

DESIGN OF AN E-COMMERCE DATABASE

ПРОЕКТУВАННЯ БАЗИ ДАНИХ
ЕЛЕКТРОННОЇ КОМЕРЦІЇ

Petrenko L.M.¹

Kravtsov S.O.²

DOI: <https://doi.org/10.30525/978-9934-26-351-4-7>

З кожним роком Інтернет усе більше проникає у життя людей, та знаходить своє місце в усіх сферах сучасного бізнесу, стає атрибутом ділового життя кожного, хто прагне процвітати і поліпшити економічну ситуацію в країні. Електронна комерція є одним із найперспективніших напрямів розвитку сучасної цифрової економіки, який має широкі можливості для розвитку, вдосконалення та поширення на майбутній період. Становлення і розвиток ринку електронної комерції в Україні відбувається в період розвитку ринкової економіки та елементів її інфраструктури. Цей етап співпадає з кардинальними змінами у світовій економіці, які визначаються процесами глобалізації фінансових ринків і міжнародною інтеграцією фінансових інститутів.

Зростання обсягу товарів та послуг, які поширюються на ринку засобами електронної комерції спостерігається протягом двох останніх десятиліть, і така тенденція продовжує зберігатись. Електронна комерція являє собою середовище, в якому юридична або фізична особа, що знаходиться в будь-якій точці економічної системи, може легко контактувати (з мінімальними витратами) з будь-якою іншою юридичною або фізичною особою з метою спільної діяльності: торгівлі, обміну ідеями, «ноу-хау» тощо.

Головними рушійними силами активного розвитку електронної комерції в світових масштабах, поряд з економічними, також є технологічні чинники, основними серед яких, зокрема, можемо виділити такі [1; 2].

¹ Candidate of Economic Sciences, Docent,
Kyiv National Economic University named after Vadym Hetman

² 1st Year Master's Student of the Major "Computer Science"
of the Educational and Professional Programme "Artificial Intelligence Systems"
Kyiv National Economic University named after Vadym Hetman

1. Глобальність електронної комерції — відсутність часових і просторових обмежень, для діяльності, поширення та розвитку дає змогу не лише провадити діяльність неперервно, за формулою «24/7», в будь-якій частині світу, але й швидко охоплювати нові сегменти ринку, поширюючи свою діяльність на нові ринки та території.

2. Зосередження бізнес-процесів і бізнес-інфраструктури в мережевому середовищі робить їх переважно віртуальними, а така віртуальність надає низку переваг перед матеріальними аналогами, зокрема, гнучкість, здатність до адаптації і швидких змін, низька (порівняно з матеріальною інфраструктурою) собівартість, швидкість та оперативність виробничих дій.

3. Основні ресурси і засоби виробництва в електронній комерції є не матеріальними, а інформаційними та програмними засобами, що робить простішими та дешевшими процеси їхнього створення, підтримання, нарощення та поширення.

4. Для електронної комерції, на відміну від традиційної торгівлі, характерним є дуже високий рівень технологій та інновацій. Це дає змогу не лише швидко й оперативно вдосконалювати бізнес-процеси шляхом запровадження нових підходів та технологій, але і змінювати модель ведення бізнесу на основі таких інновацій. Наприклад, поширення соціальних мереж як засобу комунікації та публічності створило можливість персоналізувати роботу з клієнтом на основі його особистих даних, поширених у мережі. Використання мобільних технологій дало змогу не лише постійно та неперервно комунікувати з клієнтом, а й дало поштовх для створення нового виду послуг «на ходу» тощо.

Оскільки основним засобом виробництва електронної комерції є інформаційні технології, то саме новітні ефективні вирішення в цьому напрямі є вирішальним ресурсом для підвищення конкурентоздатності та просування на ринку. Головною метою систем електронної комерції є підтримання процесу здійснення будь-яких ділових угод за допомогою інформаційних мереж, забезпечуючи ключові процеси онлайн. Головна проблема, пов'язана з автоматизацією вирішення таких задач, полягає у необхідності швидко оперувати великими масивами даних, аналізувати їх, проводити необхідні розрахунки. Цю проблему досить ефективно усуває впровадження реляційних баз даних. Організаційно-правові засади діяльності у сфері електронної комер-

ції, встановлення порядку вчинення електронних правочинів із використанням інформаційно-телекомунікаційних систем забезпечується законодавчо [4–7].

Під електронною комерцією (e-commerce) розуміють будь-які форми ділових угод, при яких взаємодія сторін здійснюється електронним способом замість фізичного обміну або безпосереднього фізичного контакту, і в результаті якого право власності або право користування товаром або послугою передається від однієї особи іншій. Електронній комерції притаманні всі ознаки підприємницької (комерційної) діяльності, яка проводиться за допомогою Інтернету чи інших мереж. Для повного та всебічного розуміння поняття «електронна комерція» слід виділити її основні ознаки:

- комплексність: електронна комерція є системним поняттям, яке включає в себе відносини у сфері електронних документів та документообігу, у сфері укладення правочинів, у сфері використання електронних підписів;

- наявність мережі електронного зв'язку: середовищем для вчинення правочинів та надання банківських послуг є мережа електронного зв'язку;

- диспозитивний підхід до вибору мережі: не вимагається використання конкретної мережі. Звичайно, здебільшого така діяльність здійснюється з використанням Інтернету, однак не виключається можливість використання інших;

- невичерпність переліку правочинів, які можуть укладатись з використанням мереж: законодавством не встановлено обмежень ні щодо предмету, ні щодо виду правочинів, які укладаються вищевказаним способом.

При цьому повинні бути забезпечені:

- онлайнове надання інформації про товари та послуги;

- онлайнове оформлення замовлення і видача рахунка;

- онлайнова оплата й обробка транзакцій.

Таким чином, «електронна комерція» відповідає процесам здійснення продажів.

До переваг використання засобів електронної комерції відносять [3]:

- істотне зниження витрат на організацію та функціонування інфраструктури господарської діяльності;

- зниження витрат на рекламу;
 - підвищення конкурентоспроможності та мінімізація собівартості товару;
 - скорочення часу виходу на ринки, оперативність укладення та виконання господарських договорів;
 - отримання доступу до глобальних ринків та істотне розширення ринку збуту товарів і послуг для продавця;
 - доступність інформації про товари і послуги, що реалізуються;
 - розширений вибір продукції, товарів і послуг для покупця.
- Виділяють 4 основні групи суб'єктів електронної комерції:
1. Споживачі (C-consumer) – фізичні особи.
 2. Бізнесові організації (B-business).
 3. Державні органи (G-government, A-administration).
 4. Фінансові установи, що забезпечують розрахунки між іншими суб'єктами електронної комерції.

Відповідно до цього, розглянемо саме облік відвантаженої та облік реалізованої готової продукції, а також облік покупців і постачальників для електронної комерції, а саме для інтернет-магазину. Облік відвантаження (відпуск готової продукції покупцям (замовникам)) в організаціях здійснюється на підставі накладних. В якості типової форми накладної може використовуватися «Накладна на відпуск товарно-матеріальних цінностей». Підставою для оформлення накладної на відпуск готової продукції в підрозділах організації є розпорядження керівника організації або уповноваженої ним особи, а також договір з покупцем (замовником). У ньому встановлюються права та обов'язки сторін, умови, терміни й вартість операції/операцій. Зміни та додаткові умови прописуються в додатковій угоді або специфікації.

На підставі накладних на відпуск готової продукції відділ збуту виписує рахунки-фактури у двох примірниках, перший з яких не пізніше 10 днів з дати відвантаження продукції (товару) надсилається (передається) покупцеві, а другий залишається в організації-постачальника для відображення в книзі продажу та нарахування податку на додану вартість. Рахунок має інформаційний характер і за своїм призначенням не відповідає ознакам первинного документа, оскільки в ньому не фіксується будь-яка господарська операція.

Collective monograph

Типової форми рахунку-фактури немає. При оформленні замовлення покупець здійснює операцію купівлі певної продукції. При купівлі товару (роботи, послуги) за готівку постачальник (виконавець) видає клієнту чек. Крім того, п. 11 ст. 8 Закону від 12.05.91 р. № 1023-ХІІ передбачає, що при продажі товару продавець зобов'язаний видати споживачеві розрахунковий документ встановленої форми, що підтверджує факт покупки, з відміткою про дату продажу [4]. На рис. 1 наведено інформаційну модель задачі автоматизації та оброблення інформації для електронної комерції:

До вхідних повідомлень, як правило, відносять документи, які використовуються для формування оперативних файлів бази даних; файли, що потрапляють на вхід задачі або їх сукупності, необхідної для виконання; документи та файли, які носять довідковий характер, можуть бути віднесені до умовно-постійної інформації. В табл. 1 наведено перелік, опис вхідних повідомлень.

Таблиця 1

Перелік і опис вхідних повідомлень

Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
ТТН	TTN	Документ	По мірі нових поставок	Постачальники
Відомість про постачальника	Partners	Документ	По мірі надходження нових постачальників	Постачальники
Договори	Treaties	Документ	По мірі укладення договорів	Відділ менеджменту
Анкета покупця	Users	Документ	По мірі реєстрації клієнтів/користувачів	Сайт

Вихідними повідомленнями можуть бути такі документи, які містять інформацію про реквізити постачальників до певної партії продукції, кількість продукції, її ціну, її категорії; інформацію по замовленням, поставкам, інформацію про користувачів/клієнтів, тощо. Дані про призначення і використання вихідного документа наведені в таблиці 2.

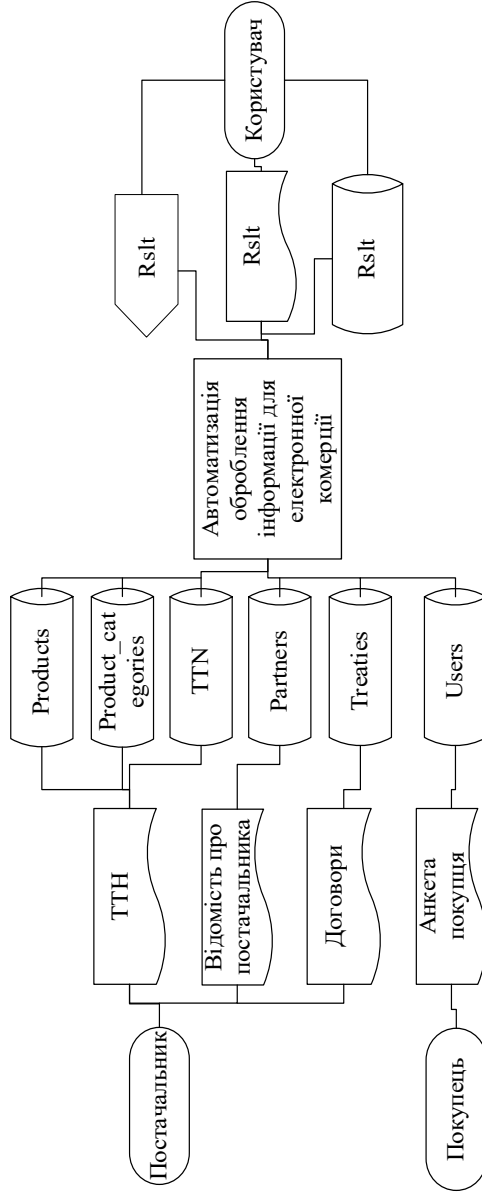


Рис. 1. Інформаційна модель задачі

Вихідні повідомлення

Назва вихідного повідомлення	Ідентифікатор	Форма подання	Періодичність видання	Користувачі інформації
Відомість про відвантажену продукції	shipped_production	Документ або візуальна екранна форма	По мірі необхідності	Керівник, відділи маркетингу та менеджменту, бухгалтерія
Відомість про замовлену продукцію	order_production	Документ або візуальна екранна форма	По мірі необхідності	Відділи маркетингу, менеджменту, бухгалтерія
Товарний чек	sales_receipt	Документ	По мірі необхідності	Покупець

Опишемо основні процедури перетворення даних.

1. Повний залишок товару ми визначаємо за такою формулою:

$$N^{zal} = \sum_{k=1}^l \sum_{i=1}^n N_{ik}^{nocm} - \sum_{k=1}^l \sum_{j=1}^m N_{jk}^{zam}, \quad (1)$$

де N^{zal} – наявний залишок всього товару,

N^{nocm} – кількість товару до замовлення (у поставках),

N^{zam} – кількість товару в замовленні,

i – індекс, що відповідає певній поставці,

j – індекс, що відповідає певному замовленню,

k – індекс, що відповідає певному виду продукції.

2. Загальна сума поставленого товару розраховується за формулою:

$$S^{nocm} = \sum_{d=1}^z \sum_{k=1}^l \sum_{i=1}^n N_{ikd}^{nocm} * \sum_{d=1}^z \sum_{k=1}^l \sum_{i=1}^n C_{ikd}^{nocm}, \quad (2)$$

де S^{nocm} – загальна сума поставленого товару,

C^{nocm} – закупівельна ціна одиниці товару,

d – індекс, що відповідає даті поставки.

3. Загальна сума замовленого товару розраховується за формулою:

$$S^{zam} = \sum_{d=1}^z \sum_{k=1}^l \sum_{j=1}^m N_{jkd}^{zam} * \sum_{d=1}^z \sum_{k=1}^l \sum_{j=1}^m C_{jkd}^{zam}, \quad (3)$$

де S^{zam} – загальна сума замовленого товару,

C^{zam} – роздрібна ціна одиниці товару.

Інфологічне моделювання полягає у виділенні інформаційних об'єктів (сутностей) предметної області, які підлягають зберіганню в БД, а також у визначенні характеристик об'єктів і взаємозв'язків між ними. Основними складовими елементами інфологічної моделі є: інформаційні об'єкти, атрибут, запит, запитувальний та структурний зв'язок.

Інформаційний об'єкт це деяка сутність предметної області, яку необхідно зберігати в базі даних з погляду прикладної програми чи користувача БД. Кожний інформаційний об'єкт описується певними його властивостями. Властивості об'єкту описують атрибути, що входять до їх складу.

Інфологічне проектування здійснюється шляхом синтезу атрибутів: спочатку визначається набір атрибутів, що характеризують предметну область, далі проводиться їх аналіз, на основі якого і виділяються інформаційні об'єкти.

Усі інформаційні об'єкти розробленої БД надходять із вхідними документами.

Інформаційні об'єкти описують атрибутами, що їх складають.

– **TTN** (**TTN_ID***, TTN_Num, TTN_Product_ID, TTN_Treaty_ID, TTN_Product_Quantity, TTN_Date, TTN_Product_Price, TTN_Amount)

– **Partners** (**Partner_ID***, Partner_IPN, Partner_Name, Partner_Address, Partner_MFO, Partner_Phone)

– **Treaties** (**Treaty_ID***, Treaty_Partner_ID, Treaty_From_Date, Treaty_To_Date, Treaty_Stock, Treaty_Amount)

– **Products** (**Product_ID***, Product_Name, Product_Price, Product_CategoryID, Product_Measure, Product_Update_Date)

– **Product_Categories** (**Category_ID***, Category_Name)

– **Orders** (**Order_ID***, Order_Product_ID, Order_User_ID, Order_Quantity, Order_Date, Order_Date_Payment, Order_Date_Delivered, Order_Address_Deliver, Order_Payment_Method)

– **User** (**User_ID***, User_First_Name, User_Second_Name, User_Password, User_Email, User_Reg_Date, User_Last_Order, User_Address)

Для того, щоб усунути аномалії оновлення, вилучення та надлишковості необхідно виконати умови нормалізації БД. Нормалізація відношень є ітераційним зворотнім процесом декомпозиції початкового відношення на декілька більш простих відношень меншої розмірності:

Перша нормальна форма (1НФ): атрибути повинні бути унікальними та атомарними.

Collective monograph

Друга нормальна форма (2НФ): проводиться аналіз на повноту функціональних залежностей (атрибут залежить від всього ключа, а не від окремих його елементів).

Третя нормальна форма (3НФ): проводиться аналіз на наявність у відношеннях транзитивних залежностей (залежностей між не ключовими атрибутами).

Четверта нормальна форма (4НФ): відношення аналізується на наявність багатозначних залежностей. Атрибут А знаходиться у багатозначній залежності від атрибута Б, якщо одному значенню Б може відповідати багато значень А.

Інформаційні об'єкти є нормалізованими, це видно з відношень між атрибутами в об'єктах. Графік інфологічної моделі для даної предметної області (див. графік 1).

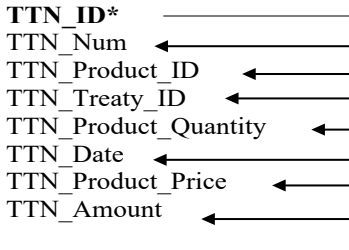
Всі інформаційні об'єкти відповідають умовам нормалізації (усі атрибути неподільні, мають унікальні імена – 1НФ, інформаційні об'єкти не містять неповних функціональних залежностей – 2НФ, немає транзитивних залежностей між неключовими атрибутами – 3,4НФ). Тому опишемо характеристику інформаційних об'єктів бази даних:

Таблиця 3

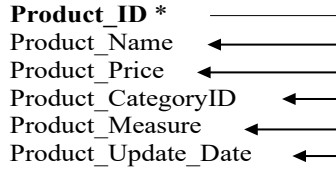
Складові інформаційного об'єкту TTN

Назва атрибута	Формат	Обов'язковість (необов'язков.)	Обмеження на право звертання	Первинний ключ (вторинний ключ)	Умови на допустимі значення	Індексне поле
Код накладної	N(10)	так	Склад, начальники відділів маркетингу і менеджменту	ПК	Унікальне	ІДД
Номер накладної	N(14)	так			-	
Код продукції	N(10)	так		ВК	Унікальне	ІДД
Код договору	N(10)	так		ВК	Унікальне	ІДД
Кількість продукції	N(10)	так				
Дата поставки	D(8)	так			-	
Ціна одиниці продукції	N(6,3)	так			-	
Сума за продукцію	N(7,3)	так			-	

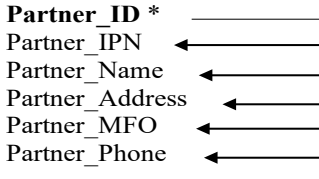
TTN



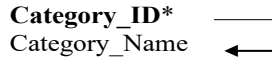
Products



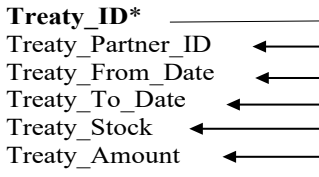
Partner



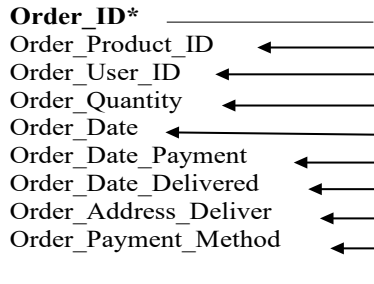
Product Categories



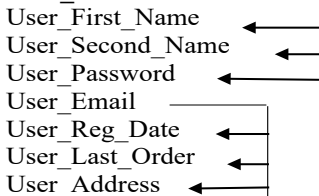
Treaties



Orders



User_ID*



Users

Графік 1

Таблиця 4

Складові інформаційного об'єкту Partners

Назва атрибута	Формат	Обов'язковість (необов'язков.)	Обмеження на право звертання	Первинний ключ (вторинний ключ)	Умови на допустимі значення	Індексне поле
Код постачальника	N(10)	так	Відсутні	ПК	Унікальне	ІДД
ІПН постачальника	N(15)	так		-		
Найменування постачальника	C(100)	так		-		
Адреса постачальника	C(80)	так		-		
МФО постачальника	C(30)	так		-		
Телефон постачальника	varchar(30)	так		-		

Таблиця 5

Складові інформаційного об'єкту Treaties

Назва атрибута	Формат	Обов'язковість (необов'язков.)	Обмеження на право звертання	Первинний ключ (вторинний ключ)	Умови на допустимі значення	Індексне поле
Код договору	N(8)	так	Відділи маркетингу і менеджменту	ПК	Унікальне	ІДД
Код постачальника	N(8)	так		ВК	Унікальне	ІДД
Початкова дата	D(8)	так		-		
Кінцева дата	D(8)	так		-		
Сума договору без ПДВ	N(7,3)	так		-		
Сума до сплати	N(7,3)	так		-		

Таблиця 6

Складові інформаційного об'єкту Products

Назва атрибута	Формат	Обов'язковість (необов'язков.)	Обмеження на право звертання	Первинний ключ (вторинний ключ)	Умови на допустимі значення	Індексне поле
Код продукції	N(8)	так	Відсутні	ПК	Унікальне	ІДД
Найменування продукції	C(100)	так		-		
Ціна продукції	N(6,3)	так		-		
Код категорії продукції	N(8)	так		ВК	Унікальне	ІДД
Одиниця виміру продукції	C(20)	так		-		
Дата оновлення продукції	D(8)	так		-		

Таблиця 7

Складові інформаційного об'єкту Product_Categories

Назва атрибута	Формат	Обов'язковість (необов'язков.)	Обмеження на право звертання	Первинний ключ (вторинний ключ)	Умови на допустимі значення	Індексне поле
Код категорії	N(8)	так	Відсутні	ПК	Унікальне	ІДД
Найменування категорії	C(50)	так		-		

Таблиця 8

Складові інформаційного об'єкту Orders

Назва атрибута	Формат	Обов'язковість (необов'язков.)	Обмеження на право звертання	Первинний ключ (вторинний ключ)	Умови на допустимі значення	Індексне поле
Код замовлення	N(8)	так	Відділ маркетингу	ПК	Унікальне	ІДД
Код продукції	N(8)	так		ВК	Унікальне	ІДД
Код користувача	N(8)	так		ВК	Унікальне	ІДД
Кількість продукції в замовленні	N(5)	так		-		

(Закінчення таблиці 8)

Назва атрибута	Формат	Обов'язковість (необов'язков.)	Обмеження на право звертання	Первинний ключ (вторинний ключ)	Умови на допустимі значення	Індексне поле
Дата замовлення	D(12)	так	Відділ маркетингу		-	
Дата оплати замовлення	D(12)	так				
Дата доставки замовлення	D(12)	так				
Адреса замовлення	C(100)	так				
Метод оплати замовлення	C(30)	так				

Таблиця 9

Складові інформаційного об'єкту Users

Назва атрибута	Формат	Обов'язковість (необов'язков.)	Обмеження на право звертання	Первинний ключ (вторинний ключ)	Умови на допустимі значення	Індексне поле
Код облікового запису покупця	N(8)	так	Користувач, начальник відділу маркетингу	ПК	Унікальне	ІДД
Ім'я покупця	C(50)	так				
Прізвище покупця	C(50)	так				
Пароль покупця	C(50)	так				
Електронна пошта покупця	C(100)	так				
Дата реєстрації облікового запису	D(8)	так				
Останнє замовлення покупця	D(8)	так				
Адреса покупця	C(100)	так				

Розглянемо інформаційні запити. Інформаційний запит – це словесний опис інформаційної потреби користувача чи прикладної програми. Запитувальний зв'язок будується на основі запиту. Він являє собою структурований опис інформаційного запиту, в якому відображено об'єкти, необхідні для його реалізації з урахуванням навігації між ними. У цьому разі під навігацією мають на увазі шлях інформаційного

пошуку при виконанні запиту. Можна дати ще таке визначення запитувального зв'язку. Запитувальні зв'язки застосовуються для визначення структурних зв'язків між об'єктами. Структурний зв'язок – це асоціації, що описують ієрархічні зв'язки між парами інформаційних об'єктів, один з яких виступає як власник, а інший – як підпорядкований об'єкт.

У даній БД основними запитамі, які відповідатимуть нашим вимогам, будуть такі:

1. Відомість про відвантажену продукцію

$$Z_{Products}^{TTN} + Z_{Treaties}^{TTN}$$

2. Відомість про замовлену продукцію

$$Z_{Products}^{Orders} + Z_{Users}^{Orders}$$

3. Товарний чек

$$Z_{Products}^{Orders}$$

Структурний зв'язок – це асоціації, що описують ієрархічні зв'язки між парами інформаційних об'єктів, один з яких виступає як власник, а інший – як підпорядкований об'єкт. Екземпляр структурного зв'язку являє собою екземпляр об'єкта власника та певну сукупність зв'язаних з ним екземплярів підпорядкованого об'єкта. Тепер необхідно визначити типи співвідношень між атрибутами:

Побудовану інфологічну модель можна спроектувати за допомогою CASE-засобу ERwin на двох рівнях: логічному та фізичному. Їх подано на рис. 2 та рис. 3 відповідно. На логічному рівні проектується інформаційна модель предметної області, тобто концептуальна (інфологічна) модель.

На фізичному рівні проектується модель з урахування специфіки та особливостей вибраної СКБД, заздалегідь обраної в діалоговому вікні створення моделі інтерфейсу CASE-засобу ERwin.

На етапі даталогічного проектування здійснюється перехід від інфологічної моделі ПО до логічної (даталогічної) моделі, яка підтримується засобами вибраної СКБД. Процес переходу від інфологічної до даталогічної моделі називається відображенням [4, с. 53].

В інтерфейсі SQL Server створюємо нову діаграму бази даних: натискаємо праву кнопку мишки на об'єкті Database Diagrams та обравши New Database Diagram, у спливаючому вікні потрібно додати усі таблиці (рис. 4).

Таблиця 10

Зв'язки інформаційних об'єктів

Головний ІВ	Ключ зв'язку		Підлеглий ІВ	Тип відносини	Тип зв'язку
	Головного ІВ	Підлеглого ІВ			
Partners	Partner_ID*	(Treaty_ID+Treaty_Partner_ID)*	Theatics	1-Б	неідентифікуючий обов'язковий
Treaties	Treaty_ID*	(TTN_ID+TTN_Product_ID+TTN_Treaty_ID)*	TTN	1-Б	неідентифікуючий обов'язковий
Products	Product_ID*	(TTN_ID+TTN_Product_ID+TTN_Treaty_ID)*	TTN	1-Б	неідентифікуючий обов'язковий
Products	Product_ID*	(Order_ID + Order_Product_ID+Order_User_ID)*	Orders	1-Б	неідентифікуючий обов'язковий
Product_Categories	Category_ID*	(Product_ID +Product_Category_ID)*	Products	1-Б	неідентифікуючий обов'язковий
Users	User_ID*	(Order_ID + Order_Product_ID+Order_User_ID)*	Orders	1-Б	неідентифікуючий обов'язковий

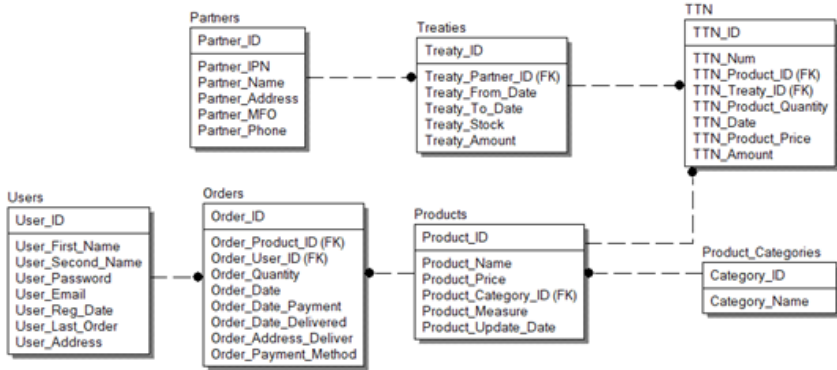


Рис. 2. Логічний рівень ІЛМ в CASE-засобі ERwin

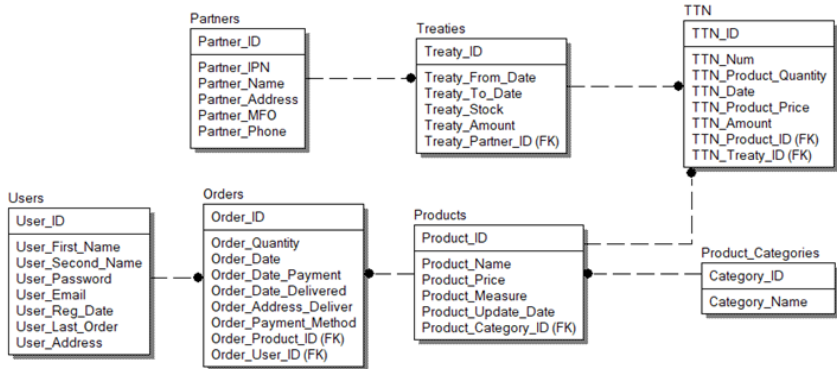


Рис. 3. Фізичний рівень ІЛМ в CASE-засобі ERwin

Таким чином ми перенесли логічну модель, створену у CASE-засобі ERwin, у СКБД Microsoft SQL server.

Після перенесення в СКБД формати даних зберігаються, а в інтерфейсі SQL Server будуть створені макети таблиць, що відповідають реляційним відношенням. Розглянемо їх на рис. 5.

Усі користувачі, що мають доступ до внесення змін, повинні уважно та обережно вводити нові дані або видаляти чи змінювати старі. Адміністратор, що вводить дані, має орієнтуватися в усіх зв'язках між створеними таблицями. Це є дуже важливою умовою користування БД. Заповнена база даних приведена нижче на рисунках (рис. 6–12).

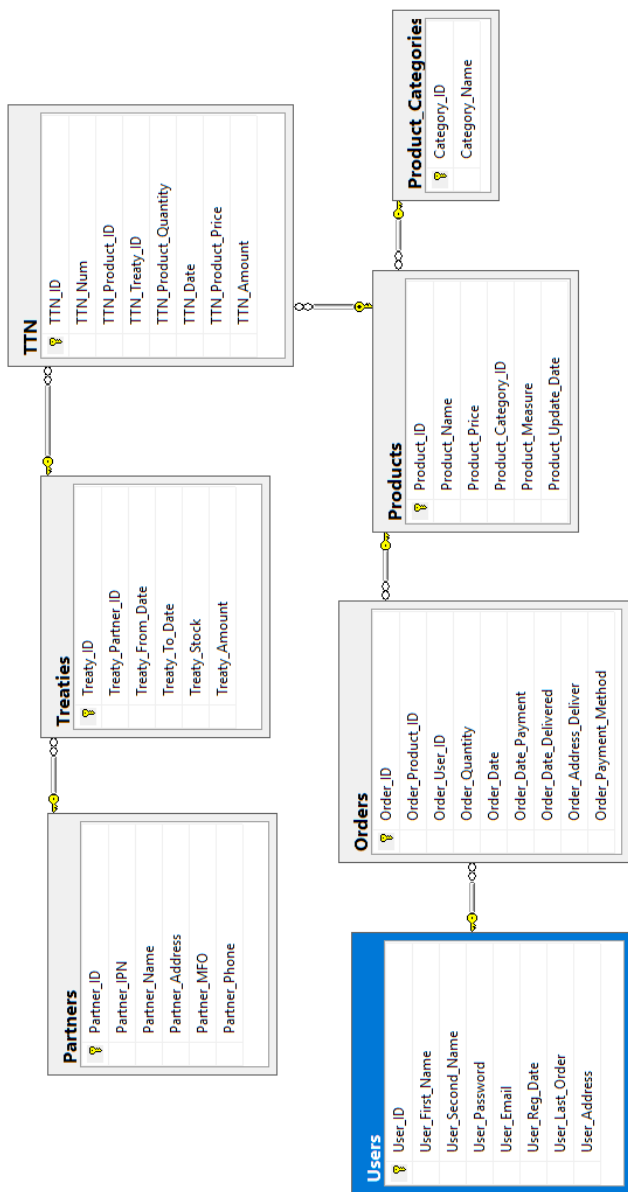


Рис. 4. Створена діаграма бази даних

Petrenko L.M., Kravtsov S.O.

DESKTOP-NMBN2R1...ch - dbo.Partners* ×		
Имя столбца	Тип данных	Разрешить ...
Partner_ID	int	<input type="checkbox"/>
Partner_IPN	varchar(30)	<input type="checkbox"/>
Partner_Name	varchar(100)	<input type="checkbox"/>
Partner_Address	varchar(80)	<input type="checkbox"/>
Partner_MFO	varchar(30)	<input type="checkbox"/>
Partner_Phone	varchar(30)	<input type="checkbox"/>

DESKTOP-NMBN2R1...ch - dbo.Treaties* ×		
Имя столбца	Тип данных	Разрешить ...
Treaty_ID	int	<input type="checkbox"/>
Treaty_Partner_ID	int	<input type="checkbox"/>
Treaty_From_Date	datetime	<input type="checkbox"/>
Treaty_To_Date	datetime	<input type="checkbox"/>
Treaty_Stock	float	<input type="checkbox"/>
Treaty_Amount	float	<input type="checkbox"/>

DESKTOP-NMBN2R1.kursach - dbo.TTN* ×		
Имя столбца	Тип данных	Разрешить ...
TTN_ID	int	<input type="checkbox"/>
TTN_Num	varchar(20)	<input type="checkbox"/>
TTN_Product_ID	int	<input type="checkbox"/>
TTN_Treaty_ID	int	<input type="checkbox"/>
TTN_Product_Quantity	int	<input type="checkbox"/>
TTN_Date	datetime	<input type="checkbox"/>
TTN_Product_Price	float	<input type="checkbox"/>
TTN_Amount	float	<input type="checkbox"/>

DESKTOP-NMBN2R1...h - dbo.Products* ×		
Имя столбца	Тип данных	Разрешить ...
Product_ID	int	<input type="checkbox"/>
Product_Name	varchar(100)	<input type="checkbox"/>
Product_Price	float	<input type="checkbox"/>
Product_Category_ID	int	<input type="checkbox"/>
Product_Measure	varchar(20)	<input type="checkbox"/>
Product_Update_Date	datetime	<input type="checkbox"/>

DESKTOP-NMBN2R1...h - dbo.Products* ×		
Имя столбца	Тип данных	Разрешить ...
Product_ID	int	<input type="checkbox"/>
Product_Name	varchar(100)	<input type="checkbox"/>
Product_Price	float	<input type="checkbox"/>
Product_Category_ID	int	<input type="checkbox"/>
Product_Measure	varchar(20)	<input type="checkbox"/>
Product_Update_Date	datetime	<input type="checkbox"/>

DESKTOP-NMBN2R1...oduct_Categories* ×		
Имя столбца	Тип данных	Разрешить ...
Category_ID	int	<input type="checkbox"/>
Category_Name	varchar(50)	<input type="checkbox"/>

DESKTOP-NMBN2R1...sach - dbo.Orders* ×		
Имя столбца	Тип данных	Разрешить ...
Order_ID	int	<input type="checkbox"/>
Order_Product_ID	int	<input type="checkbox"/>
Order_User_ID	int	<input type="checkbox"/>
Order_Quantity	int	<input type="checkbox"/>
Order_Date	datetime	<input type="checkbox"/>
Order_Date_Payment	datetime	<input type="checkbox"/>
Order_Date_Delivered	datetime	<input type="checkbox"/>
Order_Address_Deliver	varchar(100)	<input type="checkbox"/>
Order_Payment_Method	varchar(30)	<input type="checkbox"/>

Collective monograph

DESKTOP-NMBN2R1....ch - dbo.Partners						
	Partner_ID	Partner_IPN	Partner_Name	Partner_Address	Partner_MFO	Partner_Phone
	1	23948385	ТОВ "Інтел-Укр...	04073, м. Київ, ...	465783	(044) 5037750
	2	49586735	ТОВ "Асус-Укр...	03022, м. Київ, ...	196846	(044) 4942347
	3	85634756	ТОВ "Гігабайт ...	79005, м. Львів, ...	453675	(032) 2256184
	4	46673845	ТОВ "Кінгстон ...	03148, м. Київ, ...	946573	(067) 6933107
	5	37546891	ТОВ "Вестерн ...	79040, м. Львів, ...	256438	(032) 2420089

Рис. 6. Таблиця постачальників

DESKTOP-NMBN2R1.k...ch - dbo.Treaties						
	Treaty_ID	Treaty_Partner...	Treaty_From_D...	Treaty_To_Date	Treaty_Stock	Treaty_Amount
	1	1	2017-03-19 00:0...	2017-06-19 00:0...	75000	90000
	2	1	2017-03-19 00:0...	2017-06-19 00:0...	100000	120000
	3	2	2017-03-20 00:0...	2017-06-20 00:0...	30000	36000
	4	2	2017-03-20 00:0...	2017-06-20 00:0...	50000	60000
	5	3	2017-03-25 00:0...	2017-06-25 00:0...	15000	18000
	6	3	2017-03-25 00:0...	2017-06-25 00:0...	80000	96000
	7	4	2017-03-26 00:0...	2017-07-26 00:0...	10000	12000
	8	4	2017-03-26 00:0...	2017-07-26 00:0...	15000	18000
	9	5	2017-03-28 00:0...	2017-07-28 00:0...	40000	48000
	10	5	2017-03-28 00:0...	2017-07-28 00:0...	25000	30000

Рис. 7. Таблиця договорів

DESKTOP-NMBN2R1.kursach - dbo.TTN								
	TTN_ID	TTN_Num	TTN_Product_ID	TTN_Treaty_ID	TTN_Product_...	TTN_Date	TTN_Product_...	TTN_Amount
	1	37481857391	1	1	15	2017-03-22 00:0...	2500	45000
	2	37481857391	2	2	10	2017-03-22 00:0...	5300	63600
	3	57386902849	3	3	15	2017-03-23 00:0...	1000	18000
	4	57386902849	4	4	10	2017-03-23 00:0...	2000	24000
	5	19580375849	5	5	10	2017-03-28 00:0...	1000	12000
	6	19580375849	6	6	5	2017-03-28 00:0...	3800	22800
	7	36485037485	7	7	15	2017-03-29 00:0...	600	10800
	8	36485037485	8	8	10	2017-03-29 00:0...	800	9600
	9	58475938571	9	9	15	2017-04-01 00:0...	1500	27000
	10	58475938571	10	10	7	2017-04-01 00:0...	800	6720

Рис. 8. Таблиця поставок

Petrenko L.M., Kravtsov S.O.

DESKTOP-NMBN2R1....h - dbo.Products						
Product_ID	Product_Name	Product_Price	Product_Categ...	Product_Meas...	Product_Upda...	
1	Intel Core i3-71...	3361	1	од.	2017-05-20 18:0...	
2	Intel Core i5-66...	6615	1	од.	2017-05-20 18:0...	
3	Asus H110M-K ...	1430	2	од.	2017-05-20 18:0...	
4	Asus Prime B25...	2667	2	од.	2017-05-20 18:0...	
5	Gigabyte PCI-E...	1415	3	од.	2017-05-20 18:0...	
6	Gigabyte PCI-E...	4699	3	од.	2017-05-20 18:0...	
7	Kingston DDR3...	870	4	од.	2017-05-20 18:0...	
8	Kingston DDR4...	1130	4	од.	2017-05-20 18:0...	
9	Western Digital ...	2275	5	од.	2017-05-20 18:0...	
10	Western Digital ...	1313	5	од.	2017-05-20 18:1...	

Рис. 9. Таблица продукції

DESKTOP-NMBN2R1....oduct_Categories	
Category_ID	Category_Name
1	Процесори
2	Материнські п...
3	Відеокарти
4	Оперативна па...
5	Жорсткі диски
6	Системи охол...
7	Блоки живлення

Рис. 10. Таблица категорій продукції

DESKTOP-NMBN2R1....sach - dbo.Orders									
Order_ID	Order_Product...	Order_User_ID	Order_Quantity	Order_Date	Order_Date_Pa...	Order_Date_De...	Order_Address...	Order_Paymen...	
1	4	1	1	2017-05-02 14:1...	2017-05-02 14:3...	2017-05-05 16:1...	79035, м. Львів, ...	PayPal	
2	10	1	1	2017-05-02 14:1...	2017-05-02 14:3...	2017-05-05 16:1...	79035, м. Львів, ...	PayPal	
3	6	4	1	2017-05-03 13:3...	2017-05-03 13:5...	2017-05-06 17:4...	36009, м. Полт...	Visa	
4	14	4	1	2017-05-03 13:3...	2017-05-03 13:5...	2017-05-06 17:4...	36009, м. Полт...	Visa	
5	2	4	1	2017-05-03 13:3...	2017-05-03 13:5...	2017-05-06 17:4...	36009, м. Полт...	Visa	
6	7	2	1	2017-05-04 11:2...	2017-05-04 11:3...	2017-05-07 09:1...	79034, м. Львів, ...	MasterCard	
7	1	2	1	2017-05-04 11:2...	2017-05-04 11:3...	2017-05-07 09:1...	79034, м. Львів, ...	MasterCard	
8	11	7	1	2017-05-04 19:1...	2017-05-04 19:3...	2017-05-08 15:5...	61054, м. Харків...	Privat24	
9	2	10	1	2017-05-05 16:3...	2017-05-05 16:5...	2017-05-06 17:4...	04112, м. Київ, ...	PayPal	
10	4	10	1	2017-05-05 16:3...	2017-05-05 16:5...	2017-05-06 17:4...	04112, м. Київ, ...	PayPal	
11	6	10	1	2017-05-05 16:3...	2017-05-05 16:5...	2017-05-06 17:4...	04112, м. Київ, ...	PayPal	
12	8	10	1	2017-05-05 16:3...	2017-05-05 16:5...	2017-05-06 17:4...	04112, м. Київ, ...	PayPal	

Рис. 11. Таблица замовлень

Collective monograph

User_ID	User_First_Name	User_Second_...	User_Password	User_Email	User_Req_Date	User_Last_Order	User_Address
1	Мартин	Мамедов	gxYhbySwt1CUG	MamedovMart...	2017-04-09 09:3...	NULL	79035, м. Львів,...
2	Дар'я	Гопко	BVrSAtkBlcgH	GopkoDarya220...	2017-04-09 16:4...	NULL	79034, м. Львів,...
3	Ярослава	Іліна	Um8NHjF47p59	IliinaYaroslava51...	2017-04-14 14:3...	NULL	21034, м. Вінни...
4	Рада	Сморчкова	YMERt0iZicPO	SmorchkovaRa...	2017-04-16 22:1...	NULL	36009, м. Полт...
5	Юрій	Воронін	EPmHHWrREOV8	VoroninYuriy11...	2017-04-17 18:3...	NULL	36007, м. Полт...
6	Костянтин	Чаурін	OdR3Bd2eFpiw	ChaurinKonsta...	2017-04-18 11:4...	NULL	79015, м. Львів,...
7	Ангеліна	Павлова	MDNFrl0aGKW0	PavlovaAngelin...	2017-04-19 08:0...	NULL	61054, м. Харкі...
8	В'ячеслав	Соколов	FCSHkVhujbm4	SokolovVyache...	2017-04-20 17:1...	NULL	79014, м. Львів,...
9	Андрій	Чистяков	OkvuyWG9odq1	ChistyakovAnd...	2017-04-20 12:5...	NULL	04080, м. Київ, ...
10	Артем	Тарський	64mBYvN40pGe	TarskiyArtemiy...	2017-04-21 19:1...	NULL	04112, м. Київ, ...

Рис. 12. Таблиця облікових записів покупців

Програма на об'єктно орієнтованій мові C#:

```
using Bank.DAL.EFContext;
using Bank.DAL.Entities;
using Microsoft.EntityFrameworkCore.Internal;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleTest
{
    public class Queries
    {
        public void MusicLovers()
        {
            using (var db = new BankContext())
            {
                var likes = db.Likes
                    .GroupBy(s => s.SongId)
                    .Select(g => new
                    {
                        g.Key,
                        Count = g.Count()
                    })
                    .Where(p => p.Count > 1);
                var users = db.Likes
                    .Where(p => likes.Any(s => s.Key == p.SongId));
                var musicLovers = db.Likes
                    .Where(p => !users.Any(s => s.UserId == p.UserId))
                    .Select(p => p.User);
                foreach (var user in musicLovers)
                {
                    Console.WriteLine($"{user.Id} {user.FirstName} {user.LastName}");
                }
            }
        }
    }
}
```

```
}
}
}

public void NewYorkGenre()
{
    using(var db = new BankContext())
    {
        var gernre = db.Songs.Join(
            db.Likes,
            p => p.Id,
            s => s.SongId,
            (x, y) => new
            {
                x.Id,
                y.Score,
                y.User.City.Name
            }).Join(
            db.SongGenres,
            p => p.Id,
            s => s.SongId,
            (x, y) => new
            {
                x.Id,
                x.Score,
                CityName = x.Name,
                GenreName = y.Genre.Name
            })
        .Where(p => p.CityName == "New York")
        .GroupBy(p => p.GenreName)
        .Select(p => new
        {
            GenreName = p.Key,
            Sum = p.Sum(s => s.Score)
        })
        .OrderByDescending(p => p.GenreName)
        .First();
        Console.WriteLine($"{gernre.GenreName} {gernre.Sum}");
    }
}

public void TheBeatles()
{
    using (var db = new BankContext())
    {
        var users = db.Users.Where(p => p.City.Country.Name == "USA");
        var likes = db.Likes.Join(
            db.SongArtists,
            p => p.SongId,
            t => t.SongId,
            (x, y) => new
```

Collective monograph

```
{
x.UserId,
y.Artist
})
.Where(p => p.Artist.Name == "The Beatles" && users.Any(s => s.Id == p.
UserId)).Count();
Console.WriteLine(likes);
}
}
public void TheMostPopularRockNRoll()
{
using (var db = new BankContext())
{
var songs = db.Songs.Join(
db.Likes,
s => s.Id,
l => l.SongId,
(x, y) => new
{
x.Id,
x.Title,
y.Score,
y.Date,
})
.Join(
db.SongGenres,
s => s.Id,
g => g.SongId,
(x, y) => new
{
x.Id,
x.Score,
x.Title,
x.Date,
y.Genre.Name
})
.Where(p => p.Date.Year == 2012 && p.Name == "Rock-n-Roll")
.GroupBy(p => new { p.Id, p.Title })
.Select(p => new
{
p.Key.Id,
p.Key.Title,
TotalScore = p.Sum(p => p.Score)
})
.OrderByDescending(p => p.TotalScore);
foreach (var song in songs)
{
Console.WriteLine($"Id:{song.Id} Title:{song.Title} Total:{song.TotalScore}");
}
}
}
```

```
}
public void RecomendationToJohn()
{
using (var db = new BankContext())
{
var likes = db.Likes
.Join(
db.Users,
l => l.UserId,
u => u.Id,
(x, y) => new
{
x.SongId,
y.Email
})
.Where(p => p.Email == "john@mail.com");

var songs = db.SongGenres
.Where(p => p.Genre.Name == "Heavy Metal")
.Where(p => !likes.Any(s => s.SongId == p.SongId))
.OrderByDescending(p => p.Song.Likes.Sum(p => p.Score))
.Select(p => p.Song)
.Join(db.SongArtists, s => s.Id, a => a.SongId, (x, y) => new { x.Title,
y.Artist.Name})
.Take(3);
foreach (var song in songs)
{
Console.WriteLine($"{song.Title} {song.Name}");
}
}
}
public void DeleteLikeWithSmallScore()
{
using (var db = new BankContext())
{
foreach (var like in db.Likes)
{
Console.WriteLine($"{like.SongId} {like.UserId} {like.Score}");
}
Console.WriteLine(new string('-', 10));

var likes = db.Likes.Where(p => p.Score < 3);
db.Likes.RemoveRange(likes);
db.SaveChanges();
foreach (var like in db.Likes)
{
Console.WriteLine($"{like.SongId} {like.UserId} {like.Score}");
}
}
}
```


Collective monograph

```
public void DeleteGenre()
{
    using (var db = new BankContext())
    {
        try
        {
            var genre = db.Genres.Where(p => p.Name == "Heavy Metal").First();
            db.Remove(genre);
            db.SaveChanges();
        }
        catch(Exception e)
        {
            Console.WriteLine(e.InnerException.Message);
        }
    }
}

public void UpdateUsers()
{
    using (var db = new BankContext())
    {
        foreach (var user in db.Users)
        {
            Console.WriteLine($"{user.FirstName} {user.LastName}");
        }
        var users = db.Users;
        foreach (var user in users)
        {
            user.FirstName = user.FirstName.First().ToString();
            user.LastName = user.LastName.First().ToString();
        }
        db.SaveChanges();
        foreach (var user in db.Users)
        {
            Console.WriteLine($"{user.FirstName} {user.LastName}");
        }
    }
}

public void CheckConstraint()
{
    using (var db = new BankContext())
    {
        try
        {
            db.Likes.First().Score = 10;
            db.SaveChanges();
        }
        catch(Exception e)
        {
            Console.WriteLine(e.InnerException.Message);
        }
    }
}
```

```
}
}
}
public void IndexUniqueCheck()
{
    using (var db = new BankContext())
    {
        try
        {
            db.Users.Find(2).Email = "john@mail.com";
            db.SaveChanges();
        }
        catch(Exception e)
        {
            Console.WriteLine(e.InnerException.Message);
        }
    }
}
}
using Dotify.DAL.EFContext;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Data;
namespace ConsoleTest
{
    public class Transactions
    {
        public void ReadUncommitted()
        {
            using (var db = new DotifyContext())
            {
                Task.Run(() => DirtyRead());
                using (var transaction = db.Database.BeginTransaction(IsolationLevel.
ReadUncommitted))
                {
                    Thread.Sleep(500);
                    var city = db.Cities.First();
                    Console.WriteLine(city.Name);
                    Thread.Sleep(2000);
                    db.Entry(city).Reload();
                    Console.WriteLine(city.Name);
                    transaction.Rollback();
                }
            }
        }
    }
}
```

Collective monograph

```
}
private void DirtyRead()
{
    using (var db = new DotifyContext())
    {
        using (var transaction = db.Database.BeginTransaction(IsolationLevel.
ReadUncommitted))
        {
            db.Cities.First().Name = "LALA";
            db.SaveChanges();
            Thread.Sleep(1000);
            transaction.Rollback();
        }
    }
}
public void ReadCommitted()
{
    using (var db = new DotifyContext())
    {
        Task.Run(() => NonRepeatableRead());
        using (var transaction = db.Database.BeginTransaction(IsolationLevel.
ReadCommitted))
        {
            var city = db.Cities.First();
            Console.WriteLine(city.Name);
            Thread.Sleep(2000);
            db.Entry(city).Reload();
            Console.WriteLine(city.Name);
            transaction.Commit();
        }
    }
}
private void NonRepeatableRead()
{
    using (var db = new DotifyContext())
    {
        using (var transaction = db.Database.BeginTransaction(IsolationLevel.
ReadCommitted))
        {
            Thread.Sleep(1000);
            db.Cities.First().Name = "LALA";
            db.SaveChanges();
            transaction.Commit();
        }
    }
}
public void RepeatableRead()
{
    using (var db = new DotifyContext())
    {
```

```
Task.Run(() => PhantomRead());
using (var transaction = db.Database.BeginTransaction(IsolationLevel.
RepeatableRead))
{
    var country = db.Countries.Where(p => p.Name == "MyCountry").
FirstOrDefault();
    if(country == null)
        Console.WriteLine($"Not Found Count:{db.Countries.Count()}");
    else
        Console.WriteLine(country.Name);
    Thread.Sleep(2000);
    country = db.Countries.Where(p => p.Name == "MyCountry").FirstOrDefault();
    Console.WriteLine($"{country.Name} Count:{db.Countries.Count()}");
    transaction.Commit();}}}
private void PhantomRead()
{
    using (var db = new DotifyContext())
    {
        using (var transaction = db.Database.BeginTransaction(IsolationLevel.
RepeatableRead))
        {
            Thread.Sleep(1000);
            db.Countries.Add(new Dotify.DAL.Entities.Country()
            {
                Name = "MyCountry"
            });
            db.SaveChanges();
            transaction.Commit();}}}
public void Serealizable()
{
    using (var db = new DotifyContext())
    {
        Task.Run(() => PhantomRead());
        using (var transaction = db.Database.BeginTransaction(IsolationLevel.
Serializable))
        {
            var country = db.Countries.Where(p => p.Name == "MyCountry").
FirstOrDefault();
            if (country == null)
                Console.WriteLine($"Not Found Count:{db.Countries.Count()}");
            else
                Console.WriteLine($"{country.Name} Count:{db.Countries.Count()}");
            Thread.Sleep(2000);
            country = db.Countries.Where(p => p.Name == "MyCountry").FirstOrDefault();
            if (country == null)
                Console.WriteLine($"Not Found Count:{db.Countries.Count()}");
            else
                Console.WriteLine($"{country.Name} Count:{db.Countries.Count()}");
            transaction.Commit();}}}}}
```

У багатьох випадках електронна комерція дозволяє скоротити шлях перепродажу продукту від виробника до споживача. Це можливо завдяки використанню інтернет-технологій, що надають можливість ефективної прямої взаємодії з кінцевим споживачем, тому компанії можуть виконувати роль, яку традиційно виконували проміжні постачальники. Це також дозволяє накопичувати інформацію про усі продажі та про усіх клієнтів, що стосується і покупців і постачальників. У свою чергу це дає можливість виконати досконалий бізнес-аналіз та маркетингові дослідження за допомогою сучасних технологій з використанням сховищ даних та баз знань.

Найбільшою перевагою електронної комерції є суттєве зниження витрат на оформлення угоди та її подальше обслуговування. Тому бізнес-процеси, які можуть бути переведені на електронну основу мають потенціал зниження витрат на них, що у свою чергу призводить до зниження собівартості товару чи послуги. Найвідомішим прикладом здійснення електронної комерції є інтернет-магазин, який являє собою веб-ресурс з каталогом продукції та можливістю замовлення і оплати товарів, які сподобались покупцю. Все більше компаній у світі впроваджують рішення електронної комерції у своєму бізнесі, тому розроблена БД актуальна та зручна у використанні. Розроблена база даних у перспективі є повністю готовою для впровадження у бізнесі сфери електронної комерції.

Список використаних джерел:

1. Allen R. Top E-commerce Trends to inform your 2017 marketing strategy. *Smart Insights*. 2017. URL: <https://www.smartinsights.com/ecommerce/ecommerce-strategy/top-ecommerce-trends-inform-2017-marketing-strategy/> (дата звернення: 01.09.2023).
2. Radde B. Big Data: Future Of E-Commerce. 2014. URL: <https://www.linkedin.com/pulse/20141112200850-1172947-big-data-future-of-e-commerce> (дата звернення: 19.08.2023).
3. Юдін О., Макарова М., Лавренюк Р. Системи електронної комерції: створення, просування і розвиток. Полтава : РВВ ПУЕТ, 2011. 201 с.
4. Про електронну комерцію : Закон України від 03.09.2015 р. № 675-VIII : станом на 19 листопада 2022 р. URL: <https://zakon.rada.gov.ua/laws/show/675-19#Text> (дата звернення: 09.08.2023).
5. Про електронні довірчі послуги : Закон України від 05.10.2017 р. № 2155-VIII : станом на 1 січня 2023 р. URL: <https://zakon.rada.gov.ua/laws/show/2155-19#Text> (дата звернення: 09.08.2023).
6. Конвенція про захист осіб у зв'язку з автоматизованою обробкою персональних даних : Конвенція, Міжнар. док. від 28.01.1981 р. № 994 326. URL: https://zakon.rada.gov.ua/laws/show/994_326#Text (дата звернення: 09.08.2023).
7. Convention on Information and Legal Co-operation concerning «Information Society Services» : European Treaty Series – No. 180 of 04.10.2001. URL: <https://rm.coe.int/1680080625> (дата звернення: 10.08.2023).