Таблиця 1

Сутність	Атрибут	Ключ	Домен
Measurement_units	measurement_unit_ID	PK	Number
	measurement_unit_name		String
Products	product_ID	PK	Number
	product_name		String
Suppliers	supplier_ID	PK	Number
	address		String
	phone		String
Price_types	price_type_ID	PK	Number
	price_type_name		String
Contracts	contract_ID	PK	Number
	supply_date		Datetime
	comment		String
Supplied_products	amount		Number
	price_per_item		Number
Legal_entities	name		String
	tax_number		String
	VAT_cert_number		String
Ind_entrepreneurs	last_name		String
	first_name		String
	second_name		String
	reg_cert_number		String

У результаті логічна модель, яка містить створені сутності та атрибути, має виглядати так, як показано на рисунку 8.

3. Збережіть створену модель, для чого натисніть «Save Model» і введіть назву файлу «delivery».

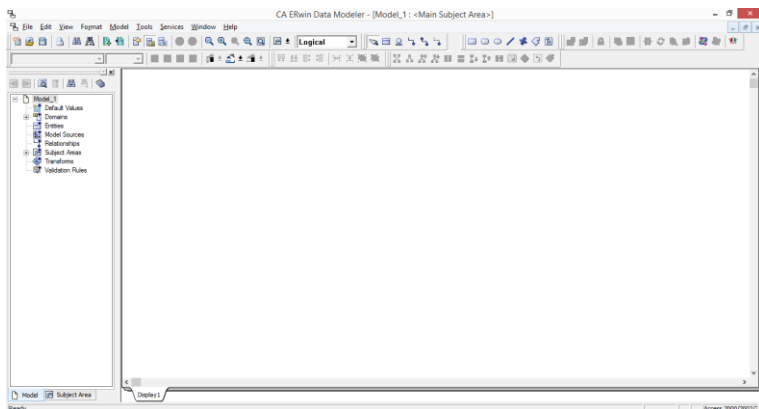


Рисунок 1



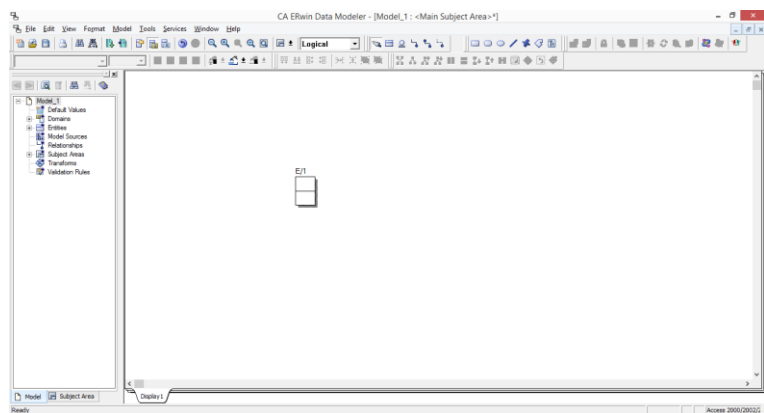


Рисунок 2

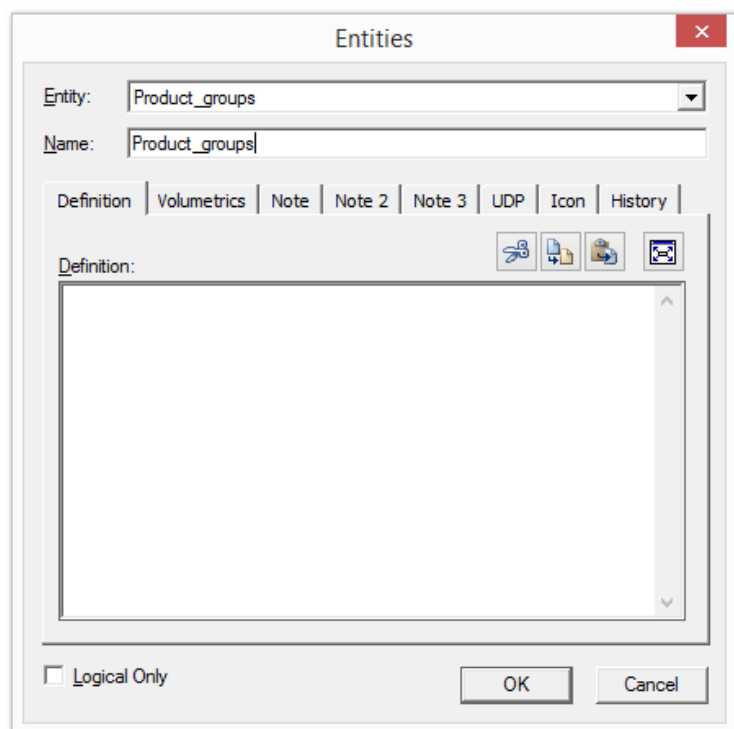


Рисунок 3

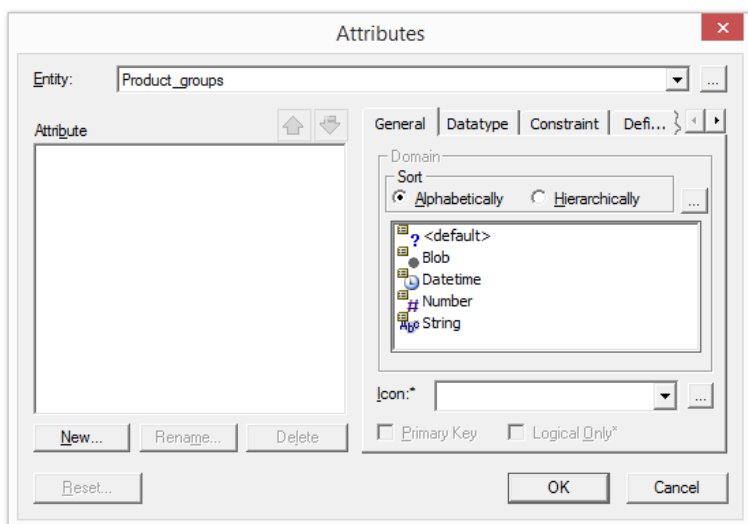


Рисунок 4

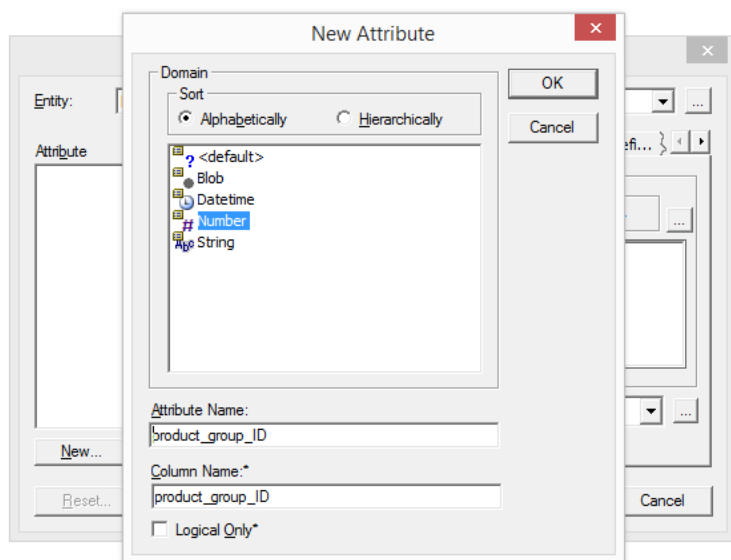


Рисунок 5

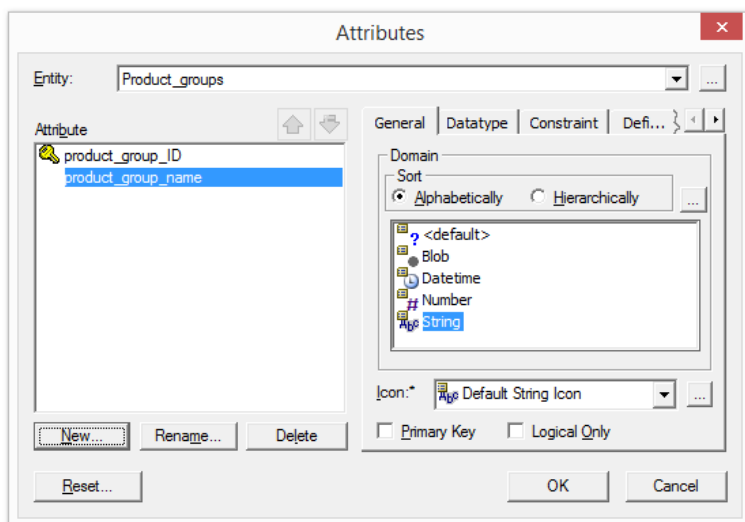


Рисунок 6

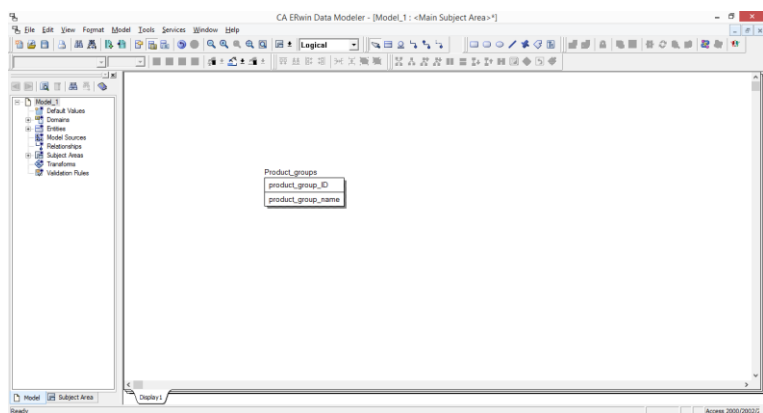


Рисунок 7

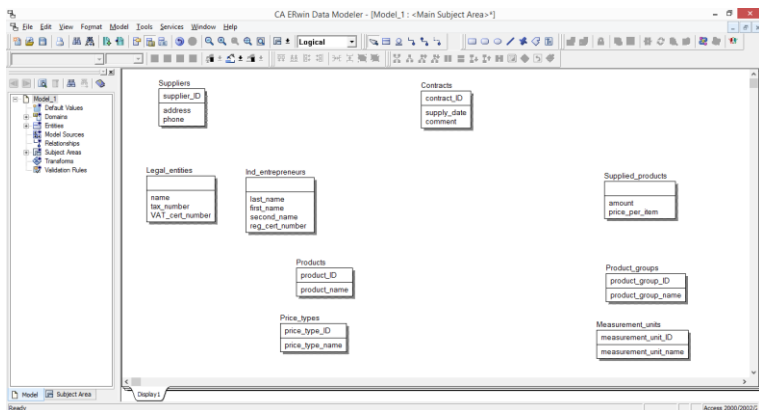


Рисунок 8

#### 4. Створення зв'язків

Для опису послідовності дій, необхідних для створення зв'язків між сутностями, розглянемо приклад створення неідентифікуючого зв'язку між сутностями «Products» та «Product\_groups»:

1) натисніть «Non-identifying relationship»;

2) натисніть на сутність «Product\_groups», а потім на «Products», щоб створити неідентифікуючий зв'язок між цими сутностями (рисунок 9). Первинний ключ (PK) сутності «Product\_groups» тепер є зовнішнім ключем (FK) сутності «Products». Оскільки FK не є частиною PK, створений зв'язок не є ідентифікуючим.

#### Примітка.

Якщо зв'язок на схемі показаний не так, як має бути, натисніть на зв'язок і перемістіть його в потрібне місце.

Зв'язки між створеними сутностями можуть бути створені таким же чином. Перелік зв'язків між сутностями та їх типи наведені в таблиці 2.

Таблиця 2

Батьківська сутність	Дочірня сутність	Тип зв'язку
Product_groups	Products	Non-identifying
Measurement_units	Products	Non-identifying
Suppliers	Contracts	Non-identifying
Products	Supplied_products	Identifying
Contracts	Supplied_products	Identifying

В результаті діаграма буде виглядати так, як показано на рисунку 10.

5. Створіть категоріальні зв'язки для дочірніх сутностей в ієрархії успадкування

Ієрархія успадкування (або категоріальна ієрархія) – це особливий тип зв'язків сутностей, які мають спільні атрибути. У цьому прикладі це сутність «Suppliers». Оскільки постачальниками можуть бути як юридичні особи, так і фізичні особи-підприємці, то зберігати дані про ці типи в одній особі незручно. Таким чином, окрім сутності «Suppliers», також були створені «Legal\_entities» та «Ind\_entrepreneurs».

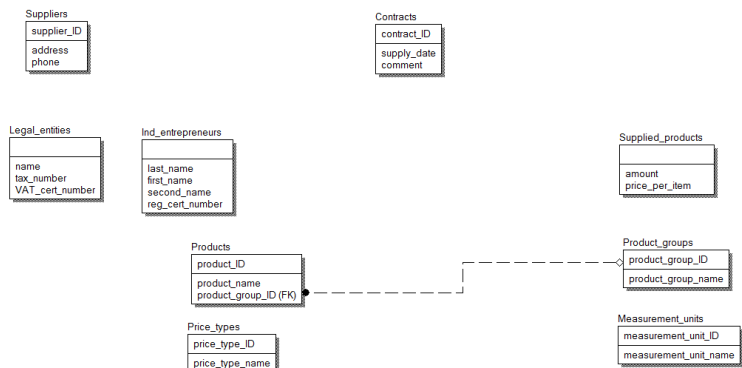


Рисунок 9

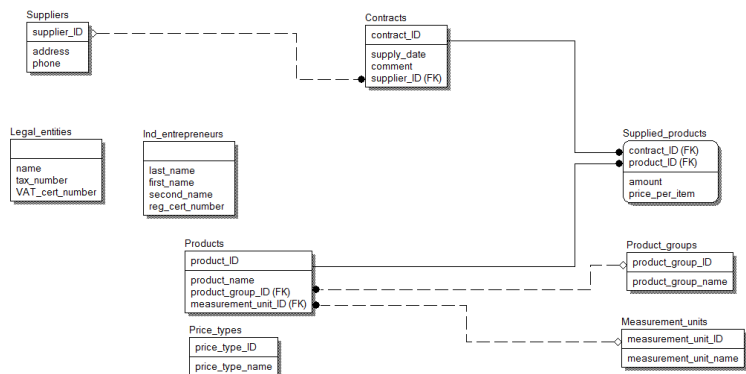


Рисунок 10

Кроки, необхідні для створення ієрархії успадкування, яка включатиме сутності «Suppliers», «Legal\_entities» та «Ind\_entrepreneurs»:

- 1) натисніть «Complete sub-category»;

2) натисніть на сутність «Suppliers», а потім на «Legal\_entities», щоб створити категоріальний зв'язок (рисунок 11). Таким чином, РК сутності «Suppliers» переміститься і також стане РК сутності «Legal\_entities»;

3) ще раз натисніть «Complete sub-category»;

4) клацніть на категоріальний зв'язок і на сутність «Ind\_entrepreneurs», щоб створити категоріальний зв'язок (рисунок 12). Таким чином, РК сутності «Suppliers» також стане РК сутності «Ind\_entrepreneurs».

## 6. Створення зв'язку «багато-до-багатьох» між сутностями

Відношення «багато-до-багатьох» буде використано для створення сутності, яка забезпечує зберігання інформації про стан ринку (цінові пропозиції постачальників). Створення зв'язку «багато-до-багатьох» між сутностями «Products», «Suppliers» та «Price\_types» включає такі кроки:

1) натисніть «Many-to-many relationship»;

2) спочатку натисніть на «Suppliers», а потім на «Products», щоб створити зв'язок (рисунок 13);

3) клацніть правою кнопкою миші на створеному зв'язку, виберіть «Create Association Entity» і натисніть «Next»;

4) введіть «Market\_prices» у поле «Entity Name», щоб створити нову сутність «Market\_prices», яка забезпечує зв'язок «багато-до-багатьох» між сутностями «Suppliers» та «Products» (рисунок 14);

5) існування зв'язку «багато-до-багатьох» показує, що кожен постачальник може пропонувати різні продукти на ринку, і навпаки, кожен продукт може пропонуватися різними постачальниками. Оскільки цей зв'язок не надає інформації про ціну, необхідно клацнути правою кнопкою миші на сутності «Market\_prices», вибрати «Attributes» і створити ще один атрибут «price» типу «Number».

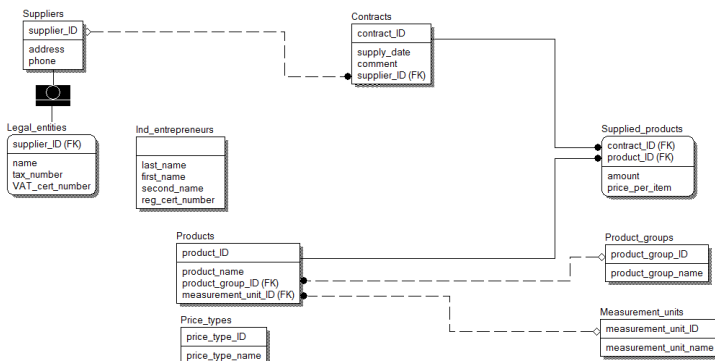


Рисунок 11

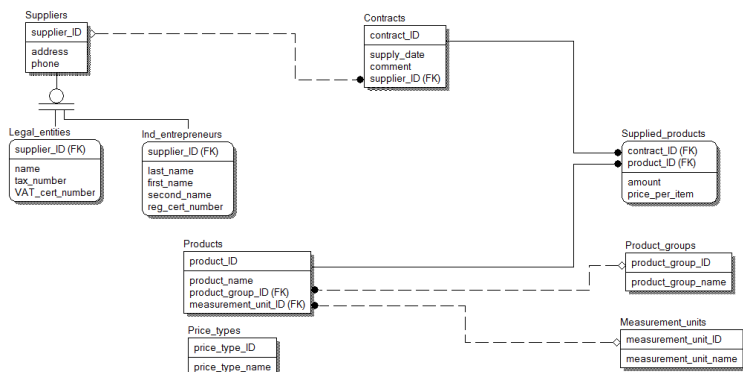


Рисунок 12

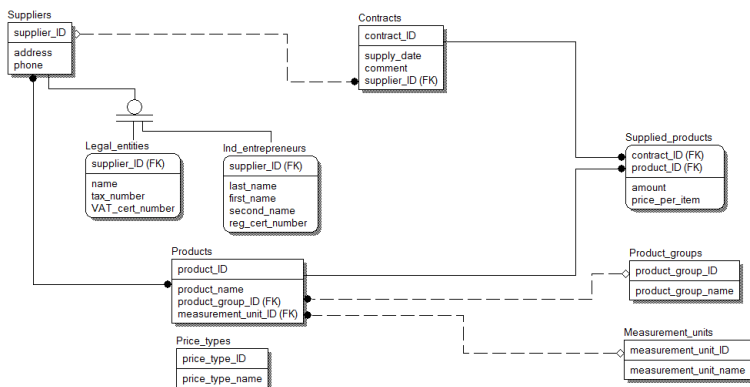


Рисунок 13

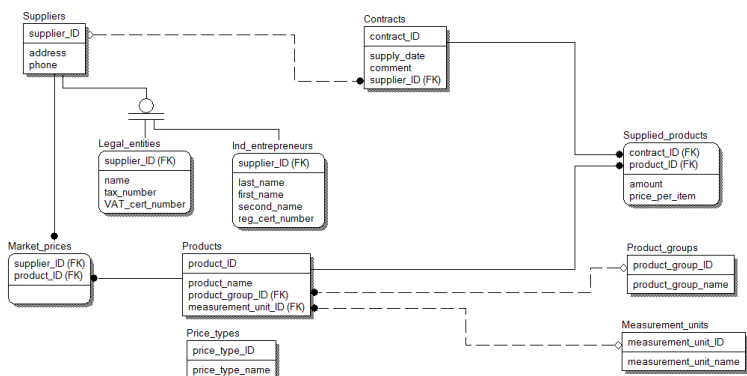


Рисунок 14

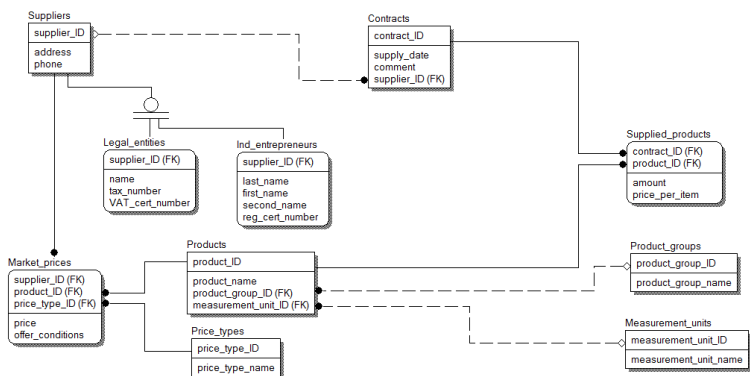


Рисунок 15

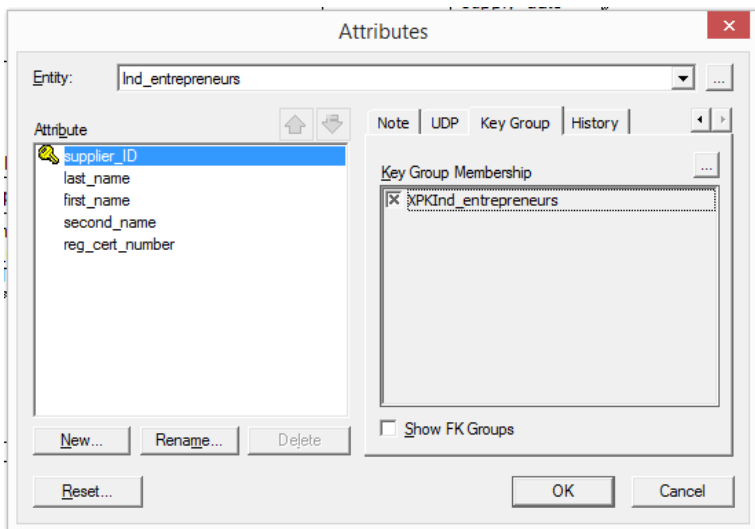


Рисунок 16



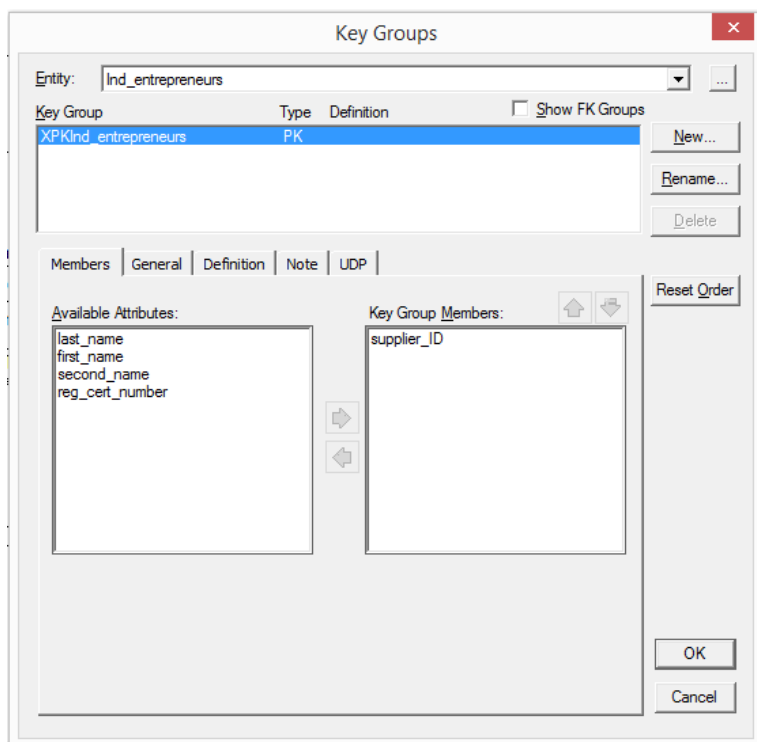


Рисунок 17

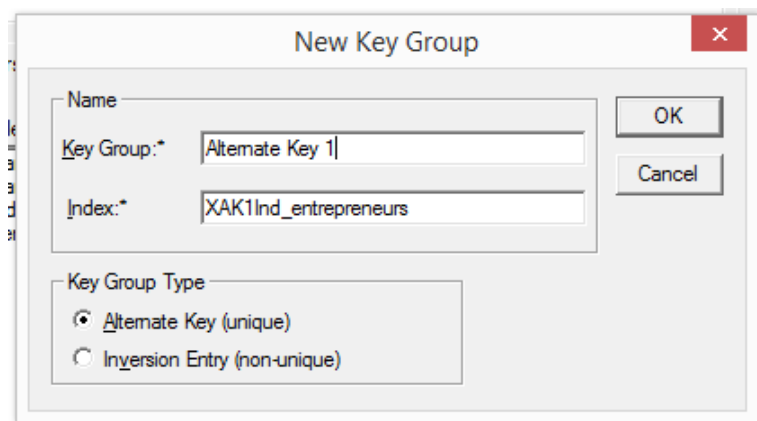


Рисунок 18

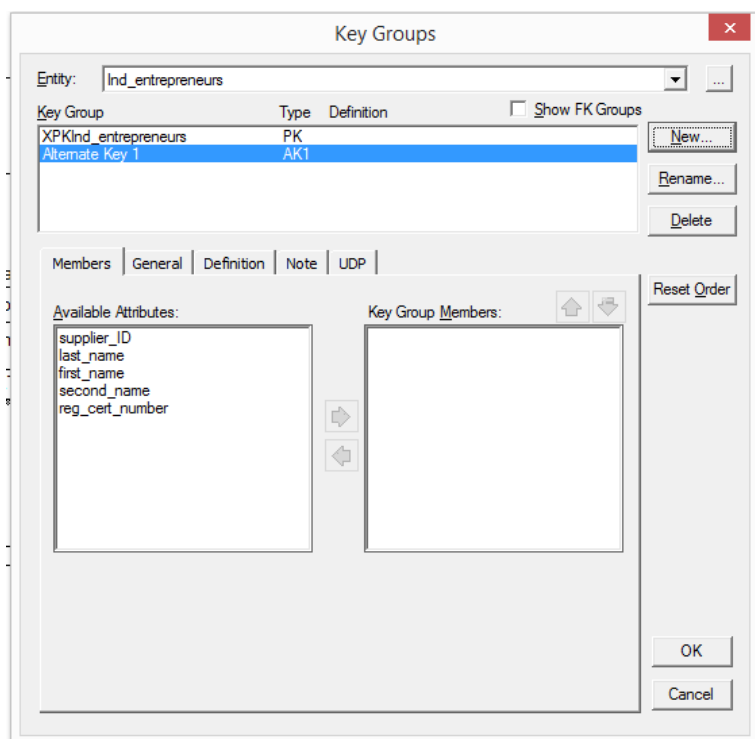


Рисунок 19

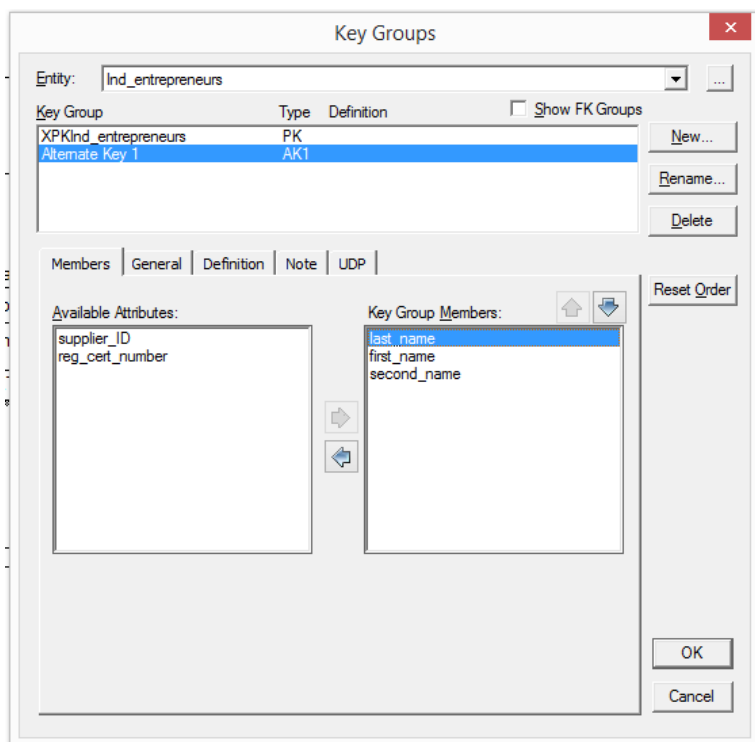


Рисунок 20

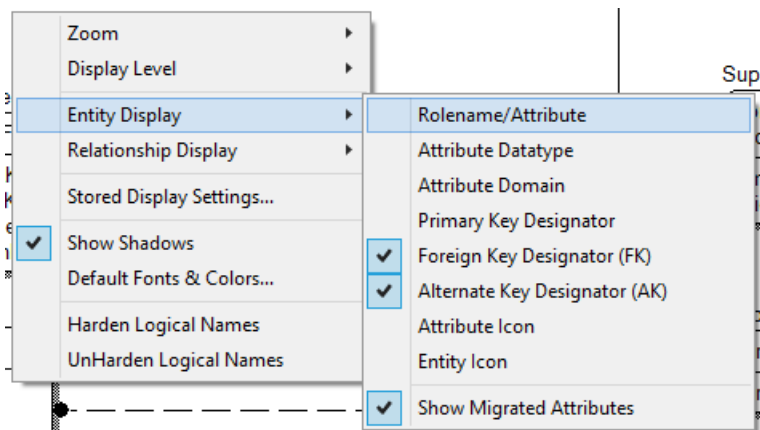


Рисунок 21

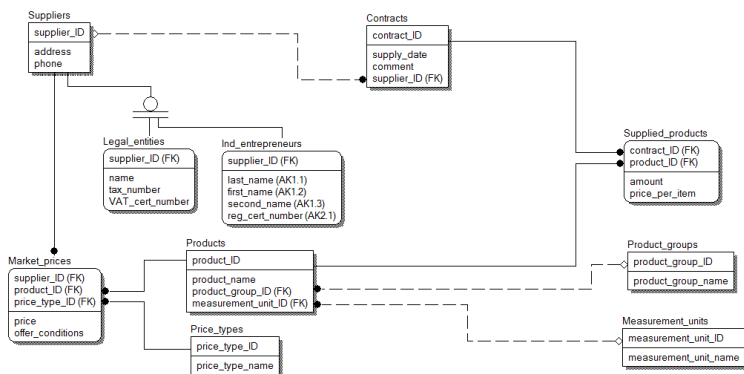


Рисунок 22

6) проте кожен постачальник може запропонувати певний продукт з єдиною ціною. Щоб забезпечити збереження інформації про різні ціни на продукцію, пропоновані одним постачальником, необхідно натиснути «Identifying relationship», а потім клацнути сутність «Price\_types», а потім «Market\_prices», щоб створити зв'язок. РК «Price\_types» увійде до РК «Market\_prices» (рисунок 15).

7) для надання клієнтам інформації про умови пропозиції, необхідно додати ще один атрибут «offer\_conditions» типу «String» (рисунок 15).

## 7. Створення альтернативних ключів

Зазвичай під час моделювання даних потрібно визначити ще один або кілька альтернативних ключів, крім РК. Як зазвичай, альтернативні ключі – це потенційні ключі, які не були обрані як основні. Наявність альтернативних ключів дозволяє контролювати унікальні атрибути, які не належать до РК. Припустимо, фізичні особи-підприємці зобов'язані зберігати унікальні прізвище, ім'я та по батькові, а також унікальний номер свідоцтва про реєстрацію. Для цього потрібно створити два альтернативних ключа. Послідовність необхідних кроків така:

1) клацніть правою кнопкою миші на «Ind\_entrepreneurs» і виберіть «Attributes...»;

2) відкрийте вкладку «Key Group» (рисунок 16) і натисніть «...». Буде показано вікно «Key Groups» з даними ключів цієї сутності (рисунок 17);

3) натисніть «New», щоб створити нові альтернативні ключі за допомогою вікна «New Key Group» (рисунок 18), потім натисніть «Ok» (рисунок 19);

4) щоб створити список ключових атрибутів, перемістіть «last\_name», «first\_name» та «second\_name» з «Available Attributes» у «Key Group Members» (рисунок 20);

5) тепер створіть другий альтернативний ключ таким же чином, включивши лише «reg\_cert\_number» до членів групи ключів;

6) після закриття вікна «Key Groups» вкладка «Key Group» міститиме змінений список поточних ключів сутності;

7) закрийте вікно «Attributes», натиснувши «Ok»;

8) щоб відобразити інформацію про альтернативні ключі, клацніть правою кнопкою миші на будь-якому порожньому місці моделі, виберіть «Entity Display», «Alternate Key Designator (AK)» (рисунок 21). У результаті будуть показані альтернативні ключі серед атрибутів сутності (рисунок 22).

#### 8. Зміна властивостей зв'язку

Модифікація властивостей раніше створених зв'язків є звичайною ситуацією процесу моделювання даних. Це залежить від типових властивостей зв'язків, які не відповідають вимогам розробленої бази даних. Давайте розглянемо приклад модифікації властивостей, використовуючи зв'язок між «Products» і «Product\_groups». Цей зв'язок показує, що кожен товар належить до певної товарної групи. Однак деякі продукти можуть не належати до жодної групи продуктів. Це можливо, оскільки FK «Product\_group\_ID» в сутності «Products» може бути порожнім (null). Ця властивість встановлена за замовчуванням, але вона не відповідає вимогам, заявленим у результаті аналізу предметної області, тому її слід змінити. Існує послідовність кроків, які використовуються для зміни властивостей зв'язку:

1) двічі клацніть на зв'язку, щоб відкрити вікно «Relationships», де відображаються властивості зв'язків (рисунок 23);

2) виберіть «No Nulls» у розділі «Nulls»;

3) натисніть «Ok», щоб закрити вікно «Relationships»;

4) проаналізуйте, як змінилося представлення зв'язків на діаграмі;

5) змініть ті ж властивості інших неідентифікуючих зв'язків;

6) збережіть всі зміни, зроблені в моделі.

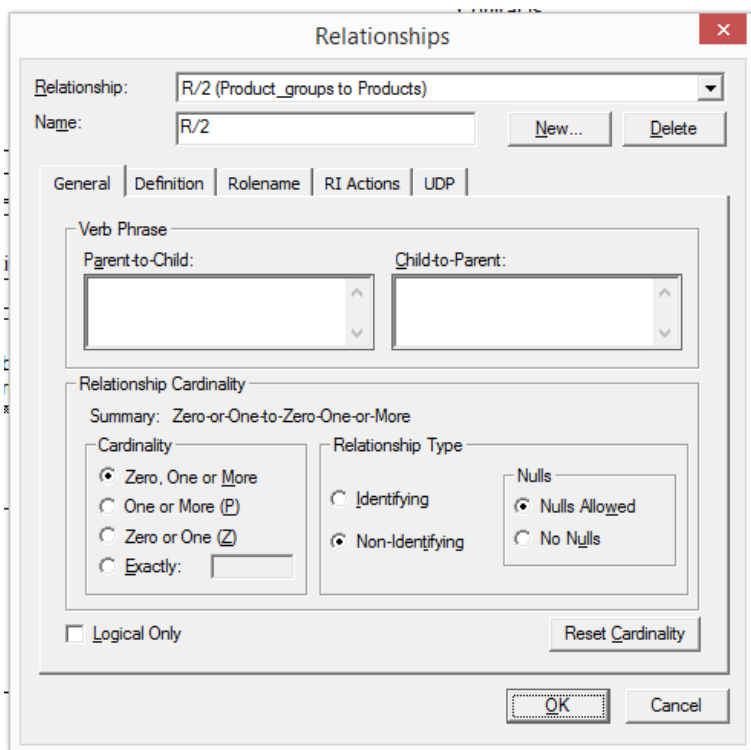


Рисунок 23

## 9. Зміна категоріальних зв'язків в ієрархії успадкування

Категоріальні відносини також можуть вимагати модифікації. Є кілька прикладів, пов'язаних із модифікацією властивостей категоріальних зв'язків або їх перетворенням.

### **Увага! Наступні зміни не потрібно зберігати!**

Приклад 1. Зміна типу категоріальної ієрархії.

Існує два типи категоріальних ієрархій – повна і неповна. Категоріальний зв'язок, реалізований раніше, є прикладом повної категорії. Щоб змінити тип категорії, необхідно виконати такі кроки:

- 1) клацніть правою кнопкою миші на категоріальному зв'язку та виберіть «Subtype Relationship...», який демонструє властивості зв'язку;
- 2) переведіть «Subtype Type» в режим «Incomplete»;
- 3) натисніть «Ok» і перевірте, як зміниться діаграма після зміни типу категорії на незавершену;
- 4) відновіть повний тип категорії.

Приклад 2. Замінити ієрархію успадкування ідентифікаційними зв'язками.

Ця заміна призначена для спрощення моделі або її вдосконалення на основі суб'єктивних дизайнерських рішень. Щоб замінити ієрархію успадкування ідентифікаційними зв'язками, необхідно виконати наступні дії:

1) клацніть категоріальний зв'язок, а потім натисніть «Supertype-Subtype Identity»;

2) у відкритому майстрі двічі натисніть «Next», а потім «Finish». Результат перетворення наведено на рисунку 24. Як показано, категоріальне відношення було замінено двома ідентифікаційними відношеннями;

3) закрити модель без збереження змін.

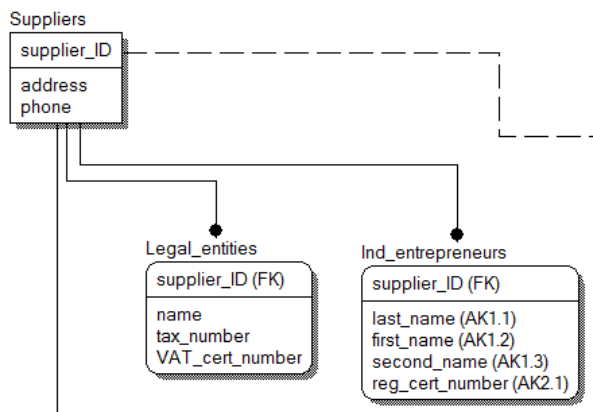


Рисунок 24

Приклад 3. Перенесіть РК і неключові атрибути в ієрархії успадкування від дочірнього до батьківського.

Ця заміна також може бути спрямована на спрощення або покращення моделі. Це завдання вимагає наступних кроків:

1) відкрийте модель;

2) клацніть на категоріальний зв'язок, а потім на «Supertype-Subtype Rollup»;

3) пропустіть всі кроки у вікні майстра. Результат показано на рисунку 25. Як показано, залишилася лише одна сутність «Suppliers». Перелік атрибутів цієї сутності містить атрибути сутностей «Legal\_entities» та «Ind\_entrepreneurs». Ця структура має очевидні недоліки, пов'язані з вимогами нормалізації;

4) закрийте модель без збереження змін.

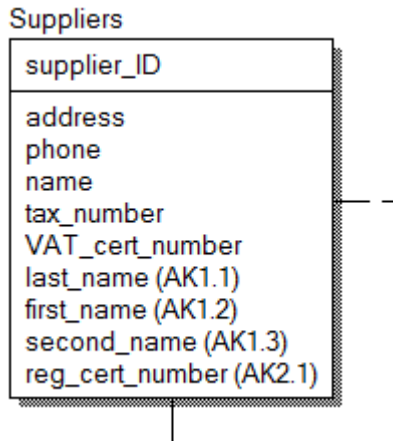


Рисунок 25

Приклад 4. Перенесіть РК і неключові атрибути в ієрархії успадкування від батьківського до дочірнього.

Ця заміна також має недоліки, пов'язані з вимогами нормалізації. Це вимагає наступних кроків:

- 1) відкрийте модель;
- 2) клацніть на категоріальний зв'язок, а потім на «Supertype-Subtype Rolldown»;
- 3) пропустіть всі кроки у вікні майстра. Результат показано на рисунку 26. Як показано, залишилося лише дві сутності «Legal\_entities» та «Ind\_entrepreneurs». Список атрибутів цих сутностей включає атрибути сутності «Suppliers»;
- 4) закрийте модель без збереження змін;
- 5) знову відкрийте модель.



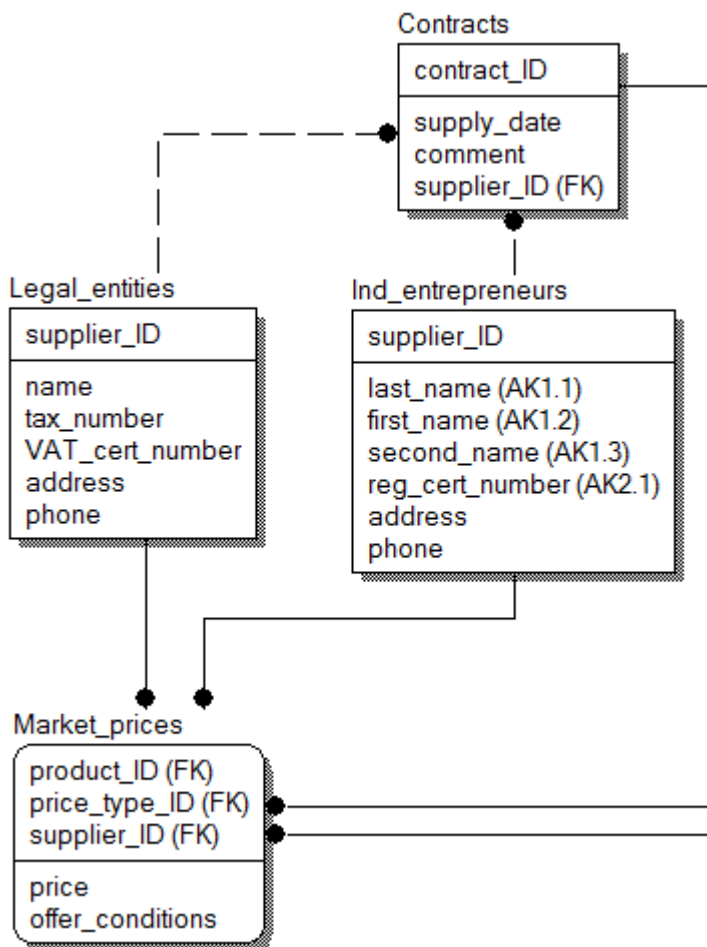


Рисунок 26

#### 10. Перехід до фізичної моделі даних

Логічна модель даних може бути створена без визначення СУБД. Однак справжню базу даних можна створити з урахуванням особливостей конкретної СУБД. У цій роботі СУБД була визначена при створенні моделі. Фізична модель даних, яка враховує особливості обраної СУБД, створюється автоматично під час проектування логічної моделі. Для доступу до фізичної моделі даних режим потрібно змінити з «Logical» на «Physical». У результаті буде показано фізичну модель (рисунок 27).

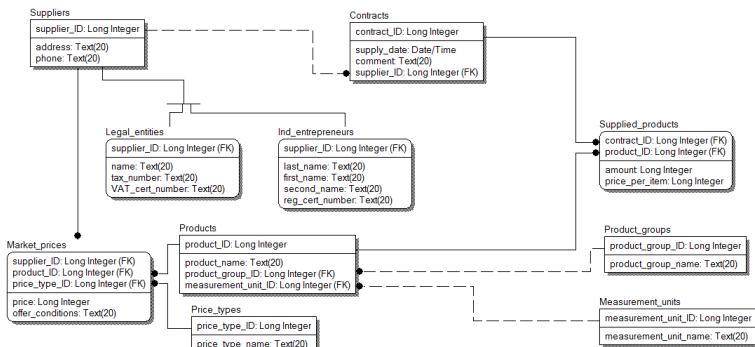


Рисунок 27

11. Збережіть створену модель і завершіть роботу.

12. Вимоги до звіту

- 1) коротко охарактеризуйте основні етапи виконаної роботи;
- 2) продемонструйте розроблені в ході роботи моделі (кінцевий вигляд моделей);
- 3) опишіть призначення основних елементів інтерфейсу ERWin, їх особливості, які використовуються для розробки моделі даних.

## СТВОРЕННЯ БАЗИ ДАНИХ НА ОСНОВІ МОДЕЛІ IDEF1X

Мета: створити базу даних на основі моделі IDEF1X.

Кроки:

1. Відкрийте модель, створену в результаті першої частини цієї роботи, збережену у файлі «delivery.erwin».

2. Виберіть фізичну модель. Тепер замість «entity» ми будемо використовувати визначення «table», а замість «attribute» ми будемо використовувати «field».

3. Зміна параметрів полів таблиці

Як приклад розглянемо зміну параметрів поля «price\_per\_item» в таблиці «Supplied\_products»:

1) клацніть правою кнопкою миші на таблиці «Supplied\_products» і виберіть «Columns...»;

2) виберіть поле «price\_per\_item» і виберіть тип даних «Currency» на вкладці «Access» (рисунок 28);

3) натисніть «Ok».

Таким же чином змініть тип даних поля «price» таблиці «Market\_prices» на «Currency».

4. Перевірка цільової СУБД

Виберіть меню «Database», потім виберіть «Choose Database». Вікно, що з'явилося, можна використати для перевірки та зміни налаштувань СУБД, яка використовується для створення бази даних на основі розробленої моделі (рисунок 29).

5. Створення бази даних за допомогою СУБД Access

Перш ніж створити базу даних на основі моделі даних, нам потрібно створити порожню базу даних без об'єктів. Послідовність кроків така:

1) створіть папку, де будуть розміщені файли бази даних (наприклад, D:\ER\_LAB);

2) запустіть СУБД Access;

3) створіть нову базу даних, введіть її назву «delivery.mdb» і виберіть місце розташування (рисунок 30);

4) у результаті буде створено нову базу даних без об'єктів (рисунок 31);

5) закрийте вікно бази даних і закрийте СУБД Access.

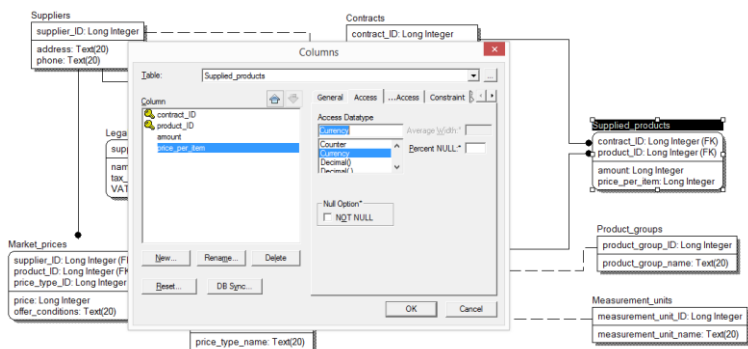


Рисунок 28

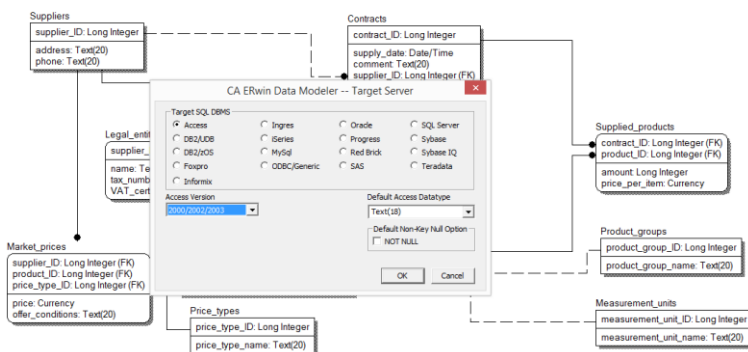


Рисунок 29

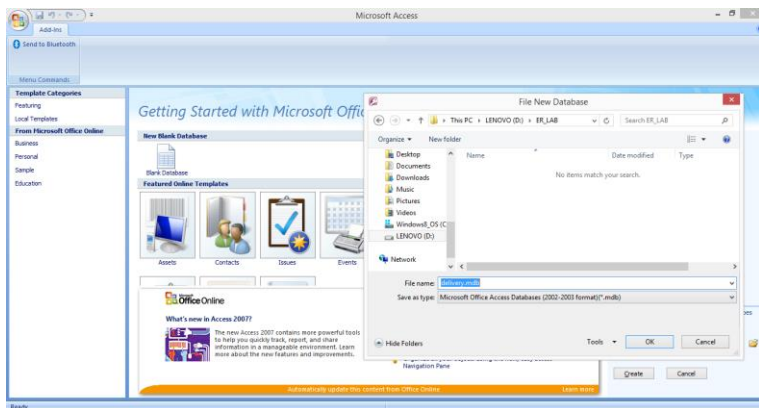


Рисунок 30

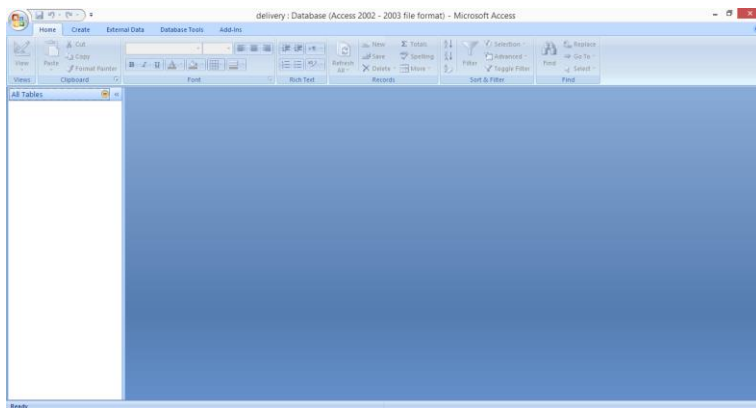


Рисунок 31

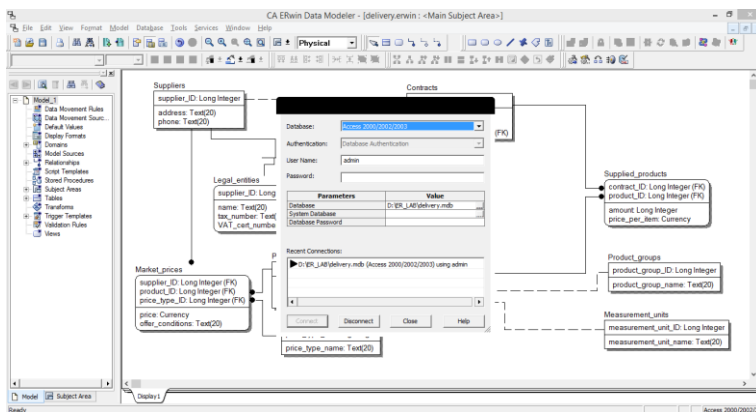


Рисунок 32

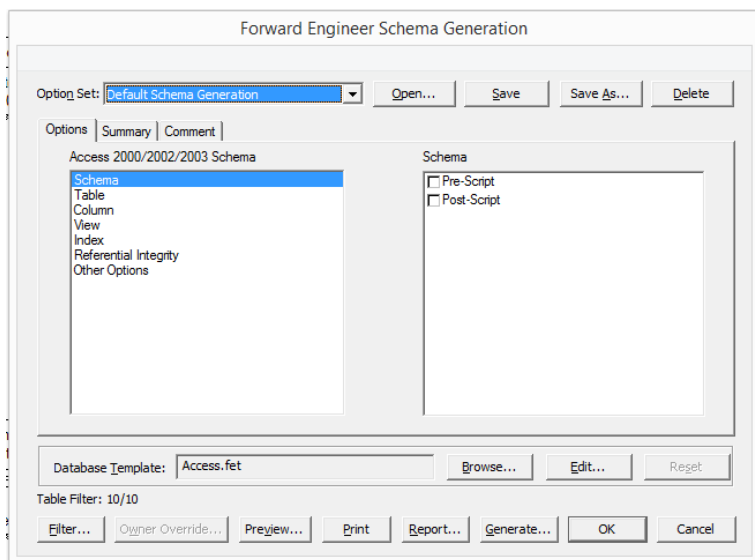


Рисунок 33

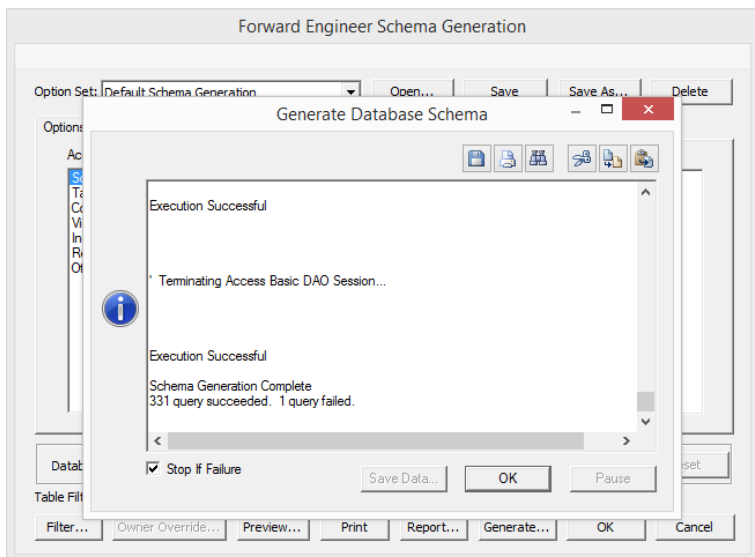


Рисунок 34

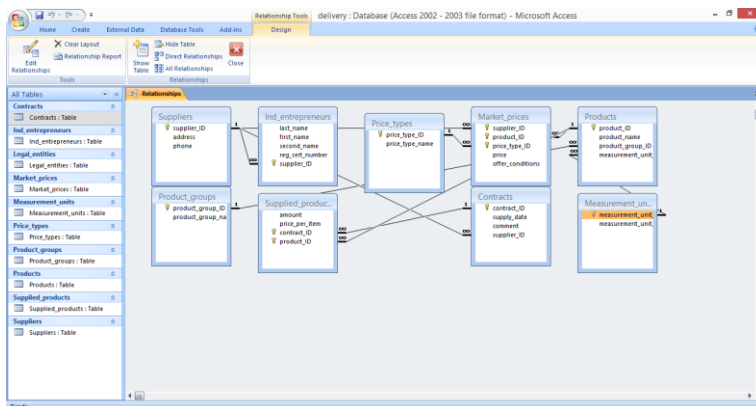


Рисунок 35

6. Підключення моделі до створеної бази даних.

Послідовність кроків така:

- 1) відкрийте ERWin;
- 2) виберіть «Database» у меню, потім виберіть «Database Connection»;
- 3) у вікні Access Connection, що з'явилася, в полі «User Name» введіть «admin», потім у полі «Database» (рисунок 32) введіть «D:\ER\_LAB\delivery.mdb»;
- 4) натисніть «Connect».

7. Створення бази даних на основі розробленої моделі.

Послідовність кроків така:

- 1) виберіть «Tools» в меню, потім виберіть «Forward Engineer/Schema Generation...»;
- 2) в результаті з'явиться вікно «Access Schema Generation» (рисунок 33);
- 3) натисніть «Generate»;
- 4) після завершення процесу натисніть «Ok» у вікні «Generate Database Schema» (рисунок 34), а потім натисніть «Ok» у вікні «Access Schema Generation».

8. Перевірка результату створення бази даних.

Послідовність кроків така:

- 1) запустити СУБД Access;
- 2) відкрити базу даних «delivery.mdb» і перевірте наявність таблиць та їх відповідність таблицям, створеним під час моделювання;
- 3) відкрити схему даних і перевірити наявність зв'язків між таблицями бази даних (рисунок 35);

4) перевірте альтернативні ключі таблиці «Ind\_entrepreneurs». Це вимагає наступних кроків:

- клацніть правою кнопкою миші на таблиці «Ind\_entrepreneurs» і виберіть «Design View»;

- у головному меню виберіть «Design», а потім виберіть «Indexes». Вікно, що з'явилося, показує список індексів цієї таблиці (рисунок 36). Як показано, є два альтернативних ключа, незважаючи на РК. Ці альтернативні ключі також унікальні.

- закрити вікно індексів таблиці;

- закрити вікно структури бази даних;

5) закрийте вікно БД і СУБД.

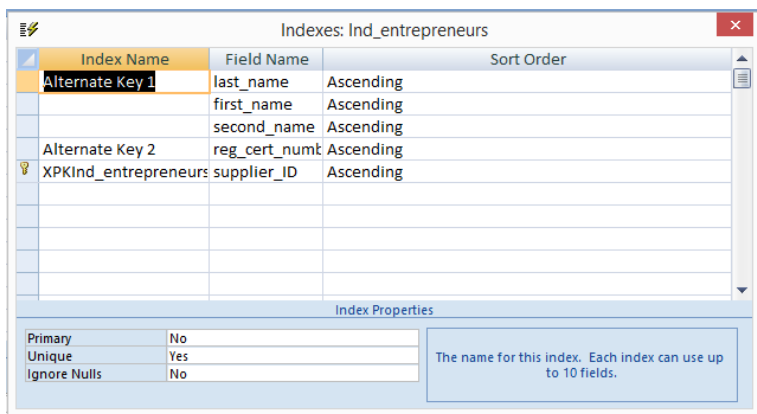


Рисунок 36

9. Збережіть створену модель і завершіть роботу.

10. Вимоги до звіту:

- 1) коротко охарактеризуйте основні етапи виконаної роботи;
- 2) продемонструйте створену базу даних (наприклад, схему даних), щоб проілюструвати її правильність;
- 3) опишіть проблеми, з якими ви зіткнулися, та проаналізуйте їх причини; опишіть, як ці проблеми були вирішені.



## СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Берко А.Ю. Системи баз даних та знань. Книга 2. Системи управління базами даних та знань: навч. посіб. / А.Ю. Берко, О.М. Верес, В.В. Пасічник. – Львів : «Магнолія-2006», 2013. – 584 с.
2. Buede D.M. The Engineering Design of Systems: Models and Methods / D.M. Buede, W.D. Miller. – John Wiley & Sons, 2016. – 592 p.
3. Sage A.P. Handbook of Systems Engineering and Management / A.P. Sage, W.B. Rouse. – John Wiley & Sons, 2014. – 1504 p.
4. Dennis A. Systems Analysis and Design / A. Dennis, B.H. Wixom, R.M. Roth. – John Wiley & Sons, 2014. – 448 p.
5. Davidson L. Pro SQL Server Relational Database Design and Implementation / L. Davidson, J. Moss. – Apress, 2016. – 791 p.
6. Foster E.C. Database Systems: A Pragmatic Approach / E.C. Foster, S. Godbole. – Apress, 2016. – 604 p.
7. Tillmann G. Usage-Driven Database Design: From Logical Data Modeling through Physical Schema Definition / G. Tillmann. – Apress, 2017. – 374 p.

Навчальне видання

**Методичні вказівки**

до виконання лабораторних робіт за темою  
«Моделювання даних з використанням нотації IDEF1X»

для студентів спеціальностей  
121 «Інженерія програмного забезпечення»,  
122 «Комп'ютерні науки»,  
126 «Інформаційні системи та технології»

Укладачі:

ОРЛОВСЬКИЙ Дмитро Леонідович  
КОПП Андрій Михайлович

Відповідальний за випуск проф. Гамаюн І.П.  
Роботу до видання рекомендував проф. Гамаюн І.П.

План 2022 р., поз. 276

Підп. до друку 26.10.2022.  
Гарнітура Times New Roman.  
Ум. друк. арк. 0,5.

---

Видавничий центр НТУ «ХП».  
Свідоцтво про державну реєстрацію ДК № 5478 від 21.08.2017 р.  
61002, Харків, вул. Кирпичова, 2

---

Самостійне електронне видання