

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ**

Кафедра цифрової економіки та бізнес-аналітики

**КУРСОВА РОБОТА
з навчальної дисципліни
“Проектування та адміністрування БД і СД”**

**на тему:
«Інформаційна система функціонування магазину книг»**

Галузь знань: _____ 05 «Соціальні та поведінкові науки» _____

Спеціальність: _____ 051 «Економіка» _____

Спеціалізація: _____ «Інформаційні технології в бізнесі» _____

Освітній ступінь: _____ бакалавр _____

Науковий керівник:

д.е.н., зав. каф. Старух А. І.

_____ (прізвище, ім'я, по-батькові)

_____ (підпис)

“ ____ ” _____ 2022 р.

Виконавець:

Ленько М. Ю.

_____ (прізвище, ім'я, по-батькові)

УФЕ-31с група
_____ (підпис)

“ ____ ” _____ 2022 р.

Загальна кількість балів _____

_____ (підпис, ПІП членів комісії)

Львів 2022

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. АНАЛІЗ ВИМОГ	5
1.1 Постановка завдання.....	5
1.2 Аналіз вимог	6
2.3 Розробка моделі варіантів використання.....	9
РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ	13
2.1 Поняття реляційних баз даних.....	13
2.2 Опис моделі	14
2.3 Нормалізація відношень	17
2.4 Визначення типів даних	19
2.5 Обмеження цілісності даних.....	25
2.6 Реалізація SQL-скрипту.....	28
ВИСНОВКИ.....	33
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	34
ДОДАТКИ.....	35

ВСТУП

Актуальність теми дослідження. В сучасному світі кожна людина потребує отримувати та аналізувати інформацію, яка може зберігатися, змінюватися та раціонально використовуватися в процесі задоволення інформаційних потреб людства в електронному вигляді за допомогою різноманітних інформаційних систем.

Попри те, що інформаційні технології дуже популяризуються та розвиваються у шаленому ритмі, дехто ще передає інформацію у паперовому варіанті, на що відповідно йде багато часового ресурсу. Власники книгарень, що є, по суті, крамницями роздрібною торгівлі книгами й іншим інформаційним товаром, зацікавлені в отриманні максимального прибутку з мінімальними затратами ресурсів. Автоматизація роботи таких організацій значно поліпшує їх діяльність, і, відповідно, прибуток. Оскільки внесення всіх даних по книгарні в одну інформаційну систему дозволить систематизувати їх, і спростити працівникам доступ до необхідної для роботи інформації, наприклад кількість куплених книг, пошук наявних книг в магазині і тд. Все це в результаті дає значне заощадження коштів та часу потрібних для вилучення тої чи іншої звітності, а це в свою чергу впливає на ефективність роботи магазину.

Як на мене, найкращою інформаційною системою для зберігання даних, є БД. Оскільки за останні десятиліття вони докорінно змінили та покращили роботу багатьох організацій та лідирують серед списку інформаційних систем. Розвиток технологій баз даних призвело до створення досить потужних та зручних у використанні програм. Вони є надійними для зберігання даних продукції, відомостей про клієнтські бази і тд за допомогою звичних таблиць.

Мета і завдання дослідження. Метою курсової роботи є створення бази даних, для покращення ефективності роботи книгарні за рахунок

вдосконалення рекомендаційної функції асортименту її системи обслуговування користувачів.

Для досягнення поставленої мети, визначено наступні *завдання*:

- охарактеризувати інформаційну систему для функціонування магазину книг;
- ознайомитись та проаналізувати предметну область;
- розробити архітектуру та спроектувати структуру бази даних;

Об'єкт дослідження. Особливість розробки інформаційної системи для функціонування магазину книг за допомогою системи управління базами даних MYSQL.

Предмет досліджень. Теоретичні, методичні та практичні аспекти для розробки бази даних.

Використане програмне забезпечення. Для досягнення поставленої мети використовувалось середовище для створення баз даних MySQL Workbench.

РОЗДІЛ 1. АНАЛІЗ ВИМОГ

1.1 Постановка завдання

В сучасному світі надзвичайно велику роль в суспільному житті відіграє інформація. Вона є основою при прийнятті рішень для здійснення будь-яких дій. Дослідження інформації є дуже актуальною темою вже протягом довгого часу. До винайдення комп'ютерів інформація досліджувалась лише як основа для діяльності в економіці та інших сферах. В нашу інформаційну епоху інформація є також товаром і засобом виробництва. Найважливішим наслідком впровадження інформаційних технологій є підвищення продуктивності праці при обробці даних.

Інформація в економіці має надзвичайно великий вплив в усіх галузях її функціонування. Сучасна інформатизація якісно поліпшує досягнення цілей діяльності організацій та підприємств і є умовою розвитку країни. Інформаційна економіка - це новий тип економічної системи, до якої відбувається активний перехід. У ній пріоритетними є знання як виробничий ресурс та продукція для споживання. Її розвиток відбувається дуже стрімко разом з розвитком науково-технічного прогресу. Проникнення інформатизації у всі сфери суспільного життя є причиною того, що вивчення інформаційної економіки потребує комплексного вивчення. Сучасні інформаційні й телекомунікаційні технології стали настільки важливою частиною інфраструктури суспільства, що від них залежить не тільки технологічний, але й соціальний поступ, економічна конкурентоспроможність країни в цілому, її місце у світовій економіці, роль у міжнародному розподілі праці, здатність розвивати демократичні інститути, зростання нових робочих місць і т.д.

Розробка та створення бази даних для функціонування магазину книг значно спростить та покращить роботу менеджерів та всіх працівників. Оскільки всі дані та інформація буде зберігатись в одній інформаційній системі, а саме:

- список та інформація про кожного клієнта, який зробив замовлення в магазині;
- інформація про наявні книги та кількість проданих;
- перелік замовлень та всі суми;
- можливість перегляду активності магазину і тд.

1.2 Аналіз вимог

SWOT-аналіз – це інструмент, який може допомогти вам проаналізувати, що ваша компанія робить найкраще зараз, і розробити успішну стратегію на майбутнє. SWOT також може виявити сфери бізнесу, які стримують вас або які можуть використовувати ваші конкуренти, якщо ви не захистите себе.

SWOT-аналіз досліджує як внутрішні, так і зовнішні фактори, тобто те, що відбувається всередині та за межами вашої організації. Тож деякі з цих факторів будуть під вашим контролем, а деякі — ні. У будь-якому випадку найрозумніші дії, які ви можете зробити у відповідь, стане зрозумілішим, коли ви виявите, зафіксуєте та проаналізуєте якомога більше факторів.

Для розуміння рівня конкуренції на ринку зробимо SWOT-аналіз для однієї із найпопулярніших книгарень «Видавництво Старого Лева».

Сильні сторони <ul style="list-style-type: none"> • наявність постійних клієнтів; • безкоштовний Wi-Fi; • мережу впізнають; • керівник здійснює необхідний рівень контролю; • можливість читати книгу в книгарні і пити каву чи чай. 	Слабкі сторони <ul style="list-style-type: none"> • мала кількість офлайн книгарень; • збільшення затрат (оренда офіса, приміщень для кав'ярень); • недостатнє фінансування маркетингу, включаючи рекламу, знижує можливість приваблення нових покупців, і як наслідок, товарооборот.
Можливості <ul style="list-style-type: none"> • ненасичений ринок українською книгою, необмежені можливості для росту; • поширення української книги закордоном; • збільшення платформ, де читачі зможуть проводити свій час з улюбленою книгою; 	Загрози <ul style="list-style-type: none"> • збільшення орендної плати може знизити прибутковість магазинів; • заповнення книжкового ринку України російськомовною літературою; • розвиток та розширення конкурентів.

4Р маркетингу є ключовими факторами, які беруть участь у маркетингу товару чи послуги. Це продукт, ціна, місце та просування товару чи послуги. 4Р, що часто називають комплексом маркетингу, обмежені внутрішніми та зовнішніми факторами в загальному бізнес-середовищі, і вони значною мірою взаємодіють один з одним.

4Р використовуються компаніями для визначення деяких ключових факторів для свого бізнесу, включаючи те, що споживачі хочуть від них, як їхній продукт або послуга задовольняють або не відповідають цим потребам, як їхній продукт або послугу сприймають у світі, як вони стоять від своїх конкурентів і як вони взаємодіють зі своїми клієнтами.

Продукт Тут можна знайти книги, журнали, аудіо книги для різного віку та вподобання. Також є різна сувенірна продукція(листівки, еко сумки, посуд і тд.) Чай, кава та солодке.	Ціни Приємні ціни, певний відсоток йде на благодійність. Часто є різні знижки та можливість 10% кешбеку.
Місце Книги можна придбати в різних книгарнях, або на сайті. Прочитати можна в трьох кав'ярнях-книгарнях у Львові.	Просування Планується збільшення кількості кав'ярень. Часто проводяться зустрічі читачів з авторами. Беруть участь у різних ярмарках. Та міжнародних виставках.

Відповідно до цього аналізу було описано наступні вимоги:

- зручне оформлення сайту;
- широкий асортимент товару;
- присутність промокодів та акцій;
- сторінки в соціальних мережах;
- використання реклами для просування
- підтримка для клієнтів;
- місця для читання книг;
- основна інформація про компанію;
- аудіо формати книг;
- відгуки.

2.3 Розробка моделі варіантів використання

Для зрозумілішого опису функціонування магазину книг, а саме бази даних, яка для цього була створена, мною було зроблена діаграма прецедентів(рис.1.2.1).

Загальні компоненти включають:

Актори: користувачі, які взаємодіють із системою. Актором може бути людина, організація або стороння система, яка взаємодіє з вашим додатком або системою. Вони повинні бути зовнішніми об'єктами, які виробляють або споживають дані.

Система: конкретна послідовність дій і взаємодій між акторами та системою. Систему також можна назвати сценарієм.

Цілі: Кінцевий результат більшості випадків використання. Успішна діаграма повинна описувати дії та варіанти, які використовуються для досягнення мети.

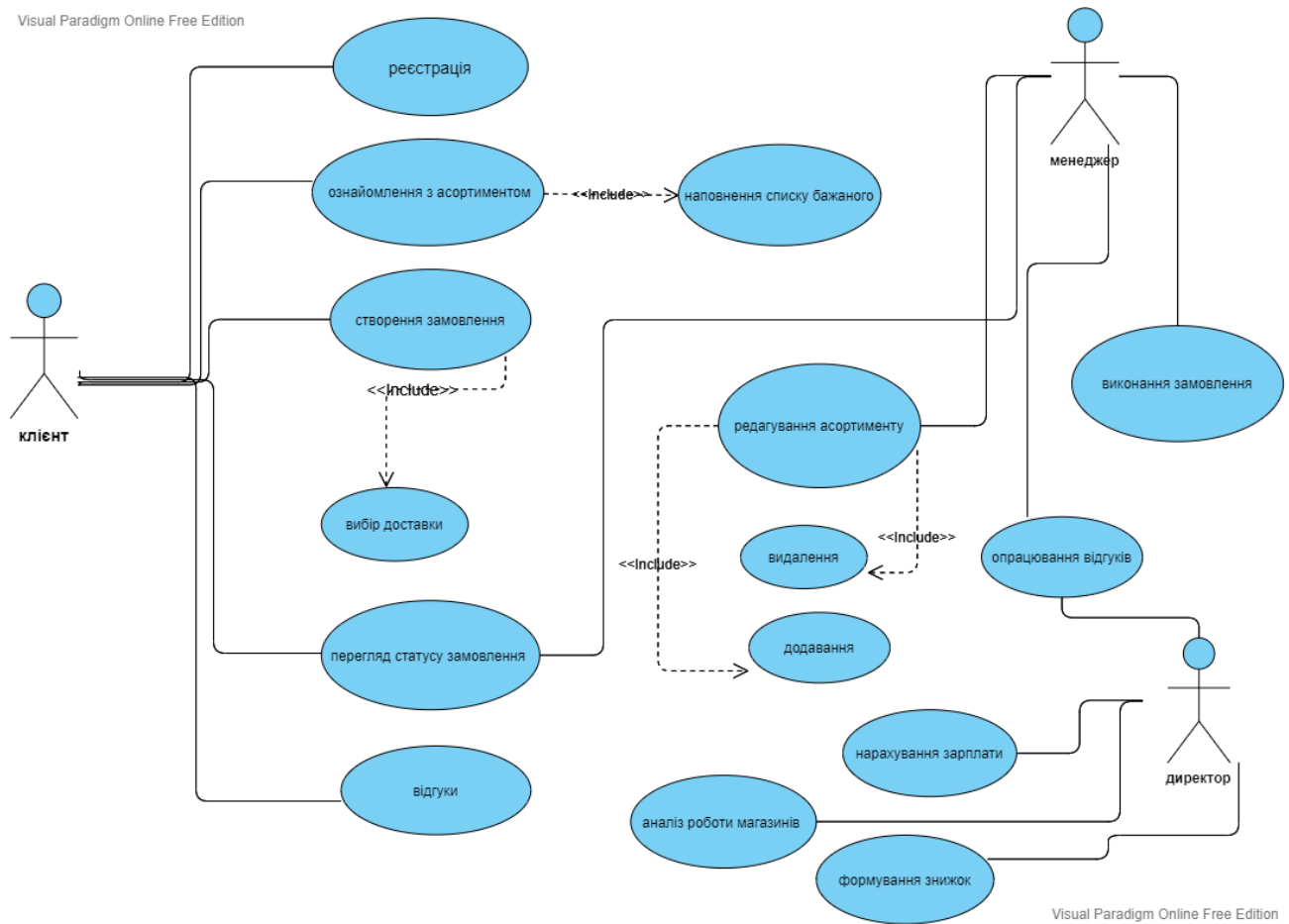


Рис. 1.2.1 Діаграма прецедентів для функціонування магазину книг

Дана діаграма включає в себе трьох акторів, кожен з яких може робити ту чи іншу справу.

1. Клієнт

- реєструється на сайті, створює свій власний кабінет, для того, щоб отримувати знижки, заповняти список бажаного і тд;
- ознайомлюється з асортиментом, переглядає книги по жанрах, авторах, видавництва для швидкості. Також під час цього моменту користувач має можливість додавати книги в так би мовити список бажаного, щоб потім одразу при можливості придбати ту чи іншу книжку;

- створює замовлення, обирає книги, їх кількість і так далі. Також обирає найоптимальніший спосіб доставки;
- опісля створеного замовлення, в клієнта є можливість переглядати його статус, або на власному кабінеті, або в електронній пошті, де приходять повідомлення від магазину;
- після отримання замовлення клієнт по бажанню може залишити відгук на сайті, як про саму книгу, так і про роботу менеджера, який опрацьовував замовлення.

2. Менеджер

- перегляд статусу замовлень, для моніторингу замовлень, реакції, якщо замовлення довго не відправляється і тд;
- опрацьовування та виконання замовлення;
- редагування асортименту товару, що включає:
 - ✓ видалення, якщо товару на даний момент немає в наявності, або взагалі не буде більше;
 - ✓ додавання, якщо вже залишилась мала кількість, або взагалі додавання чогось нового, відповідно від моніторингу замовлень і тд.
- опрацювання відгуків для аналізу своєї роботи, можливо треба щось вдосконалити, або навпаки переконатись в тому, що все виходить добре і можна сподіватись на премію.

3. Директор

- також ознайомлюється з відгуками, для моніторингу роботи магазину, а саме працівників, які опрацьовують замовлення. Таким чином ще можна знати які книги подобаються клієнтам.
- нараховує зарплату працівникам, додаючи премію, якщо ними були виконанні всі умови.
- аналізує роботу магазину, дивиться, які жанри цікавлять клієнтів, видавництва, автори і тд. Також переглядає на які

суми було куплено товару за місяць, чи яка кількість книг була продана в тому чи іншому магазині.

- формує знижки для клієнтів, наприклад якщо за місяць було зроблено замовлення на ту чи іншу суму, то клієнт отримує знижку на наступне замовлення. Це дозволить зберегти та базу клієнтів та збільшити виручки в місяць.

РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ

2.1 Поняття реляційних баз даних

Реляційна база даних — це тип бази даних, яка зберігає та надає доступ до точок даних, пов'язаних одна з одною. У реляційній базі даних кожен рядок таблиці є записом з унікальним ідентифікатором, який називається ключем. Стовпці таблиці містять атрибути даних, і кожен запис зазвичай має значення для кожного атрибута, що полегшує встановлення зв'язків між точками даних.

Реляційна модель означає, що логічні структури даних — таблиці даних, представлення даних та індекси — відокремлені від фізичних структур зберігання. Це поділ означає, що адміністратори баз даних можуть керувати фізичним зберіганням даних, не впливаючи на доступ до цих даних як логічної структури. Наприклад, перейменування файлу бази даних не призводить до перейменування таблиць, що зберігаються в ньому.

Щоб дані завжди були точними та доступними, реляційні бази даних дотримуються певних правил цілісності. Наприклад, правило цілісності може вказувати, що повторювані рядки не допускаються в таблиці, щоб виключити можливість введення помилкової інформації в базу даних.

У перші роки існування баз даних кожна програма зберігала дані у власній унікальній структурі. Коли розробники хотіли створювати програми для використання цих даних, вони повинні були знати багато про конкретну структуру даних, щоб знайти потрібні дані. Ці структури даних були неефективними, їх важко підтримувати та оптимізувати для забезпечення хорошої продуктивності програми. Модель реляційної бази даних була розроблена для вирішення проблеми безлічі довільних структур даних.

2.2 Опис моделі

Для написання даної курсової роботи, мною було використано саме реляційну базу даних, а сама через програму «MySQL Workbench». Вона забезпечує моделювання даних, розробку SQL та комплексні інструменти адміністрування для конфігурації сервера, адміністрування користувачів, резервного копіювання та багато іншого.

Діаграма концептуальної моделі даних для функціонування магазину книг представлена на рисунку:

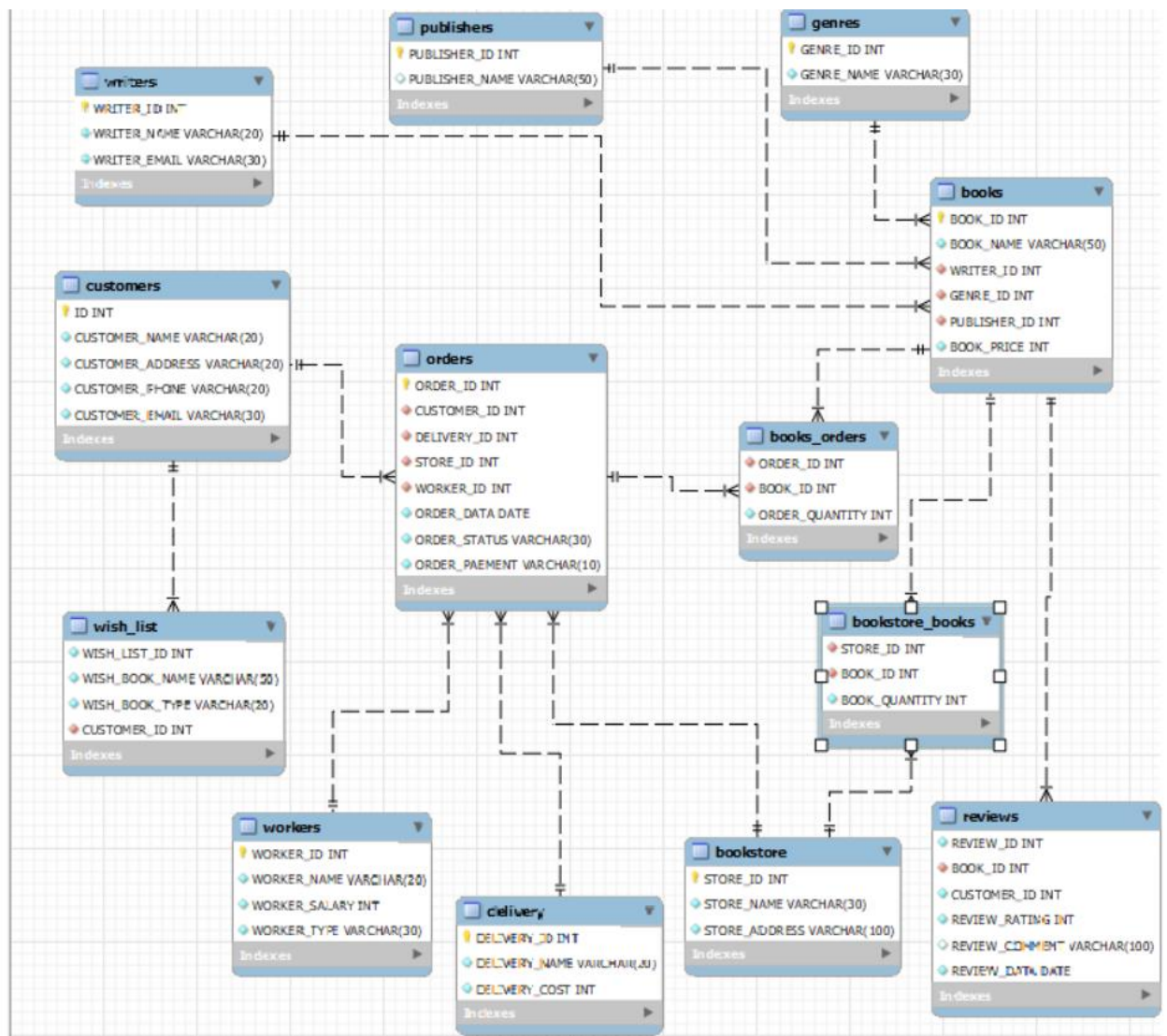


Рис 2.2.1 Концептуальна модель бази даних

База даних для функціонування магазину книг складається з таких таблиць та полів в них:

1. *Books* – містить усю інформацію про книги, що є в асортименті магазину;
 - BOOK_ID – ідифікаційний код книги;
 - BOOK_NAME – назва книги;
 - WRITER_ID – ідифікаційний код автора;
 - GENRE_ID – ідифікаційний код жанру;
 - PUBLISHER_ID – ідифікаційний код видавництва;
 - BOOK_PRICE - ціна книги;
2. *Writers* – інформація та перелік авторів;
 - WRITER_ID – ідифікаційний код автора;
 - WRITER_NAME – ім'я автора;
 - WRITER_EMAIL – email автора;
3. *Publishers* – інформація та перелік видавництв книг;
 - PUBLISHER_ID – ідифікаційний код видавництва;
 - PUBLISHER_NAME – назва видавництва;
4. *Genres* – інформація та перелік жанрів книг;
 - GENRE_ID – ідифікаційний код видавництва;
 - GENRE_NAME – вид жанру;
5. *Bookstore* – інформація та перелік книгарень, де можна придбати книги;
 - STORE_ID – ідифікаційний код книгарні;

- STORE_NAME – назва книгарні;

- STORE_ADRESS – адреса книгарні;

6. *Bookstore_books* – інформація про асортимент книгарень та кількість книг;

- STORE_ID – ідифікаційний код книгарні;
- BOOK_ID – ідифікаційний код книги;
- BOOK_QUANTITY – кількість замовлених книг;

7. *Customers* – інформація про клієнтів магазину;

- ID – ідифікаційний код клієнта;
- CUSTOMER_NAME – ім'я клієнта;
- CUSTOMER_ADRESS – адреса клієнта;
- CUSTOMER_PHONE - номер телефону клієнта;
- CUSTOMER_EMAIL – email клієнта;

8. *Wish_list* – інформація про список бажаного кожного клієнта;

- WISH_LIST_ID – ідифікаційний код списку бажаного;
- WISH_BOOK_NAME – назва книжки;
- WISH_BOOK_TYPE – жанр книжки;

9. *Orders* – інформація про кожне замовлення;

- ORDER_ID – ідифікаційний код замовлення;
- CUSTOMER_ID - ідифікаційний код клієнта;
- DELIVERY_ID - ідифікаційний код способу доставки;
- STORE_ID – ідифікаційний код книгарні;

- `WORKER_ID` – ідифікаційний код автора;
- `ORDER_DATA` – дата замовлення;
- `ORDER_STATUS` - статус замовлення;

10. *Delivery* – інформація про способи доставки, які доступні в магазині;

- `DELIVERY_ID` - ідифікаційний код способу доставки;
- `DELIVERY_NAME` - назва способу доставки;
- `DELIVERY_COST` – вартість доставки;

11. *Workers* - інформація та список працівників кожної книгарні;

- `WORKER_ID` – ідифікаційний код працівника;
- `WORKER_NAME` - ім'я працівника;
- `WORKER_SALARY` – зарплата працівника;
- `WORKER_TYPE` - тип працівника;

12. *Books-orders* - інформація про кількість замовлених книг за одне замовлення.

- `BOOK_ID` – ідифікаційний код книги;
- `ORDER_ID` – ідифікаційний код замовлення;
- `ORDER_QUANTITY` – кількість замовлених книг.

2.3 Нормалізація відношень

Нормалізація — це процес організації даних у базі даних. Це включає створення таблиць і встановлення зв'язків між цими таблицями відповідно до

правил, розроблених як для захисту даних, так і для забезпечення більшої гнучкості бази даних, усуваючи надлишковість і непотійну залежність.

Існує кілька правил нормалізації бази даних. Кожне правило називається «звичайною формою». Якщо дотримується перше правило, кажуть, що база даних знаходиться в «першій нормальній формі». Якщо дотримуються перші три правила, база даних вважається «третьою нормальній формі». Хоча можливі й інші рівні нормалізації, третя нормальна форма вважається найвищим рівнем, необхідним для більшості застосувань.

Принципи нормалізації

- в кожній таблиці БД не повинно бути повторюваних полів;
- в кожній таблиці повинен бути унікальний ідентифікатор (первинний ключ);
- кожному значенню первинного ключа повинна відповідати достатня інформація про тип суті або про об'єкт таблиці (наприклад, інформація про успішність, про групу або студентах);
- зміна значень в полях таблиці не повинна впливати на інформацію в інших полях (крім змін у полях ключа).

Перша нормальна форма (1NF)

Щоб таблиця була в першій нормальній формі, вона повинна відповідати наступним 4 правилам:

- таблиця повинна мати лише одинарні (атомарні) атрибути/стовпці;
- значення, що зберігаються в стовпці, мають бути в одному домені;
- усі стовпці в таблиці повинні мати унікальні імена;
- порядок, в якому зберігаються дані, не має значення;

Друга нормальна форма (2NF)

Щоб таблиця була у другій нормальній формі:

- має бути у першій нормальній формі;
- не повинна мати часткової залежності;

Третя нормальна форма (3NF)

Таблиця знаходиться в третій нормальній формі, коли:

- знаходиться у другій нормальній формі;
- не має транзитивної залежності;

Нормальна форма Бойса і Кодда

Нормальна форма Бойса і Кодда є вищою версією третьої нормальної форми. Ця форма стосується певного типу аномалії, яка не обробляється 3NF. Кажуть, що таблиця 3NF, яка не має кількох перекриваються ключів-кандидатів, знаходиться в BCNF. Щоб таблиця була в BCNF, мають бути виконані наступні умови:

- таблиця має бути в 3-й нормальній формі;
- і для кожної функціональної залежності $(X \rightarrow Y)$ X має бути суперключем;

Четверта нормальна форма (4NF)

Таблиця знаходиться в четвертій нормальній формі, коли:

- знаходиться в нормальній формі Бойса-Кодда.
- не має багатозначної залежності.

2.4 Визначення типів даних

Фактичні одиниці даних, які вводяться в базу даних, дають атрибути для кожної сутності. Ці одиниці даних також називають елементами даних. Кожне поле має свій тип даних.

Введення даних — це спосіб класифікації значень даних, які мають спільні властивості. Різні типи значень даних також потребують різного об'єму пам'яті для їх зберігання та мають різні операції, які можна виконувати з ними.

У MySQL є три основні типи даних:

- рядкові;
- числові;
- дата і час.

Типи даних, які було використано для виконання завдання:

- INT - числове значення з максимальною кількістю цифр стовпця, зазначеною в дужках, діапазон від -2147483648 до 2147483647;
- VARCHAR - рядок символів змінної довжини. Максимальний розмір вказується в дужках. Найбільшим може бути 255;
- DATA - дата у форматі: YYYY-MM-DD;
- BOOL - нуль вважається хибним, ненульові значення вважаються істинними;

У таблиці `writers` містяться такі поля з такими типами даних та розмірами (рис 2.4.1):

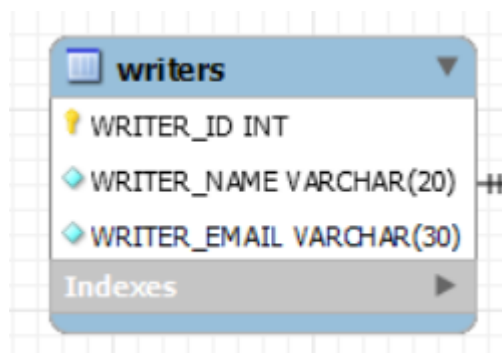


Рис 2.4.1. Типи даних атрибутів таблиці `writers`

У таблиці `publishers` містяться такі поля з такими типами даних та розмірами (рис. 2.4.2):

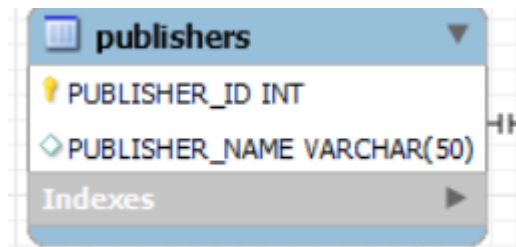


Рис 2.4.2. Типи даних атрибутів таблиці publishers

У таблиці genres міститься такі поля з такими типами даних та розмірами (рис. 2.4.3):

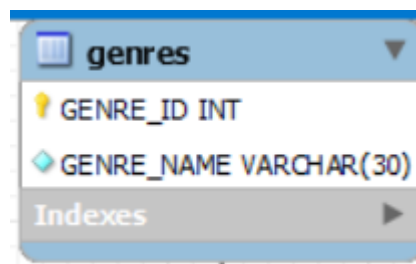


Рис 2.4.3. Типи даних атрибутів таблиці genres

У таблиці books міститься такі поля з такими типами даних та розмірами (рис. 2.4.4):

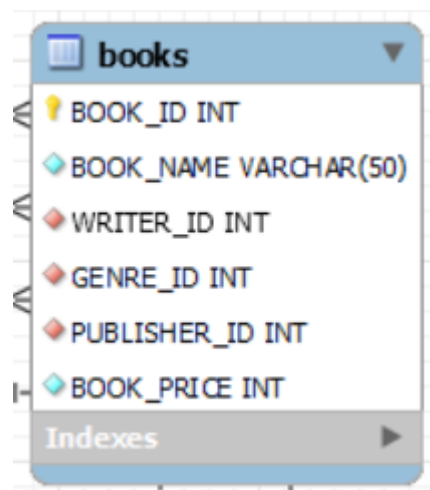


Рис 2.4.4. Типи даних атрибутів таблиці books

У таблиці books_orders міститься такі поля з такими типами даних та розмірами (рис. 2.4.5):

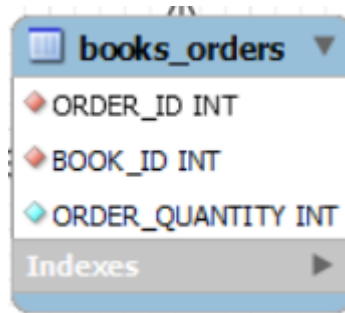


Рис 2.4.5. Типи даних атрибутів таблиці books

У таблиці bookstore_book містяться такі поля з такими типами даних та розмірами (рис. 2.4.6):

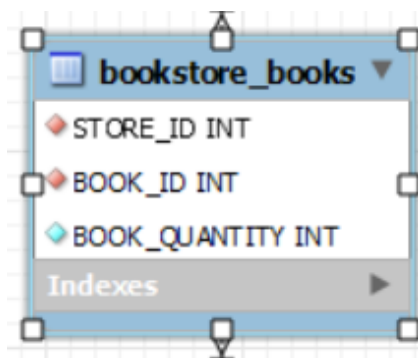


Рис 2.4.6. Типи даних атрибутів таблиці bookstore_books

У таблиці reviews містяться такі поля з такими типами даних та розмірами (рис. 2.4.7):

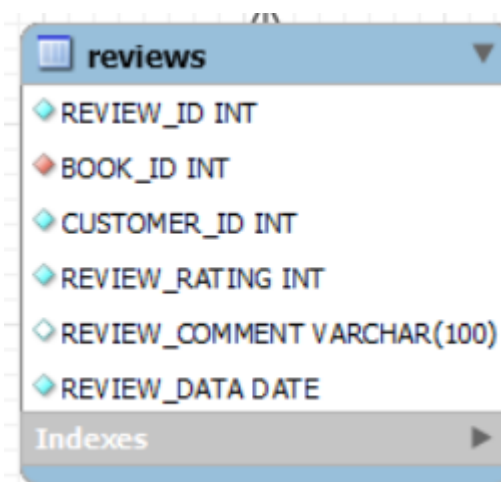


Рис 2.4.7. Типи даних атрибутів таблиці reviews

У таблиці customers містяться такі поля з такими типами даних та розмірами (рис. 2.4.8):

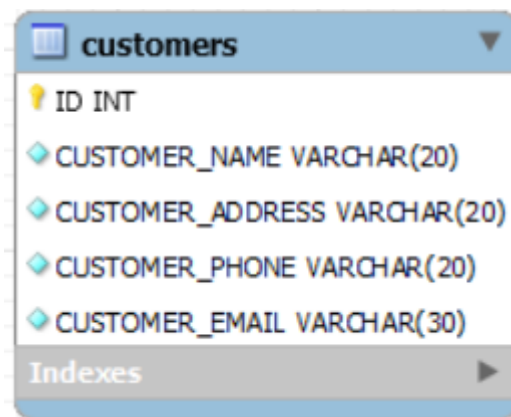


Рис 2.4.8. Типи даних атрибутів таблиці customers

У таблиці wish_list містяться такі поля з такими типами даних та розмірами (рис. 2.4.9):

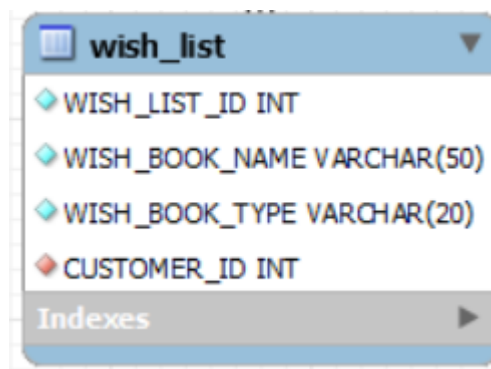


Рис 2.4.9. Типи даних атрибутів таблиці wish_list

У таблиці workers містяться такі поля з такими типами даних та розмірами (рис. 2.4.10):

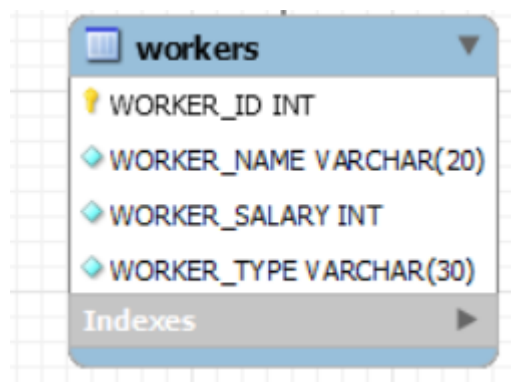


Рис 2.4.10. Типи даних атрибутів таблиці workers

У таблиці delivery міститься такі поля з такими типами даних та розмірами (рис. 2.4.11):

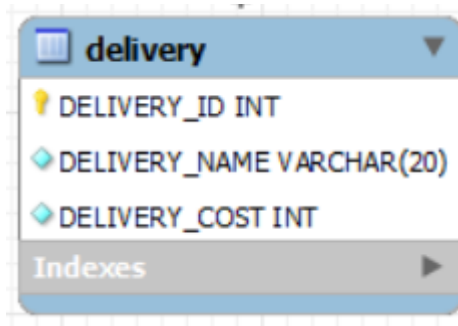


Рис 2.4.11. Типи даних атрибутів таблиці delivery

У таблиці bookstore міститься такі поля з такими типами даних та розмірами (рис. 2.4.12):

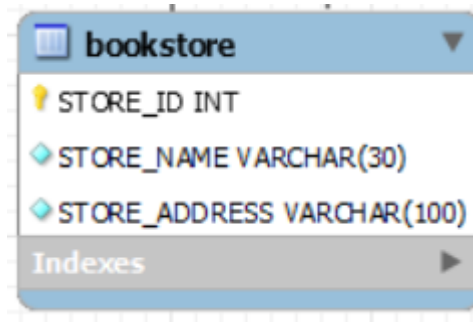


Рис 2.4.12. Типи даних атрибутів таблиці bookstore

У таблиці orders міститься такі поля з такими типами даних та розмірами (рис. 2.4.13):

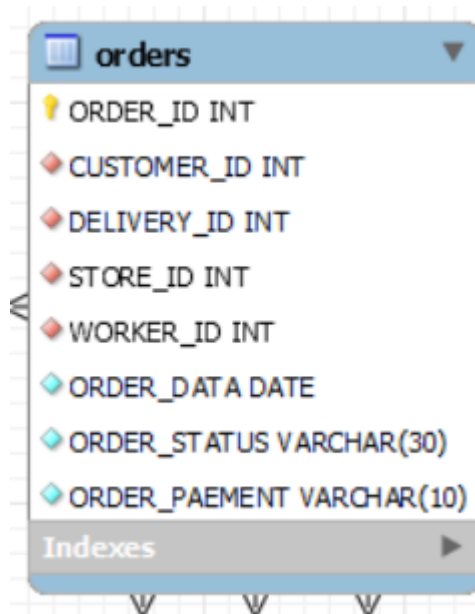


Рис 2.4.13. Типи даних атрибутів таблиці orders

2.5 Обмеження цілісності даних

Цілісність даних є фундаментальним компонентом інформаційної безпеки. У своєму найширшому застосуванні «цілісність даних» відноситься до точності та узгодженості даних, що зберігаються в базі даних.

Важливість збереження цілісності даних очевидна при створенні зв'язків між різними елементами даних. Це гарантує, що дані, що передаються з одного етапу на інший, є точним і безпомилковим.

Можна виділити два обмеження, що стосуються цілісності даних та первинного та зовнішнього ключів:

1. Цілісність сутності гарантує, що первинний ключ у таблиці є унікальним, а значення не має значення null.
2. Посилальна цілісність вимагає, щоб кожне значення в стовпці зовнішнього ключа було знайдено в первинному ключі таблиці, з якої воно походить.

Найпоширенішими типами цілісності даних в sql є:

- NOT NULL/NULL — заборона/ дозвіл на використання не заданих або не визначених значень;
- UNIQUE — контроль унікальності значень атрибутів;
- CHECK — контроль допустимих значень атрибутів;
- PRIMARY KEY — первинний ключ;
- FOREIGN KEY — зовнішній ключ.

Первинний і зовнішній ключі визначають реляційну структуру бази даних. Ці ключі дозволяють ідентифікувати кожен рядок у таблицях бази даних і визначають зв'язки між таблицями.

Усі таблиці в реляційній базі даних повинні мати первинний ключ. Первинний ключ — це стовець або набір стовпців, який дозволяє однозначно ідентифікувати кожен рядок таблиці. Жодні два рядки в таблиці з первинним ключем не можуть мати однакове значення первинного ключа.

Якщо первинний ключ не призначено, усі стовпці разом стають первинним ключем. Рекомендується зберігати первинний ключ для кожної таблиці якомога компактнішим. Інформація в одній таблиці пов'язана з інформацією в інших таблицях зовнішніми ключами.

При розробці бази даних для функціонування магазину книг було використано: PRIMARY KEY, FOREIGN KEY, NOT NULL.

Розглянемо використання типу цілісності за допомогою первинного та вторинного ключа (рис.2.5.1, рис.2.5.2)

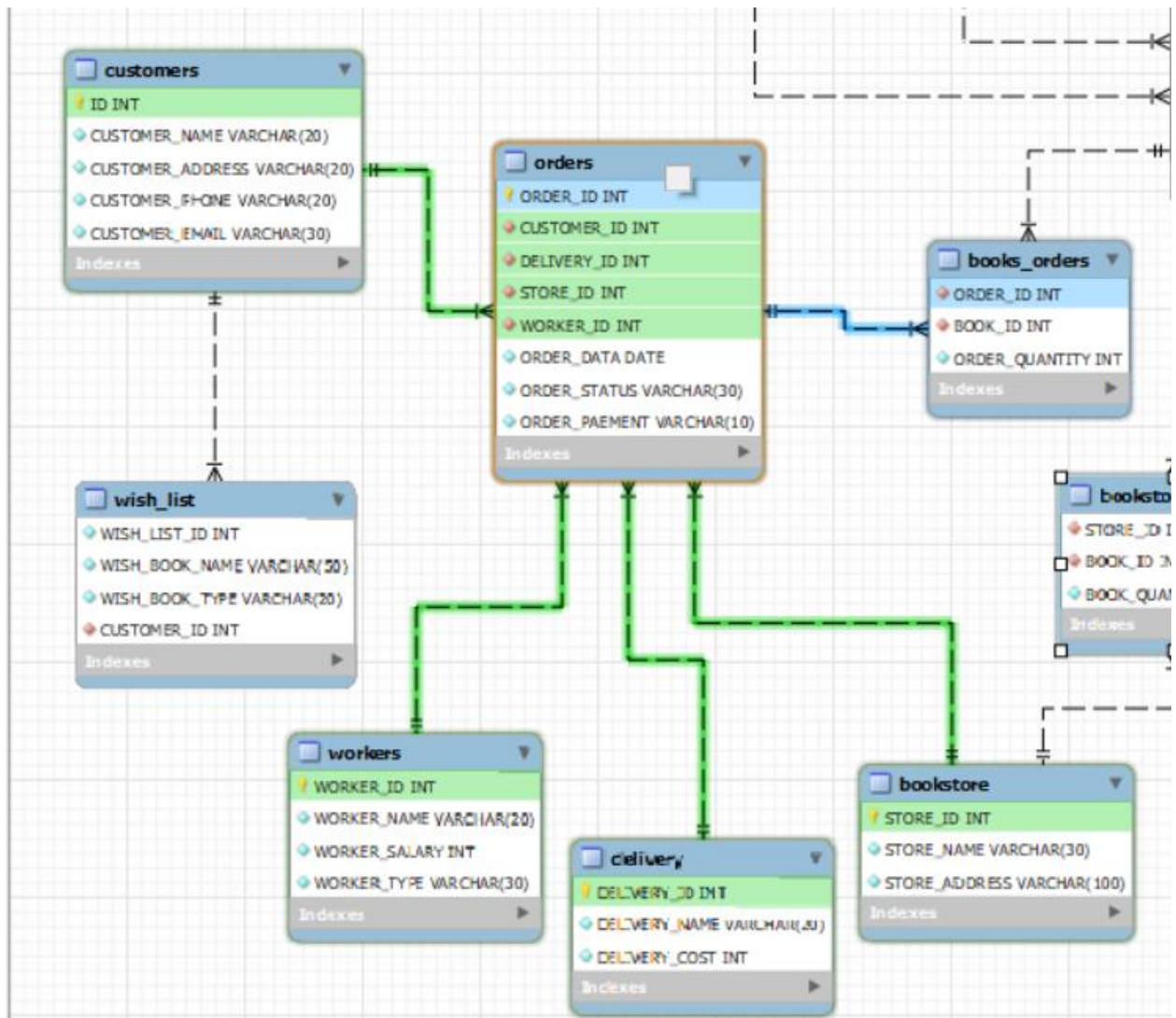


Рис.2.5.1 Діаграма відносин зв'язків між таблицею orders та іншими створена за допомогою PRIMARY KEY і FOREIGN KEY

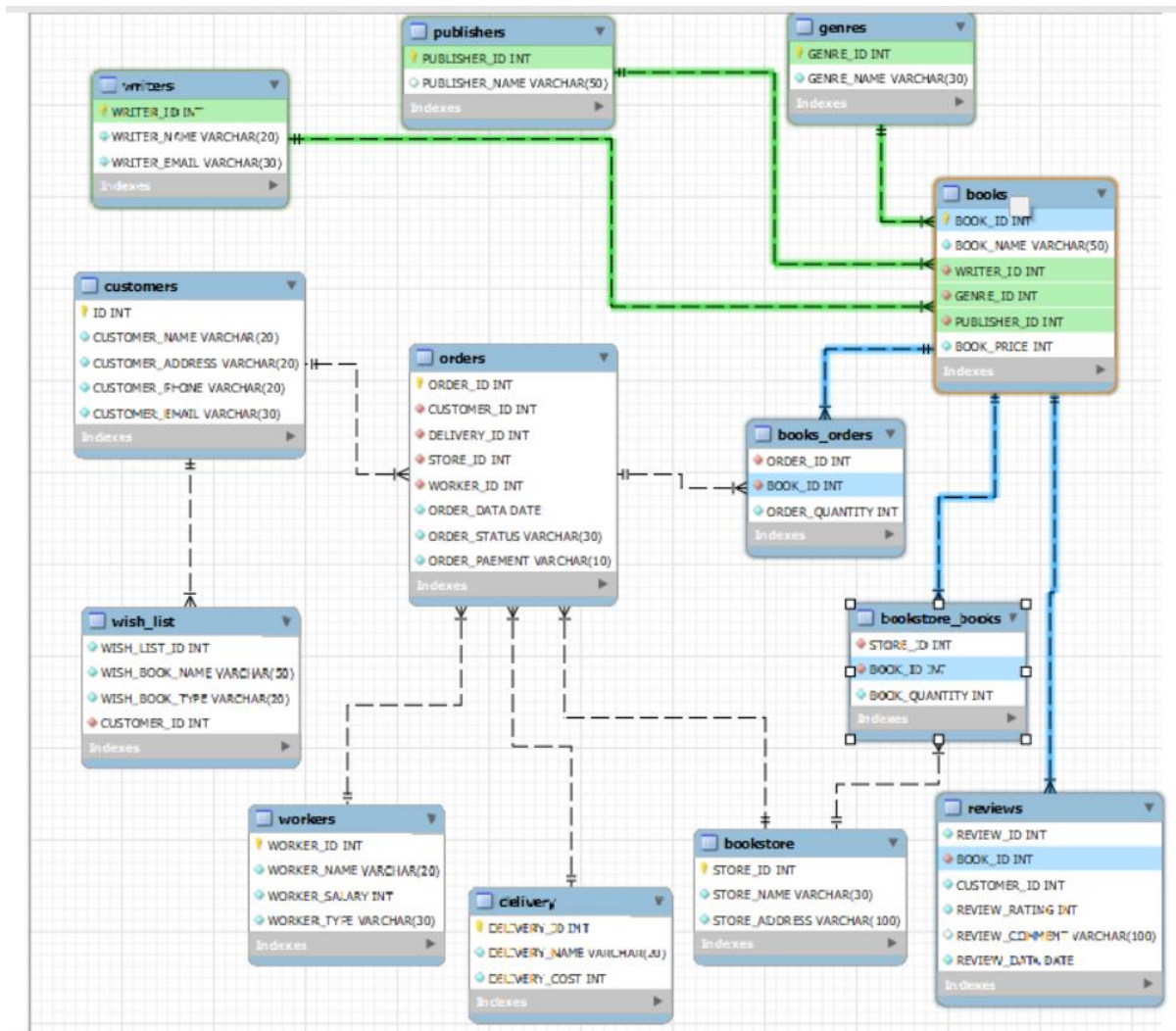


Рис.2.5.2 Діаграма відносин зв'язків між таблицею books та іншими створена за допомогою PRIMARY KEY і FOREIGN KEY

2.6 Реалізація SQL-скрипту

В базі даних було реалізовано запити, що покривають потреби діаграми прецедентів. Розглянемо декілька і з них, а решта відображені в Додатку

1. Сортуння усіх книг, автором яких є Галина Вдовиченко. Відповідно автора можна змінювати. Цей запит допоможе клієнту, швидше знайти книги, якщо він знає лише автора. Результат виведено на рис.2.6.1

```

SELECT BOOKS.BOOK_ID, books.BOOK_NAME,
writers.WRITER_NAME
FROM BOOKS inner join

```

WRITERS on WRITERS.writer_ID = BOOKS.writer_ID
 where writers.WRITER_NAME like 'Галина Вдовиченко';

	BOOK_ID	BOOK_NAME	WRITER_NAME
▶	2	Найважливіше-наприкінці	Галина Вдовиченко
	12	Земля	Галина Вдовиченко
	20	Політ в глибінь	Галина Вдовиченко

Рис.2.6.1 Результат першого запиту

- Запит, реалізований через функцію, який виводить, чи може клієнт замовити книгу в певній кількості в магазині, чи її немає в наявності.
 USE BOOKSTORE;
 SET GLOBAL log_bin_trust_function_creators = 1;

DELIMITER //

```
CREATE FUNCTION ChechBookWhichCanBeBooked ( bookId INT,
wantToBuy INT )
RETURNS VARCHAR(100)
```

BEGIN

```
    DECLARE booked_quantity INT;
    DECLARE can_booked INT;
    DECLARE bookexist bool;
    DECLARE bookName varchar(100);
```

```
    SET bookexist = (SELECT exists (
SELECT
BOOKS.BOOK_ID FROM books WHERE BOOK_ID = bookId ));
```

```
    IF (bookexist<1) THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'книги з таким
кодом немає';
    END IF;
```

```
    SET booked_quantity = (SELECT
COALESCE(SUM(ORDER_QUANTITY),0)
FROM books_orders
```

```
where BOOK_ID = bookId  
group by BOOK_ID);
```

```
set can_booked = (SELECT COALESCE(SUM(BOOK_QUANTITY),0)  
FROM bookstore_books  
where BOOK_ID = bookId  
group by BOOK_ID);
```

```
IF (can_booked - booked_quantity < wantToBuy) THEN  
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Немає в  
наявності!';  
END IF;  
set bookName = ( SELECT BOOK_NAME from BOOKS where  
BOOK_ID = bookId);
```

```
RETURN (SELECT CONCAT('УРА! можна придбати:', bookName ));
```

```
END; //
```

```
DELIMITER ;
```

```
SELECT ChechBookWhichCanBeBooked(17,10);
```

Якщо така книга є в наявності і її можна придбати, то виводить ось такий результат:

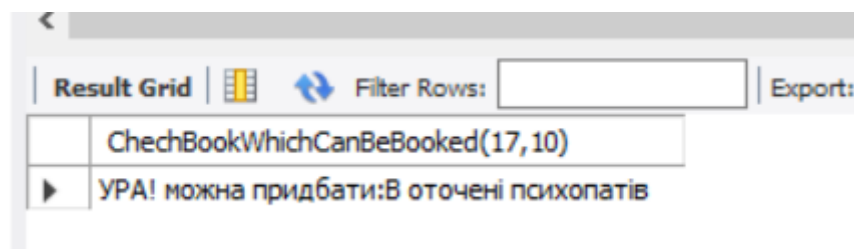


Рис.2.6.2 Результат другого запиту

Якщо книги немає, то відповідно виводить наступний результат

Error Code: 1644. Немає в наявності!

Рис.2.6.3 Результат другого запиту

3. Запит, що відображає усі замовлення статусу «нове», щоб менеджер міг брати його до опрацювання. Результат виведений на рисунку 2.6.4

```
SELECT o.ORDER_ID, customers.CUSTOMER_NAME,  
customers.CUSTOMER_EMAIL,customers.CUSTOMER_ADDRESS,cust  
omers.CUSTOMER_PHONE ,books. book_name, od.order_quantity,  
o.order_data, o.ORDER_STATUS  
FROM ORDERS as o LEFT JOIN BOOKS_ORDERS as od on  
o.ORDER_ID = od.ORDER_ID  
left join Customers on o.customer_id = customers.id  
left join books on od.book_id = books.book_id  
where o.ORDER_STATUS like 'нове';
```

ORDER_ID	CUSTOMER_NAME	CUSTOMER_EMAIL	CUSTOMER_ADDRESS	CUSTOMER_PHONE	book_name	order_quantity	order_data	ORDER_STATUS
21	Хадай Андріана	Andriana0987@gmail.com	Чернівці	0982514763	Поліпсети	1	2022-05-16	нове
22	Кріль Каріна	KrilKarina09@gmail.com	Тернопіль	0962581473	І знову я влізаю в танк	3	2022-05-17	нове
23	Босак Василь	Bossaakk09@gmail.com	Київ	0965252148	Крила	1	2022-05-20	нове

Рис.2.6.4 Результат третього запиту

4. Запит, що дозволяє менеджеру редагувати асортимент товару, вивчивши, які книги, якого жанру купляють найчастіше, відсортовано по порядку спадання Результат виведений на рисунку 2.6.5



Result Grid  Filter Rows: Export:  Wrap Cell Content: <u>1</u> A				
	ORDER_ID	book_name	GENRE_NAME	max
▶	7	Мати або бути	психологія	22
	1	А що Бог їсть на сніданок?	дитяча література	20
	8	Музей покинутих предків	роман	8
	8	І знову я влізаю в танк	класика	8
	2	Згодом	детектив	7
	4	Століття Якова	роман	6
	5	Крила	психологія	6
	9	Ніби мене нема	класика	6
	1	Найважливіше-наприкінці	роман	5
	11	Дитячий сміх	дитяча література	5
	9	Дотик неба	поезія	3
	17	Щоденник 3	комікс	3
	6	Сила думки	психологія	2
	10	Політ в глибіню	психологія	2
	10	Вічна Любов	комікс	2
	18	Полімпсети	поезія	2
	19	В оточені психопатів	психологія	2
	19	Чуєш,коли прийдеш додо...	класика	1

Рис. 2.6.5 Результат четвертого запиту

ВИСНОВКИ

Під час виконання курсової роботи проаналізовано та вивчено використання MySQL, також розроблено базу даних для функціоналу магазину книг, що дозволить менеджерам значно швидше виконувати свою роботу, а клієнту оформляти замовлення і тд.

Найперше було проаналізовано ринок, а саме конкурентів, через SWOT-аналіз та 4P, відповідно до того було прописано вимоги для майбутнього магазину книг.

Наступним кроком було побудовано Use case діаграму, яка складається з трьох акторів. Дана діаграма дозволяє краще зрозуміти майбутню структуру БД чи веб-сторінки.

Перш ніж створювати базу даних, було побудовано ER діаграму, що показує взаємодію між відношеннями і дозволяє приступити до програмування бази даних. Це дозволяє користувачу краще зрозуміти структуру та функціонал БД. Опісля вже було наповнено саму базу даних, полями, визначено для них тип даних та цілість. Продумано та спроектовано зв'язки між полями. Розроблено запити та функції для зручності використання .

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. SQL ANYWHERE. URL: <https://www.sqlanywhere.info/EN/>;
2. Description of the database normalization basics. URL: <https://docs.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description>;
3. Основні поняття реляційних БД: нормалізація, зв'язок та ключі. URL: <https://bondarenko.dn.ua/osnovni-ponyattya-relyatsijnih-bd-normalizatsiya-zv-yazok-ta-klyuchi/>;
4. Стеняшин А. Ю. Про формальний опис типів і структур даних різнорідних програм. *Проблеми програмування*. № 2. 2011. С. 50-61;
5. What Is A Database? URL: <https://phoenixnap.com/kb/what-is-a-database>;

Лістинг А.1 Створення таблиць

```
CREATE DATABASE BOOKSTORE;
USE BOOKSTORE;

CREATE TABLE WRITERS (
    WRITER_ID INT NOT NULL ,
    WRITER_NAME VARCHAR(20) NOT NULL,
    WRITER_EMAIL VARCHAR(30) NOT NULL
);

CREATE TABLE GENRES (
    GENRE_ID INT NOT NULL,
    GENRE_NAME VARCHAR(30) NOT NULL
);

CREATE TABLE PUBLISHERS (
    PUBLISHER_ID INT NOT NULL,
    PUBLISHER_NAME VARCHAR(50)
);

CREATE TABLE BOOKS (
    BOOK_ID INT NOT NULL ,
    BOOK_NAME VARCHAR(50) NOT NULL,
    WRITER_ID INT NOT NULL,
    GENRE_ID INT NOT NULL,
    PUBLISHER_ID INT NOT NULL,
    BOOK_PRICE INT NOT NULL
);

CREATE TABLE BOOKSTORE (
    STORE_ID INT NOT NULL ,
    STORE_NAME VARCHAR(30) NOT NULL,
    STORE_ADDRESS VARCHAR(100) NOT NULL
);

CREATE TABLE BOOKSTORE_BOOKS (
    STORE_ID INT NOT NULL,
```

```

        BOOK_ID INT NOT NULL,
        BOOK_QUANTITY INT NOT NULL
    );

CREATE TABLE DELIVERY(
    DELIVERY_ID INT NOT NULL ,
    DELIVERY_NAME VARCHAR(20) NOT NULL,
    DELIVERY_COST INT NOT NULL
);

CREATE TABLE WORKERS(
    WORKER_ID INT NOT NULL,
    WORKER_NAME VARCHAR(20) NOT NULL,
    WORKER_SALARY INT NOT NULL,
    WORKER_TYPE VARCHAR(30) NOT NULL
);

CREATE TABLE CUSTOMERS(
    ID INT NOT NULL,
    CUSTOMER_NAME VARCHAR(20) NOT NULL,
    CUSTOMER_ADDRESS VARCHAR(20) NOT NULL,
    CUSTOMER_PHONE VARCHAR(20) NOT NULL,
    CUSTOMER_EMAIL VARCHAR(30) NOT NULL
);

CREATE TABLE ORDERS(
    ORDER_ID INT NOT NULL,
    CUSTOMER_ID INT NOT NULL,
    DELIVERY_ID INT NOT NULL,
    STORE_ID INT NOT NULL,
    WORKER_ID INT NOT NULL,
    ORDER_DATA date NOT NULL,
    ORDER_STATUS VARCHAR(30) NOT NULL,
    ORDER_PAEMENT VARCHAR(10) NOT NULL
);

CREATE TABLE BOOKS_ORDERS(
    ORDER_ID INT NOT NULL,
    BOOK_ID INT NOT NULL,
    ORDER_QUANTITY INT NOT NULL
);

```

```

CREATE TABLE WISH_LIST(
    WISH_LIST_ID INT NOT NULL,
    WISH_BOOK_NAME VARCHAR(50) NOT NULL,
    WISH_BOOK_TYPE VARCHAR(20) NOT NULL,
    CUSTOMER_ID INT NOT NULL
);

CREATE TABLE REVIEWS (
    REVIEW_ID INT NOT NULL,
    BOOK_ID INT NOT NULL,
    CUSTOMER_ID INT NOT NULL,
    REVIEW_RATING INT NOT NULL,
    REVIEW_COMMENT VARCHAR (100),
    REVIEW_DATA date not null
);

```

Лістинг А.2 Створення зв'язків між таблицями

```

ALTER TABLE WRITERS ADD CONSTRAINT WRITER_ID PRIMARY KEY(WRITER_ID);

ALTER TABLE BOOKS ADD FOREIGN KEY (WRITER_ID) REFERENCES
WRITERS(WRITER_ID);

ALTER TABLE DELIVERY ADD CONSTRAINT DELIVERY_ID PRIMARY KEY(DELIVERY_ID);

ALTER TABLE ORDERS ADD FOREIGN KEY (DELIVERY_ID) REFERENCES
DELIVERY(DELIVERY_ID);

ALTER TABLE BOOKSTORE ADD CONSTRAINT STORE_ID PRIMARY KEY(STORE_ID);

ALTER TABLE ORDERS ADD FOREIGN KEY (STORE_ID) REFERENCES
BOOKSTORE(STORE_ID);

ALTER TABLE PUBLISHERS ADD CONSTRAINT PUBLISHER_ID PRIMARY
KEY(PUBLISHER_ID);

ALTER TABLE BOOKS ADD FOREIGN KEY (PUBLISHER_ID) REFERENCES
PUBLISHERS(PUBLISHER_ID);

ALTER TABLE GENRES ADD CONSTRAINT GENRE_ID PRIMARY KEY(GENRE_ID);

ALTER TABLE BOOKS ADD FOREIGN KEY (GENRE_ID) REFERENCES GENRES(GENRE_ID);

ALTER TABLE BOOKS ADD CONSTRAINT BOOK_ID PRIMARY KEY(BOOK_ID);

```

```

ALTER TABLE BOOKS_ORDERS ADD FOREIGN KEY (BOOK_ID) REFERENCES
BOOKS (BOOK_ID);

ALTER TABLE BOOKSTORE_BOOKS ADD FOREIGN KEY (BOOK_ID) REFERENCES
BOOKS (BOOK_ID);

ALTER TABLE REVIEWS ADD FOREIGN KEY (BOOK_ID) REFERENCES books (BOOK_ID);

ALTER TABLE ORDERS ADD CONSTRAINT ORDER_ID PRIMARY KEY (ORDER_ID);

ALTER TABLE BOOKS_ORDERS ADD FOREIGN KEY (ORDER_ID) REFERENCES
ORDERS (ORDER_ID);

ALTER TABLE BOOKSTORE_BOOKS ADD FOREIGN KEY (STORE_ID) REFERENCES
BOOKSTORE (STORE_ID);

ALTER TABLE CUSTOMERS ADD CONSTRAINT ID PRIMARY KEY (ID);

ALTER TABLE WISH_LIST ADD FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMERS
(ID);

ALTER TABLE ORDERS ADD FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMERS (ID);

ALTER TABLE WORKERS ADD CONSTRAINT WORKER_ID PRIMARY KEY (WORKER_ID);

ALTER TABLE ORDERS ADD FOREIGN KEY (WORKER_ID) REFERENCES
WORKERS (WORKER_ID);

```

Лістинг А.3 Заповнення таблиці Writers

```

INSERT INTO WRITERS (WRITER_ID, WRITER_NAME, WRITER_EMAIL) values (1,
'Галина Вдовиченко', 'HALUNA@gmail.com');

INSERT INTO WRITERS (WRITER_ID, WRITER_NAME, WRITER_EMAIL) values (2,
'Оксана Забужко', 'OZabu@gmail.com');

INSERT INTO WRITERS (WRITER_ID, WRITER_NAME, WRITER_EMAIL) values (3, 'Юлія
Сливка', 'Slyvka12@gmail.com');

INSERT INTO WRITERS (WRITER_ID, WRITER_NAME, WRITER_EMAIL) values (4,
'Василь Шкляр', 'Vasul938393@gmail.com');

INSERT INTO WRITERS (WRITER_ID, WRITER_NAME, WRITER_EMAIL) values (5,
'Василь Стус', '165STYS@gmail.com');

```

```

INSERT INTO WRITERS(WRITER_ID, WRITER_NAME, WRITER_EMAIL) values (6,
'Sтiвен Кiнг', 'kingStiven@gmail.com');

INSERT INTO WRITERS(WRITER_ID, WRITER_NAME, WRITER_EMAIL) values (7, 'Люко
Дашвар', 'LDaschvar@gmail.com');

INSERT INTO WRITERS(WRITER_ID, WRITER_NAME, WRITER_EMAIL) values (8,
'Володимир Лис', 'VolodL232@gmail.com');

INSERT INTO WRITERS(WRITER_ID, WRITER_NAME, WRITER_EMAIL) values (9,
'Томас Епiксон', 'TOM323453@gmail.com');

INSERT INTO WRITERS(WRITER_ID, WRITER_NAME, WRITER_EMAIL) values (10,
'Гравiтi Фолс', 'Fols43884Hr@gmail.com');

```

Лістинг А.3 Заповнення таблиці GENRES

```

INSERT INTO GENRES(GENRE_ID, GENRE_NAME) values (1, 'класика');

INSERT INTO GENRES(GENRE_ID, GENRE_NAME) values (2, 'психологія');

INSERT INTO GENRES(GENRE_ID, GENRE_NAME) values (3, 'роман');

INSERT INTO GENRES(GENRE_ID, GENRE_NAME) values (4, 'комікс');

INSERT INTO GENRES(GENRE_ID, GENRE_NAME) values (5, 'детектив');

INSERT INTO GENRES(GENRE_ID, GENRE_NAME) values (6, 'поезія');

INSERT INTO GENRES(GENRE_ID, GENRE_NAME) values (7, 'дитяча література');

```

Лістинг А.4 Заповнення таблиці PUBLISHERS

```

INSERT INTO PUBLISHERS(PUBLISHER_ID, PUBLISHER_NAME) values (1,
'Видавництво Юлії Сливки');

INSERT INTO PUBLISHERS(PUBLISHER_ID, PUBLISHER_NAME) values (2,
'Видавництво Старого Лева');

INSERT INTO PUBLISHERS(PUBLISHER_ID, PUBLISHER_NAME) values (3, 'А-ба-ба-
га-ла-ма-га');

INSERT INTO PUBLISHERS(PUBLISHER_ID, PUBLISHER_NAME) values (4, 'Клуб
Сімейного Дозвілля');

INSERT INTO PUBLISHERS(PUBLISHER_ID, PUBLISHER_NAME) values (5, 'Біват');

INSERT INTO PUBLISHERS(PUBLISHER_ID, PUBLISHER_NAME) values (6, 'Човен');

```

```
INSERT INTO PUBLISHERS(PUBLISHER_ID, PUBLISHER_NAME) values (7, 'Грані-  
Т');
```

```
INSERT INTO PUBLISHERS(PUBLISHER_ID, PUBLISHER_NAME) values (8, 'Ранок');
```

```
INSERT INTO PUBLISHERS(PUBLISHER_ID, PUBLISHER_NAME) values (9, 'Фабола');
```

```
INSERT INTO PUBLISHERS(PUBLISHER_ID, PUBLISHER_NAME) values (10, 'Зоря');
```

Лістинг А.5 Заповнення таблиці BOOKS

```
INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID, GENRE_ID, PUBLISHER_ID,  
BOOK_PRICE) values (1, 'А що Бог їсть на сніданок?' , 3, 7, 1, 250);
```

```
INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID, GENRE_ID, PUBLISHER_ID,  
BOOK_PRICE) values (2, 'Найважливіше-наприкінці' , 1, 3, 4, 180);
```

```
INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID, GENRE_ID, PUBLISHER_ID,  
BOOK_PRICE) values (3, 'Згодом' , 6, 5, 5, 220);
```

```
INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID, GENRE_ID, PUBLISHER_ID,  
BOOK_PRICE) values (4, 'Полімпсети' , 5, 6, 2, 300);
```

```
INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID, GENRE_ID, PUBLISHER_ID,  
BOOK_PRICE) values (5, 'Століття Якова' , 8, 3, 9, 400);
```

```
INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID, GENRE_ID, PUBLISHER_ID,  
BOOK_PRICE) values (6, 'Ніби мене нема' , 7, 1, 10, 140);
```

```
INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID, GENRE_ID, PUBLISHER_ID,  
BOOK_PRICE) values (7, 'Чуєш, коли приїдеш додому?' , 3, 1, 1, 450);
```

```
INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID, GENRE_ID, PUBLISHER_ID,  
BOOK_PRICE) values (8, 'Музей покинутих предків' , 2, 3, 7, 156);
```

```
INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID, GENRE_ID, PUBLISHER_ID,  
BOOK_PRICE) values (9, 'Щоденник 3' , 10, 4, 8, 500);
```

```
INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID, GENRE_ID, PUBLISHER_ID,  
BOOK_PRICE) values (10, 'Крила' , 9, 2, 6, 236);
```

```
INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID, GENRE_ID, PUBLISHER_ID,  
BOOK_PRICE) values (11, 'Дотик неба' , 4, 6, 5, 420);
```

```
INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID, GENRE_ID, PUBLISHER_ID,  
BOOK_PRICE) values (12, 'Земля' , 1, 1, 3, 520);
```



```

INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID,GENRE_ID,PUBLISHER_ID,
BOOK_PRICE) values (13, 'Мати або бути' ,9,2,10,290);

INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID,GENRE_ID,PUBLISHER_ID,
BOOK_PRICE) values (14, 'І знову я влізаю в танк' ,2,1,6,259);

INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID,GENRE_ID,PUBLISHER_ID,
BOOK_PRICE) values (15, 'День, що навчив мене жити' ,7,2,1,199);

INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID,GENRE_ID,PUBLISHER_ID,
BOOK_PRICE) values (16, 'Дитячий сміх' ,2,7,3,300);

INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID,GENRE_ID,PUBLISHER_ID,
BOOK_PRICE) values (17, 'В оточені психопатів' ,9,2,7,200);

INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID,GENRE_ID,PUBLISHER_ID,
BOOK_PRICE) values (18, 'Сила думки' ,5,2,8,320);

INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID,GENRE_ID,PUBLISHER_ID,
BOOK_PRICE) values (19, 'Вічна Любов' ,6,4,3,540);

INSERT INTO BOOKS(BOOK_ID, BOOK_NAME, WRITER_ID,GENRE_ID,PUBLISHER_ID,
BOOK_PRICE) values (20, 'Політ в глибінь' ,1,2,4,210);

```

Лістинг А.5 Заповнення таблиці DELIVERY

```

INSERT INTO DELIVERY(DELIVERY_ID, DELIVERY_NAME,DELIVERY_COST) values (1,
'Нова пошта', 70);

INSERT INTO DELIVERY(DELIVERY_ID, DELIVERY_NAME,DELIVERY_COST) values (2,
'Укрпошта', 45);

INSERT INTO DELIVERY(DELIVERY_ID, DELIVERY_NAME,DELIVERY_COST) values (3,
'Самовивіз', 0);

INSERT INTO DELIVERY(DELIVERY_ID, DELIVERY_NAME,DELIVERY_COST) values (4,
'Кур`єрська доставка', 120);

INSERT INTO DELIVERY(DELIVERY_ID, DELIVERY_NAME,DELIVERY_COST) values (5,
'MEEST', 50);

INSERT INTO DELIVERY(DELIVERY_ID, DELIVERY_NAME,DELIVERY_COST) values (6,
'Yustin', 25);

```

Лістинг А.6 Заповнення таблиці WORKERS

```
INSERT INTO WORKERS(WORKER_ID, WORKER_NAME, WORKER_SALARY,WORKER_TYPE)
values (1, ' Андрій Мороз', 12080, 'менеджер');

INSERT INTO WORKERS(WORKER_ID, WORKER_NAME, WORKER_SALARY,WORKER_TYPE)
values (2, ' Олег Семків', 12040, 'менеджер');

INSERT INTO WORKERS(WORKER_ID, WORKER_NAME, WORKER_SALARY,WORKER_TYPE)
values (3, ' Ілля Пукач', 10000, 'менеджер');

INSERT INTO WORKERS(WORKER_ID, WORKER_NAME, WORKER_SALARY,WORKER_TYPE)
values (4, ' Марія Ленько', 20000, 'менеджер');

INSERT INTO WORKERS(WORKER_ID, WORKER_NAME, WORKER_SALARY,WORKER_TYPE)
values (5, 'Богдан Крупський', 15080, 'менеджер');
```

Лістинг А.7 Заповнення таблиці BOOKSTORE

```
INSERT INTO BOOKSTORE( STORE_ID, STORE_NAME, STORE_ADDRESS) values (1,
'КнигарняЄ', 'Львів,вул. Княгині Ольги 35' );

INSERT INTO BOOKSTORE( STORE_ID, STORE_NAME, STORE_ADDRESS) values (2,
'КнигарняЄ', 'Львів,вул. Вагонна 20');

INSERT INTO BOOKSTORE( STORE_ID, STORE_NAME, STORE_ADDRESS) values (3,
'КнигарняЄ', 'Київ,вул. Перемоги 22');
```

Лістинг А.8 Заповнення таблиці BOOKSTORE_BOOKS

```
INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (1,
1, 10);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (1,
2, 100);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (1,
10, 200);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (1,
13, 100);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (1,
20, 100);
```

```
INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (1,
17, 10);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (1,
8, 100);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (1,
14, 100);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (1,
17, 50);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (1,
13, 100);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (1,
4, 100);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (1,
9, 80);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
1, 50);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
16, 100);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
8, 100);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
15, 80);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
14, 50);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
2, 60);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
6, 120);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
13, 80);
```

```
INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
9, 50);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
16, 100);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
8, 100);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
15, 80);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
10, 50);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
3, 60);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
18, 20);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
4, 80);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
6, 120);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (2,
13, 80);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (3,
10, 50);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (3,
3, 100);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (3,
7, 96);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (3,
5, 80);

INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (3,
11, 50);
```

```
INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (3,  
13, 60);
```

```
INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (3,  
8, 200);
```

```
INSERT INTO BOOKSTORE_BOOKS( STORE_ID, BOOK_ID, BOOK_QUANTITY ) values (3,  
19, 80);
```

Лістинг А.9 Заповнення таблиці CUSTOMERS

```
INSERT INTO CUSTOMERS(ID,CUSTOMER_NAME,  
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 1, ' Семків  
Олег', 'Батятичі', '0692522681','OLEHSEMKIV@gmail.com');
```

```
INSERT INTO CUSTOMERS(ID,CUSTOMER_NAME,  
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 2, ' Ільків  
Марта', 'Самбір', '0982561479','MartaIlkiv@gmail.com');
```

```
INSERT INTO CUSTOMERS(ID,CUSTOMER_NAME,  
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 3, ' Гула Ольга',  
'Львів', '0682569871','HulaOx@gmail.com');
```

```
INSERT INTO CUSTOMERS(ID,CUSTOMER_NAME,  
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 4, ' Весела  
Оксана', 'Добромиль', '0692523698','VESELKA122@gmail.com' );
```

```
INSERT INTO CUSTOMERS(ID,CUSTOMER_NAME,  
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 5, ' Галайко  
Григорій', 'Хирів', '0693652681','Halhruh2542@gmail.com' );
```

```
INSERT INTO CUSTOMERS(ID,CUSTOMER_NAME,  
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 6, ' Федечко  
Роман', 'Київ', '069894581','RomanFed879@gmail.com' );
```

```
INSERT INTO CUSTOMERS(ID,CUSTOMER_NAME,  
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 7, ' Белей  
Вікторія ', 'Стрий', '0972651846','Viktoriaaa12@gmail.com' );
```

```
INSERT INTO CUSTOMERS(ID,CUSTOMER_NAME,  
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 8, ' Ленько  
Світлана', 'Львів', '0691478537','LenkoSvitlana@gmail.com' );
```

```

INSERT                                INTO                                CUSTOMERS (ID,CUSTOMER_NAME,
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 9, ' Пришляк
Віра', 'Доброміль', '0668951432','ViraPr1234@gmail.com');

INSERT                                INTO                                CUSTOMERS (ID,CUSTOMER_NAME,
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 10, ' Волошин
Анна', 'Одесса', '0692522451','AnnaVol109@gmail.com' );

INSERT                                INTO                                CUSTOMERS (ID,CUSTOMER_NAME,
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 11, ' Кріль
Каріна', 'Тернопіль', '0962581473','KrillKarina09@gmail.com');

INSERT                                INTO                                CUSTOMERS (ID,CUSTOMER_NAME,
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 12, ' Бабій
Аліна', 'Івано-Франківськ', '0688561473' , 'AlinaBabiy@gmail.com');

INSERT                                INTO                                CUSTOMERS (ID,CUSTOMER_NAME,
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 13, ' Босак
Василь', 'Київ', '0965252148','Bossaakkl09@gmail.com' );

INSERT                                INTO                                CUSTOMERS (ID,CUSTOMER_NAME,
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 14, ' Теш
Сергій', 'Новий Калинів', '0698521473','SerhiyT56509@gmail.com' );

INSERT                                INTO                                CUSTOMERS (ID,CUSTOMER_NAME,
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 15, ' Малецький
Олег', 'Львів', '0698521473','OlehMal989@gmail.com' );

INSERT                                INTO                                CUSTOMERS (ID,CUSTOMER_NAME,
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 16, ' Луш Марта',
'Львів', '0965214756','MartaL7564@gmail.com' );

INSERT                                INTO                                CUSTOMERS (ID,CUSTOMER_NAME,
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 17, ' Хадай
Андріана', 'Чернівці', '0982514763','Andriana0987@gmail.com');

INSERT                                INTO                                CUSTOMERS (ID,CUSTOMER_NAME,
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 18, ' Кузьмін
Людмила', 'Дрогобич', '0987652341','ludmylaK123@gmail.com');

INSERT                                INTO                                CUSTOMERS (ID,CUSTOMER_NAME,
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 19, ' Фалач
Тарас', 'Сколе', '0999733678','tarasF345@gmail.com');

```

```

INSERT                                INTO                                CUSTOMERS (ID,CUSTOMER_NAME,
CUSTOMER_ADDRESS,CUSTOMER_PHONE,CUSTOMER_EMAIL) values ( 20, 'Паврозник
Ірина', 'Чернівці', '0974513692','Iryna546Pav@gmail.com');

```

Лістинг А.10 Заповнення таблиці ORDERS

```

INSERT                                INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (1,1,3,1,1,'2021-09-08', 'прийнято в
роботу', 'cash');

```

```

INSERT                                INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (2,2,1,1,2,'2021-10-08',
'оформлено', 'card');

```

```

INSERT                                INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (3,3,2,3,5,'2021-12-30',
'відправлено', 'cash');

```

```

INSERT                                INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (4,3,2,3,5,'2021-05-
18','відправлено', 'cash');

```

```

INSERT                                INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (5,9,4,3,5,'2021-09-20',
'виконано', 'cash');

```

```

INSERT                                INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (6,11,1,1,2,'2021-02-08',
'оформлено', 'card');

```

```

INSERT                                INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (7,3,5,1,2,'2021-05-15',
'відправлено', 'cash');

```

```

INSERT                                INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD

```

```

ER_STATUS,ORDER_PAEMENT) values (8,4,1,2,3,'2022-04-11', 'прийнято в
роботу','cash');

INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (9,7,3,1,3,'2022-04-08',
'оформлено','card');

INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (10,4,1,2,2,'2022-03-18',
'оформлено','card');

INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (11,2,1,3,5,'2022-04-17',
'відправлено','cash');

INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (12,5,3,1,3,'2022-04-09',
'оформлено','card');

INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (13,6,5,1,1,'2022-03-26',
'оформлено','cash');

INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (14,20,1,2,2,'2021-09-08',
'оформлено','cash');

INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (15,16,5,1,4,'2022-03-20', 'виконано',
'card');

INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT) values (16,18,4,2,5,'2022-05-01',
'відправлено','cash');

```



```
INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT)          values          (17,17,2,1,2,'2022-05-11',
'оформлено','cash');
```

```
INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT)  values  (18,8,5,1,3,'2022-05-08',  'прийнято в
роботу','card');
```

```
INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT)          values          (19,10,3,2,5,'2022-05-14',
'відпратлено','cash');
```

```
INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT)          values          (20,15,3,2,2,'2022-05-16',
'оформлено','cash');
```

```
INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT)          values          (21,17,3,2,1,'2022-05-16',
'нове','cash');
```

```
INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT)          values          (22,11,1,2,1,'2022-05-17',
'нове','cash');
```

```
INSERT INTO
ORDERS (ORDER_ID,CUSTOMER_ID,DELIVERY_ID,STORE_ID,WORKER_ID,ORDER_DATA,ORD
ER_STATUS,ORDER_PAEMENT)          values          (23,13,2,3,4,'2022-05-20',
'нове','cash');
```

Лістинг A.11 Заповнення таблиці BOOKS_ORDERS

```
INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(1,1,1);
```

```
INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(1,2,5);
```

```

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(2,3,3);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(2,1,7);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(3,1,5);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(4,5,3);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(4,1,1);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(4,3,2);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(5,5,1);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(5,10,1);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(6,18,2);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(7,13,12);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(8,8,8);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(8,14,2);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(8,1,5);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(9,6,6);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(9,11,2);

```

```

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(9,5,1);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(10,20,2);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(10,19,1);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(11,16,1);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(12,13,10);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(12,1,1);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(13,11,1);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(14,16,2);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(15,10,4);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(16,14,3);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(17,5,1);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(17,16,2);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(17,9,3);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(17,3,2);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(18,4,1);

```

```

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(19,17,2);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(19,7,1);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(20,19,1);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(21,4,1);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(22,14,3);

INSERT INTO BOOKS_ORDERS (ORDER_ID,BOOK_ID, ORDER_QUANTITY ) values
(23,10,1);

```

Лістинг А.12 Заповнення таблиці WISH_LIST

```

INSERT INTO
WISH_LIST(WISH_LIST_ID,WISH_BOOK_NAME,WISH_BOOK_TYPE,CUSTOMER_ID) values
(1, ' Чуеш коли приїдеш додому?', 'класика', 2);

INSERT INTO
WISH_LIST(WISH_LIST_ID,WISH_BOOK_NAME,WISH_BOOK_TYPE,CUSTOMER_ID) values
(2, 'Вічна любов', 'детектив', 3);

INSERT INTO
WISH_LIST(WISH_LIST_ID,WISH_BOOK_NAME,WISH_BOOK_TYPE,CUSTOMER_ID) values
(3, 'В оточенні психопатів', 'психологія', 10);

INSERT INTO
WISH_LIST(WISH_LIST_ID,WISH_BOOK_NAME,WISH_BOOK_TYPE,CUSTOMER_ID) values
(4, 'Крила', 'роман', 10);

INSERT INTO
WISH_LIST(WISH_LIST_ID,WISH_BOOK_NAME,WISH_BOOK_TYPE,CUSTOMER_ID) values
(5, 'Щоденник 3', 'комікс', 4);

INSERT INTO
WISH_LIST(WISH_LIST_ID,WISH_BOOK_NAME,WISH_BOOK_TYPE,CUSTOMER_ID) values
(6, 'Сила думки', 'психологія', 7);

```

```

INSERT                                                    INTO
WISH_LIST(WISH_LIST_ID,WISH_BOOK_NAME,WISH_BOOK_TYPE,CUSTOMER_ID) values
(7, 'Ніби мене нема', 'класика', 10);

INSERT                                                    INTO
WISH_LIST(WISH_LIST_ID,WISH_BOOK_NAME,WISH_BOOK_TYPE,CUSTOMER_ID) values
(8, 'Полімпсети', 'поезія', 1);

INSERT                                                    INTO
WISH_LIST(WISH_LIST_ID,WISH_BOOK_NAME,WISH_BOOK_TYPE,CUSTOMER_ID) values
(9, 'А що Бог їсть на сніданок?', 'дитяча література', 5);

INSERT                                                    INTO
WISH_LIST(WISH_LIST_ID,WISH_BOOK_NAME,WISH_BOOK_TYPE,CUSTOMER_ID) values
(10, 'Ніби мене нема', 'класика', 1);

INSERT                                                    INTO
WISH_LIST(WISH_LIST_ID,WISH_BOOK_NAME,WISH_BOOK_TYPE,CUSTOMER_ID) values
(11, ' І знову я взлізаю в танк', 'класика', 9);

INSERT                                                    INTO
WISH_LIST(WISH_LIST_ID,WISH_BOOK_NAME,WISH_BOOK_TYPE,CUSTOMER_ID) values
(12, 'Дитячий сміх', 'дитяча література', 10);

INSERT                                                    INTO
WISH_LIST(WISH_LIST_ID,WISH_BOOK_NAME,WISH_BOOK_TYPE,CUSTOMER_ID) values
(13, 'День, що навчив мене жити', 'психологія', 8);

INSERT                                                    INTO
WISH_LIST(WISH_LIST_ID,WISH_BOOK_NAME,WISH_BOOK_TYPE,CUSTOMER_ID) values
(14, 'Музей покинутих предків', 'роман', 6);

```

Лістинг А.13 Заповнення таблиці REVIEWS

```

INSERT                                                    INTO REVIEWS
(REVIEW_ID,BOOK_ID,CUSTOMER_ID,REVIEW_RATING,REVIEW_COMMENT,REVIEW_DATA)
values (1,1,1,10, 'швидка робота та дуже гарна книга', '2021-09-08');

INSERT                                                    INTO REVIEWS
(REVIEW_ID,BOOK_ID,CUSTOMER_ID,REVIEW_RATING,REVIEW_COMMENT,REVIEW_DATA)
values (2,4,10,8, 'швидка робота та дуже гарна книга', '2022-05-08');

```

```

INSERT                                INTO                                REVIEWS
(REVIEW_ID,BOOK_ID,CUSTOMER_ID,REVIEW_RATING,REVIEW_COMMENT,REVIEW_DATA)
values (3,14,10,5, 'довго відправляли, але книжка класна', '2022-02-09');

```

```

INSERT                                INTO                                REVIEWS
(REVIEW_ID,BOOK_ID,CUSTOMER_ID,REVIEW_RATING,REVIEW_COMMENT,REVIEW_DATA)
values (4,10,13,10, 'швидка та якісна робота менеджерів, книжка відповідає
дійсності','2022-05-18');

```

```

INSERT                                INTO                                REVIEWS
(REVIEW_ID,BOOK_ID,CUSTOMER_ID,REVIEW_RATING,REVIEW_COMMENT,REVIEW_DATA)
values (5,16,20,10, 'все супер', '2022-03-10');

```

```

INSERT                                INTO                                REVIEWS
(REVIEW_ID,BOOK_ID,CUSTOMER_ID,REVIEW_RATING,REVIEW_COMMENT,REVIEW_DATA)
values (6,11,7,1, 'замовлення скасували, хоча до мене ніхто не
телефонував', '2022-05-10');

```

Лістинг А.14 Запити бази даних

```

/* асортимент усіх товарів */

```

```

SELECT  books.BOOK_ID, BOOKS.BOOK_NAME,    COALESCE(SUM(BOOK_QUANTITY),0)
as SUM_BOOK_QUANTITY

```

```

FROM bookstore LEFT JOIN bookstore_books ON bookstore_books.STORE_ID =
bookstore.STORE_ID

```

```

LEFT JOIN BOOKS ON BOOKS.BOOK_ID = bookstore_books.BOOK_ID group by
BOOKS.BOOK_ID;

```

```

/* сортування книг по жанрах*/

```

```

SELECT book_id,book_name,GENRES.GENRE_NAME,book_price

```

```

FROM BOOKS inner join

```

```

GENRES on GENRES.GENRE_ID =  BOOKS.GENRE_ID order by GENRES.GENRE_NAME
asc;

```

```

/* сортування книг за жанром роман*/

```

```

SELECT book_id,book_name,GENRES.GENRE_NAME,book_price

FROM BOOKS inner join

GENRES on GENRES.GENRE_ID = BOOKS.GENRE_ID where genres.GENRE_NAME like
'роман';

```

```

/* сортування книг по видавництвах */

```

```

SELECT book_id,book_name,PUBLISHER_NAME,book_price

FROM BOOKS inner join

publishers on publishers.PUBLISHER_ID = BOOKS.publisher_ID order by
publishers.PUBLISHER_NAME asc;

```

```

/* сортування книг Видавництво Юлії Сливки */

```

```

SELECT book_id,book_name,PUBLISHER_NAME,book_price

FROM BOOKS inner join

publishers on publishers.PUBLISHER_ID = BOOKS.publisher_ID where
publishers.PUBLISHER_NAME like 'Видавництво Юлії Сливки';

```

```

/* сортування книг за авторами */

```

```

SELECT BOOKS.BOOK_ID, books.BOOK_NAME, writers.WRITER_NAME

FROM BOOKS inner join

WRITERS on WRITERS.writer_ID = BOOKS.writer_ID

order by writers.WRITER_NAME asc;

```

```

/* список книг, автора Галини Вдовиченко */

```

```

SELECT BOOKS.BOOK_ID, books.BOOK_NAME, writers.WRITER_NAME

FROM BOOKS inner join

WRITERS on WRITERS.writer_ID = BOOKS.writer_ID

```

```

where writers.WRITER_NAME like 'Галина Вдовиченко';

/* список книг, які є у вışлісті 1 та 2 клієнта */

SELECT      CUSTOMERS.CUSTOMER_NAME, CUSTOMER_PHONE,      CUSTOMER_EMAIL,
WISH_BOOK_NAME, WISH_BOOK_TYPE

FROM wish_list inner join

customers on wish_list.CUSTOMER_ID =  customers.ID

where customers.id=1;

/* чи можна замовити на даний момент книгу чи ні */

USE BOOKSTORE;

SET GLOBAL log_bin_trust_function_creators = 1;

DELIMITER //

CREATE FUNCTION ChechBookWhichCanBeBooked ( bookId INT, wantToBuy INT )

RETURNS VARCHAR(100)

BEGIN

    DECLARE booked_quantity INT;

    DECLARE can_booked INT;

    DECLARE bookexist bool;

    DECLARE bookName varchar(100);

    SET bookexist = (SELECT exists (

```



```

SELECT

    BOOKS.BOOK_ID  FROM books WHERE BOOK_ID = bookId ));

    IF (bookexist<1) THEN

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'книги з таким кодом немає';

END IF;


    SET booked_quantity =    (SELECT COALESCE(SUM(ORDER_QUANTITY),0)

    FROM books_orders

    where BOOK_ID = bookId

group by  BOOK_ID);


    set can_booked = (SELECT  COALESCE(SUM(BOOK_QUANTITY),0)

    FROM  bookstore_books

    where BOOK_ID = bookId

group by BOOK_ID);


    IF (can_booked - booked_quantity < wantToBuy) THEN

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Немає в наявності!';

END IF;

    set bookName = ( SELECT BOOK_NAME from BOOKS where BOOK_ID =  bookId);


    RETURN (SELECT CONCAT('УПА! можна придбати:', bookName ));


END; //
```

```
DELIMITER ;
```

```
SELECT ChechBookWhichCanBeBooked(17,10);
```

```
/* сортування вартості доставки */
```

```
SELECT * FROM DELIVERY ORDER BY DELIVERY_COST;
```

```
/* статуси для конкретного клієнта */
```

```
SELECT ORDER_ID, CUSTOMERS.CUSTOMER_NAME, ORDER_ID, ORDER_STATUS,  
ORDER_DATA
```

```
FROM ORDERS LEFT JOIN CUSTOMERS ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID WHERE  
CUSTOMERS.ID= 4;
```

```
/* МЕНЕДЖЕР */
```

```
/* сортування по статусу */
```

```
SELECT CUSTOMERS.ID, CUSTOMERS.CUSTOMER_NAME, ORDER_ID, ORDER_STATUS,  
ORDER_DATA
```

```
FROM ORDERS LEFT JOIN CUSTOMERS ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID;
```

```
/* сортування по статусу нове */
```

```
SELECT o.ORDER_ID, customers.CUSTOMER_NAME,  
customers.CUSTOMER_EMAIL,customers.CUSTOMER_ADDRESS,customers.CUSTOMER_PHONE ,books. book_name, od.order_quantity, o.order_data, o.ORDER_STATUS
```

```
FROM ORDERS as o LEFT JOIN BOOKS_ORDERS as od on o.ORDER_ID = od.ORDER_ID
```

```
left join Customers on o.customer_id = customers.id
```

```
left join books on od.book_id = books.book_id
```

```

where o.ORDER_STATUS like 'нове';

/* для моніторингу асортименту */

SELECT BOOK_NAME,bookstore.STORE_ADDRESS, bookstore_books.BOOK_QUANTITY
- books_orders.ORDER_QUANTITY AS 'книги доступні до замовлення'

FROM BOOKS LEFT JOIN bookstore_books ON bookstore_books.BOOK_ID =
BOOKS.BOOK_ID

left join bookstore on bookstore.store_id = bookstore_books.store_id

LEFT JOIN books_orders ON books_orders.BOOK_ID= bookstore_books.BOOK_ID
group by books.BOOK_NAME,bookstore.STORE_ADDRESS;

/* максимальна кількість книг, замовлених*/

SELECT      books_orders.ORDER_ID,  books.book_name,  GENRES.GENRE_NAME,
sum(ORDER_QUANTITY) as `max`

FROM books_orders left join books on books_orders.BOOK_ID = books.book_ID

LEFT JOIN GENRES ON GENRES.GENRE_ID = BOOKS.GENRE_ID

group by books.book_name ORDER BY  sum(ORDER_QUANTITY) desc ;

/* відгуки по книжках */

SELECT      REVIEW_ID,      customers.customer_name,books.book_name
,REVIEW_COMMENT,REVIEW_RATING,REVIEW_DATA

FROM reviews left join customers on customers.id = reviews.CUSTOMER_ID

left join books on books.BOOK_ID = reviews.BOOK_ID;

/* вартість замовлення*/

SELECT      books_orders.order_id,      customers.CUSTOMER_NAME,
sum(books_orders.ORDER_QUANTITY * books.BOOK_PRICE) as Sum_orders

FROM books left join books_orders on books_orders.BOOK_ID = books.BOOK_ID

```

```

left join ORDERS on ORDERS.ORDER_ID = books_orders.order_id

inner join customers on customers.ID = ORDERS.CUSTOMER_ID

where orders.ORDER_DATA BETWEEN '2022-05-01' and now()

group by ORDER_ID;

/* ДИРЕКТОР */

/* перевірка на ефективність роботи */

SELECT
                                BOOKSTORE.STORE_NAME,BOOKSTORE.STORE_ADDRESS,
SUM(ORDER_QUANTITY) AS "Найбільша кількість"

FROM books_orders left join orders on books_orders.order_id = orders.
order_id

LEFT JOIN bookstore_books ON bookstore_books.book_ID =
books_orders.BOOK_ID

left join bookstore on bookstore.store_id = bookstore_books.store_ID

Where ORDER_DATA between '2022-02-05' and now() group by
bookstore.STORE_ADDRESS order by SUM(ORDER_QUANTITY) desc;

/* сортування доставок, по кількості замовлень */

SELECT
        delivery.DELIVERY_ID,
                                DELIVERY.DELIVERY_NAME,
count(orders.DELIVERY_ID) as 'кількість виконаних замовлень'

FROM delivery LEFT JOIN ORDERS ON ORDERS.DELIVERY_ID =
delivery.DELIVERY_ID

group by delivery.DELIVERY_ID;

/* список працівників, що отримали хороші відгуки по роботі */

SELECT
        REVIEW_ID,
                                customers.customer_name,books.book_name
,REVIEW_COMMENT,REVIEW_RATING,REVIEW_DATA,workers.WORKER_NAME,
orders.order_id

```

```

FROM reviews left join customers on customers.id = reviews.CUSTOMER_ID

left join books on books.BOOK_ID = reviews.BOOK_ID

left join orders on orders.customer_id = customers.id

left join workers on workers.WORKER_ID = orders.WORKER_ID

where REVIEW_DATA between '2022-05-01' and now() order by REVIEW_RATING
asc;

```

```

/* працівники, що заслуговують на премію */

```

```

SELECT WORKERS.WORKER_ID, WORKERS.WORKER_NAME

FROM WORKERS left join orders on orders.WORKER_ID = WORKERS.WORKER_ID
WHERE ORDER_DATA between '2022-04-01' AND '2022-05-01'

group by WORKERS.worker_id having count(order_id) > 2;

```

```

USE BOOKSTORE;

SET GLOBAL log_bin_trust_function_creators = 1;

```

```

DELIMITER //

```

```

CREATE FUNCTION salary_accruall

(workerId INT)

returns varchar(45)

```

```

BEGIN

```

```

Update WORKERS

```

```

Set WORKER_SALARY = WORKER_SALARY + 100

where WORKERS.WORKER_ID = workerId;

RETURN 'updated';

END; //
```

```

DELIMITER ;

SELECT salary_accruall (5);

select * from WORKERS;
```

```

/* клієнти, які заслуговують на знижку */

SELECT          orders.ORDER_ID,          customers.customer_name,
sum(books_orders.ORDER_QUANTITY * books.BOOK_PRICE) as ' Сума замовлення'

FROM orders left join CUSTOMERs on orders.CUSTOMER_ID = CUSTOMERs.ID

left join books_orders on books_orders.ORDER_ID = orders.order_id

left join books on books.BOOK_ID = books_orders.BOOK_ID

Where ORDER_DATA between '2022-05-01' and now()

group by ORDER_ID having          sum(books_orders.ORDER_QUANTITY *
books.BOOK_PRICE) > 1000;
```