

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Кафедра інформаційних систем та мереж

РОЗРАХУНКОВА РОБОТА

З дисципліни «Організація баз даних та знань»

на тему:

**«ПРОЕКТУВАННЯ ТА ЗАСТОСУВАННЯ БАЗИ ДАНИХ
ІНФОРМАЦІЙНОЇ СИСТЕМИ ПОШТА»**

Виконав студент групи КН-207

Гуменчук Б.Р.

Оцінка	Балів	Дата

Керівник

Мельникова Н. І.

Львів – 2019

ЗАВДАННЯ

на розрахункову роботу з дисципліни «Організація баз даних та знань»

студента групи КН-207

Тема: Проектування бази даних інформаційної системи

«Видавництво»

ЗМІСТ ЗАВДАННЯ

№ з/п	Зміст завдання	Примітка
1	Здійснити аналітичний огляд літератури за заданою темою та обґрунтувати вибір інструментальних засобів проектування.	
2	Визначення та опис предметної області.	
3	Побудова моделі типу «сутність-зв'язок».	
4	Побудова логічної структури бази даних.	
5	Побудова обмежень відношень бази даних. Нормалізація відношень бази даних.	
6	Виконання над відношеннями операцій реляційної алгебри.	
7	Оформити записку до розрахункової роботи згідно з вимогами Міжнародних стандартів, дотримуючись такого змісту: <ul style="list-style-type: none">• вступ;• визначення та опис предметної області;• концептуальна модель “сутність-зв’язок”;• логічна структура бази даних;• розроблення обмежень відношень бази даних;• нормалізація бази даних;• виконання операцій реляційної алгебри;• висновки;• література;• додатки.	

Завдання прийнято до виконання: _____.2019р

Керівник роботи:

Мельникова Н. І.

ЗМІСТ

1. Аналітичний огляд літератури	4
2. Визначення та опис предметної області.....	6
2.1. Аналіз вимог до бази даних	6
2.2. Створення ділової моделі бази по видачі паспортів	7
3. Розроблення бази даних.	7
Концептуальна модель «сутність-зв'язок».....	7
4. Логічна структура бази даних.....	8
5. Обмеження та нормалізація	8
5.1. Побудова обмежень бази даних.....	8
5.2. Нормалізація	9
6. Реалізація бази даних.....	10
Виконання над відношеннями операцій реляційної алгебри	10
Висновок.	10

1. Аналітичний огляд літератури

Створення та впровадження в практику сучасних інформаційних систем висуває нові задачі проектування, які неможливо розв'язувати традиційними прийомами та методами. Велику увагу необхідно приділяти питанням проектування баз даних як одній із основних складових елементів. Від того, наскільки успішно буде спроектовано базу даних, залежить ефективність функціонування системи в цілому, її життєздатність і можливість розширення й подальшого розвитку. Тому питання проектування баз даних виділяються як окремий, самостійний напрям робіт при розробці інформаційних систем.

Процес проектування бази даних поділяється на етапи, кожний з яких передбачає виконання певних дій. Перший етап-розробка інформаційно-логічної моделі даних предметної області, який базується на описі предметної області, отриманому в результаті її обстеження. На цьому етапі спочатку визначають склад і структуру даних предметної області, які мають міститись у базі даних та забезпечувати виконання запитів, задач і застосувань користувача. Ці дані мають форму реквізитів, що містяться в різних документах - джерелах завантаження бази даних. Аналіз виявлених даних дозволить визначити функціональні залежності реквізитів, які використовують для виділення інформаційних об'єктів, що відповідають вимогам нормалізації даних. Подальше визначення структурних зв'язків між об'єктами дозволяє побудувати інформаційно-логічну модель.

Другий етап - визначення логічної структури бази даних. Для реляційної бази даних цей етап є значною мірою формальним, оскільки інформаційно-логічна модель відображається в структуру реляційної бази даних адекватно.

Наступний етап - конструювання таблиць бази даних, який здійснюється засобами СУБД, та узгодження їх із замовником. Структура таблиць бази даних задається за допомогою засобів опису (конструювання) таблиць у СУБД із цілковитою відповідністю інформаційним об'єктам. Крім таблиць, проектувальники розробляють й інші об'єкти бази даних, які призначені, з

одного боку, для автоматизації роботи з базою, а з іншого - для обмеження функціональних можливостей роботи з базою (безпека бази даних). Після формування структури таблиць база даних може наповнюватись даними з документів-джерел.

На даний момент створення бази даних для видавництва є актуальною, оскільки зараз лише відбувається перехід від паперової документації до комп'ютерної. Повноцінна, продумана і захищена база даних є дуже актуальною, тому що вона дозволить користуватися нею не лише людям, що безпосередньо займаються видавничою справою, а й клієнтам, які зможуть отримати доступ до процесу роботи над своїм замовленням. При створенні бази даних необхідно звернути на велику кількість ключових моментів, а саме: зберігання інформації про людину, що оформляла замовлення, створення різних рівнів доступу, для можливості надання перегляду інформації без змоги редагувати або створювати нові дані, зберігання копій документів людини, що оформляє замовлення.

2. Визначення та опис предметної області

2.1. Аналіз вимог до бази даних

В базі даних повинна зберігатися інформація:

- про виконавця: код працівника, його ім'я, дата народження, адреса електронної пошти;
- про замовника: код замовника, його ім'я, прізвище, контактний номер, адреса електронної пошти, дата народження;
- про замовлення: код замовлення, замовник, час початку і кінця роботи над замовленням, виконавець, ціна;
- про друкарні: код друкарні, розмір аркушів для друку, матеріал обгортки, ціна;
- про магазини - книгарні: код книгарні, адреса, назва;

2.2. Створення ділової моделі бази по видачі паспортів

Створимо ділову модель організації по видачі паспортів.

Classes	Customer	Performer	Printer	Book	Shops
Functions					
Order book	*	*		*	*
Getting info about book				*	
Sending/Receiving book to/from shop				*	*
Working with printing books		*	*	*	
Earning salary		*			
Getting info about performer		*			
Getting info about customer	*				

Рис 1. Ділова модель організації «Пошта»

Зірочки у діловій моделі вказують зв'язки між об'єктами, що забезпечують виконання певної функції. Для цього ідентифікують класи даних, які приймають участь у виконанні кожної функції організації. Тобто формується ряд поглядів на базу даних з різних боків її практичного використання.

3. Розроблення бази даних.

Концептуальна модель «сутність-зв'язок»

Створена спочатку схема повинна представляти базу в укрупненому вигляді і відображати об'єкти предметної області та зв'язки між ними. Схему бази даних, наведену на рис.2, інакше називають моделлю об'єкт-зв'язок, або сутність-зв'язок. (ER-моделлю).

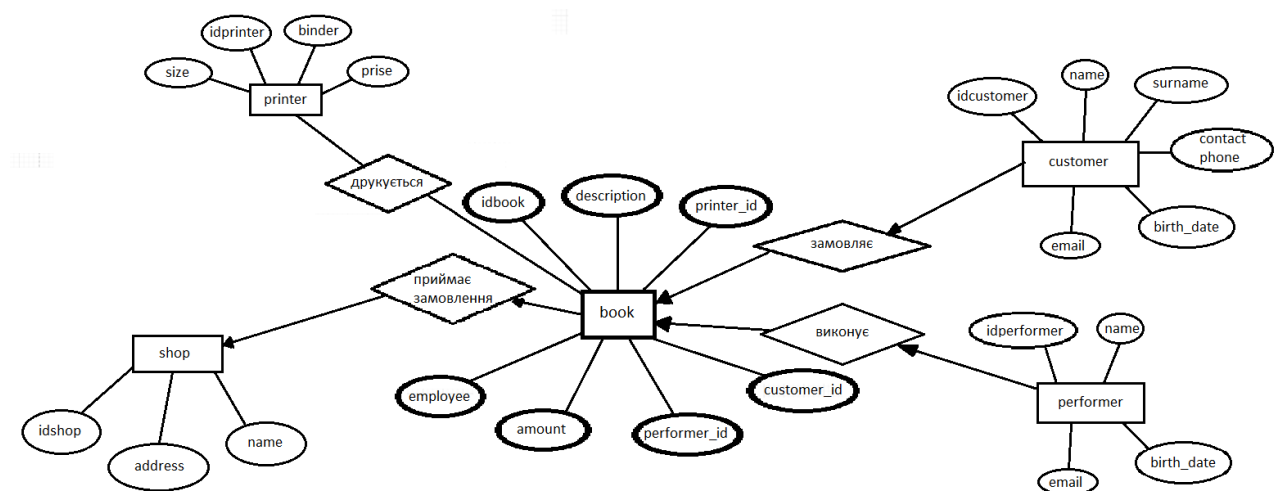


Рис. 2 Концептуальна схема бази даних "Видавництво"

4. Логічна структура бази даних

Назва таблиці	Ім'я поля	Тип даних
Виконавець	<u>КодВиконавця</u>	Лічильник
	Ім'я	Текстовий
	ЕлектроннаПошта	Текстовий
	ДатаНародження	Дата
Книга/замовлення	<u>КодЗамовлення</u>	Лічильник
	Опис	Текстовий
	Кількість	Числовий
	Виконавець	Текстовий
	КодДрукарні	Числовий(Майстер підстановки)
	КодВиконавця	Числовий(Майстер підстановки)
	КодЗамовника	Числовий(Майстер підстановки)
Замовник	<u>КодЗамовника</u>	Лічильник
	Ім'я	Текстовий
	Прізвище	Текстовий
	КонтактнийНомер	Текстовий
	ДатаНародження	Дата
	ЕлектроннаПошта	Такстовий
Магазин	<u>КодМагазину</u>	Лічильник
	Назва	Текстовий
Друкарня	<u>КодДрукарні</u>	Лічильник
	КодОбкладинки	Числовий(Майстер Підстановки)
	КодРозміру	Числовий(Майстер Підстановки)
Обкладинки	<u>КодОбкладинки</u>	Лічильник
	Назва	Числовий
	Ціна	Числовий
Розмірність	КодРозмірності	Лічильник
	Розмір	Числовий
	Ціна	Числовий

Таблиця 6.1.

Підкреслені поля є первинними ключами.

5. Обмеження та нормалізація

5.1. Побудова обмежень бази даних

Для всіх primary key встановлюємо обмеження NOT NULL, ON DELETE RESTRICT, ON UPDATE CASCADE.

Більшість відношень в даній базі є типу «один до багатьох», тобто коли одне з полів, за якими здійснюється зв'язок, — ключове. Тоді одному запису таблиці А відповідає кілька записів таблиці В, але запис із таблиці В не може мати більше від одного, який відповідає йому, записі в А.

Тому всюди також встановлюємо обмеження NOT NULL, ON DELETE RESTRICT, ON UPDATE CASCADE, оскільки всі дані повинні бути збережені.

5.2. Нормалізація

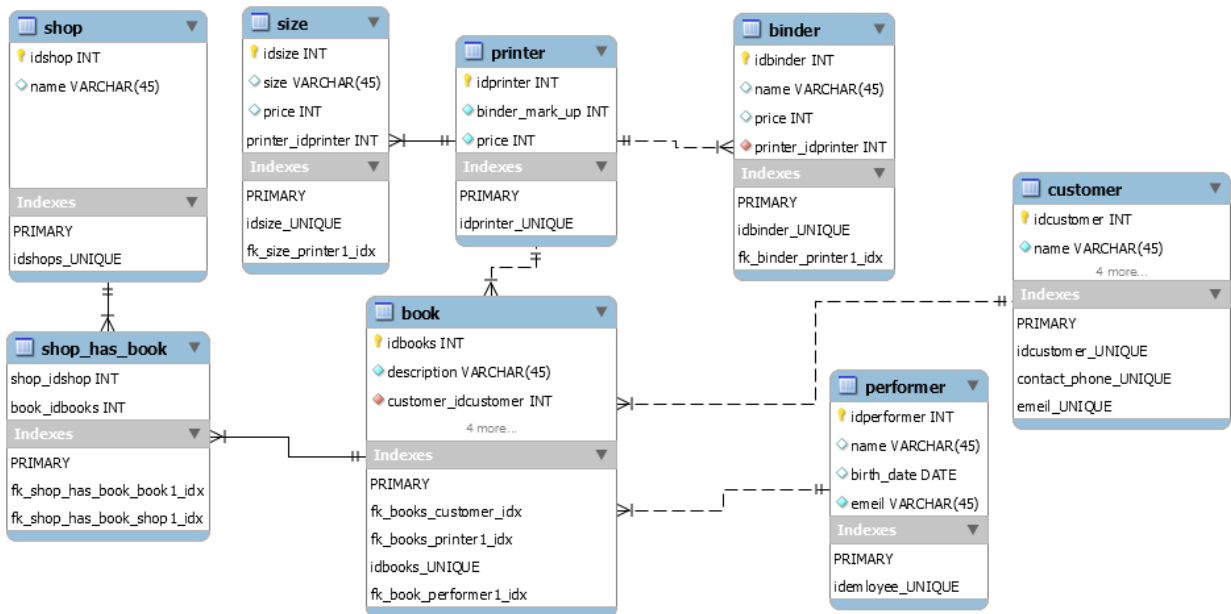


Рис.2. Схема БД «Видавництво».

База даних відноситься до другої нормальної форми, оскільки

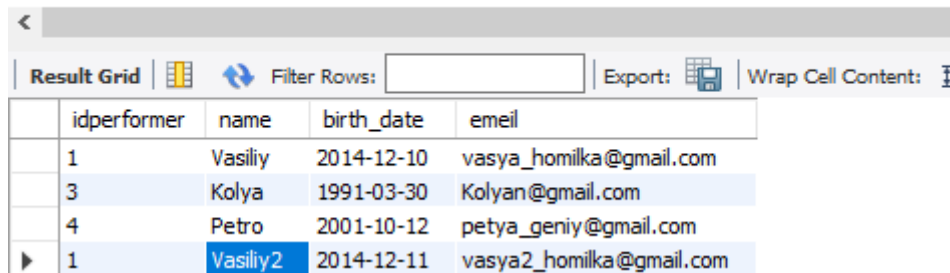
- кожна таблиця має основний ключ: мінімальний набір колонок, які ідентифікують запис;
- уникаються повторення груп (категорії даних, що можуть зустрічатися різну кількість разів у різних записах) правильно визначаючи ключові атрибути;
- атомарність: кожен атрибут має лише одне значення, а не множину значень;
- дані, що повторно з'являються в декількох рядках, винесені в окремі таблиці (в нашому випадку – це магазини).

6. Реалізація бази даних.

Виконання над відношеннями операцій реляційної алгебри

1. Запит на виконання об'єднання performer і performer2:

```
245 • select * from performer union select * from performer2;
```



The screenshot shows a database interface with a query result grid. The grid has columns: idperformer, name, birth_date, and email. The data is as follows:

	idperformer	name	birth_date	email
	1	Vasiliy	2014-12-10	vasya_homilka@gmail.com
	3	Kolya	1991-03-30	Kolyan@gmail.com
	4	Petro	2001-10-12	petya_geniy@gmail.com
▶	1	Vasiliy2	2014-12-11	vasya2_homilka@gmail.com

2. Запит на виконання перетину.

```
246 • select * from performer where idperformer in (select idperformer from performer2);
```

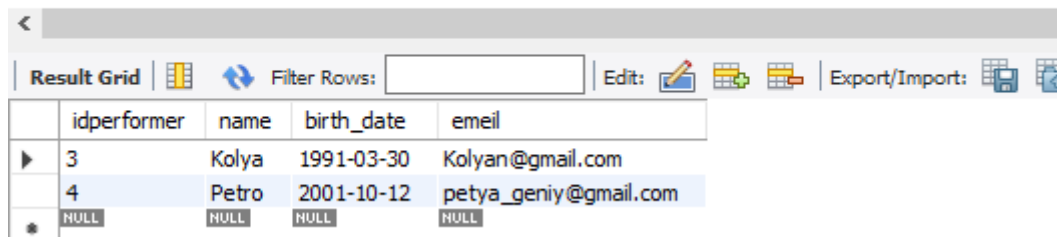


The screenshot shows a database interface with a query result grid. The grid has columns: idperformer, name, birth_date, and email. The data is as follows:

	idperformer	name	birth_date	email
▶	1	Vasiliy	2014-12-10	vasya_homilka@gmail.com
*	NULL	NULL	NULL	NULL

3. Запит на виконання різниці.

```
219 • SELECT * FROM performer  
220 WHERE idperformer NOT IN (SELECT idperformer FROM performer2);
```



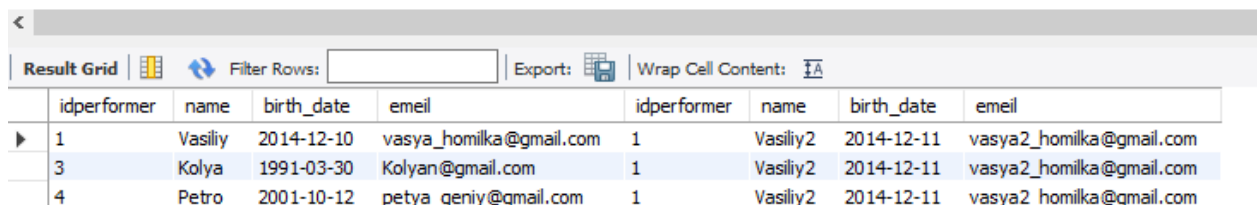
The screenshot shows a database interface with a query result grid. The grid has columns: idperformer, name, birth_date, and email. The data is as follows:

	idperformer	name	birth_date	email
▶	3	Kolya	1991-03-30	Kolyan@gmail.com
	4	Petro	2001-10-12	petya_geniy@gmail.com
*	NULL	NULL	NULL	NULL

4. Запит на виконання декартового добутку.

```
221
```

```
222 • SELECT * FROM performer, performer2;
```



The screenshot shows a database interface with a query result grid. The grid has columns: idperformer, name, birth_date, email, idperformer, name, birth_date, email. The data is as follows:

	idperformer	name	birth_date	email	idperformer	name	birth_date	email
▶	1	Vasiliy	2014-12-10	vasya_homilka@gmail.com	1	Vasiliy2	2014-12-11	vasya2_homilka@gmail.com
	3	Kolya	1991-03-30	Kolyan@gmail.com	1	Vasiliy2	2014-12-11	vasya2_homilka@gmail.com
	4	Petro	2001-10-12	petya_geniy@gmail.com	1	Vasiliy2	2014-12-11	vasya2_homilka@gmail.com

5. Запит на виконання проєкції.

	idperformer	name	birth_date	emeil
▶	1	Vasiliy	2014-12-10	vasya_homilka@gmail.com
	3	Kolya	1991-03-30	Kolyan@gmail.com
	4	Petro	2001-10-12	petya_geniy@gmail.com
*	NULL	NULL	NULL	NULL

```
174 • create view performer3 as select distinct idperformer, name, emeil from performer;
```

```
175
```

```
176 • select * from performer3;
```

```
177
```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:			
	idperformer	name	emeil
▶	1	Vasiliy	vasya_homilka@gmail.com
	3	Kolya	Kolyan@gmail.com
	4	Petro	petya_geniy@gmail.com

6. Запит на виконання селекції.

```
174 • select * from performer3 where idperformer>2 and idperformer<5;
```

```
175
```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:			
	idperformer	name	emeil
▶	3	Kolya	Kolyan@gmail.com
	4	Petro	petya_geniy@gmail.com

7. Запит на виконання натурального з'єднання.

```
174 • select performer3.idperformer, performer3.name, performer.birth_date, performer3.emeil from performer3, performer
```

```
175 where performer3.idperformer = performer.idperformer;
```

```
176
```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:				
	idperformer	name	birth_date	emeil
▶	1	Vasiliy	2014-12-10	vasya_homilka@gmail.com
	3	Kolya	1991-03-30	Kolyan@gmail.com
	4	Petro	2001-10-12	petya_geniy@gmail.com

8. Виконаємо ліве з'єднання.

```
182 • select performer.idperformer, performer.name, performer.emeil, book.description, book.amount
```

```
183 from performer left join book on
```

```
184 performer.idperformer = book.performer_idperformer;
```

```
185
```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:					
	idperformer	name	emeil	description	amount
▶	1	Vasiliy	vasya_homilka@gmail.com	book_1	10
	1	Vasiliy	vasya_homilka@gmail.com	book_2	10
	1	Vasiliy	vasya_homilka@gmail.com	book_3	10
	3	Kolya	Kolyan@gmail.com	NULL	NULL
	4	Petro	petya_geniy@gmail.com	NULL	NULL

9. Виконаємо умовне з'єднання двох таблиць.

```
188 • select performer.name, performer.emeil, book.idbooks, book.amount, book.description
189 from performer inner join book on book.performer_idperformer = performer.idperformer
190 where performer.name = "Vasiliy";
```

Result Grid					
Filter Rows: <input type="text"/>					
Export: Wrap Cell Content:					
	name	emeil	idbooks	amount	description
▶	Vasiliy	vasya_homilka@gmail.com	4	10	book_1
	Vasiliy	vasya_homilka@gmail.com	5	10	book_2
	Vasiliy	vasya_homilka@gmail.com	6	10	book_3

10. Виконаємо умовне з'єднання трьох таблиць.

```
192 • select performer.name, performer.emeil, book.idbooks, book.amount, book.description, customer.name, customer.surname
193 from (performer inner join book) inner join customer
194 on book.performer_idperformer = performer.idperformer
195 and book.customer_idcustomer = customer.idcustomer
196 where performer.name in ("Vasiliy", "Kolya");
```

Result Grid						
Filter Rows: <input type="text"/>						
Export: Wrap Cell Content:						
	name	emeil	idbooks	amount	description	name surname
▶	Vasiliy	vasya_homilka@gmail.com	4	10	book_1	Ivan Franko
	Vasiliy	vasya_homilka@gmail.com	5	10	book_2	Ivan Franko
	Vasiliy	vasya_homilka@gmail.com	6	10	book_3	Ivan Franko

11. Виконаємо підзапит в умові відбору.

```
198 • select performer.idperformer, performer.name, book.description, book.amount
199 from performer inner join book
200 on performer.idperformer = book.performer_idperformer
201 where book.performer_idperformer
202 in (select performer.idperformer from performer where performer.name in ("Vasiliy", "Kolya"))
203 order by performer.emeil desc limit 2;
```

Result Grid				
Filter Rows: <input type="text"/>				
Export: Wrap Cell Content: Fetch rows:				
	idperformer	name	description	amount
▶	1	Vasiliy	book_1	10
	1	Vasiliy	book_2	10

12. Сортування в алфавітному порядку по Name.

```
171 • select idperformer, name, emeil from performer
172 where emeil like "%gmail%" order by name asc;
```

Result Grid			
Filter Rows: <input type="text"/>			
Edit:			
	idperformer	name	emeil
▶	3	Kolya	Kolyan@gmail.com
	4	Petro	petya_geniy@gmail.com
	1	Vasiliy	vasya_homilka@gmail.com
*	NULL	NULL	NULL

13. Сортуння в зворотному порядку по email.

```
174 • select name, email from performer
175     order by email desc limit 4;
176
```

Result Grid		Filter Rows:	Export:
	name	email	
▶	Vasiliy	vasya_homilka@gmail.com	
	Petro	petya_geniy@gmail.com	
	Kolya	Kolyan@gmail.com	

14. Список робітників до групування.

```
177 • select performer.name, performer.email, description
178     from performer inner join book
179     on performer.idperformer = book.performer_idperformer;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	name	email	description	
▶	Vasiliy	vasya_homilka@gmail.com	book_1	
	Vasiliy	vasya_homilka@gmail.com	book_2	
	Vasiliy	vasya_homilka@gmail.com	book_3	

Після групування

```
177 • select performer.name, performer.email, description
178     from performer inner join book
179     on performer.idperformer = book.performer_idperformer
180     group by description;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	name	email	description	
▶	Vasiliy	vasya_homilka@gmail.com	book_1	
	Vasiliy	vasya_homilka@gmail.com	book_2	
	Vasiliy	vasya_homilka@gmail.com	book_3	

15. Процедура обрахунку кількості замовлень від одного замовника у одного виконавця.

```
176 delimiter //
177 • create procedure search_book_count_by_customer (in performer varchar(45), in customer varchar(45))
178 • begin
179   declare error_out varchar(45);
180   set error_out = "no such customer or performer";
181   if (performer = "Vasiliy" and customer = "Ivan") then
182     begin
183       create table if not exists mydb.stats (Customer varchar(45), Amount int);
184       truncate mydb.stats;
185       insert into mydb.stats select customer.name as Customer, count(book.performer_idperformer) as Amount
186       from (performer inner join book) inner join customer
187       on book.performer_idperformer = performer.idperformer and book.customer_idcustomer = customer.idcustomer
188       where performer = performer.name and customer = customer.name;
189     end;
190   else select error_out;
191   end if;
192 end
193 //delimiter ;

195 • call search_book_count_by_customer ("Vasiliy", "Ivan");
196 • select * from stats;
```

Customer	Amount
Ivan	3

```
195 • call search_book_count_by_customer ("Mykola", "Ivan");
196 • select * from stats;
```

error_out
no such customer or performer

16. Визначити кількість замовлень у кожного робітника

```
171 • select performer.name as Pname, count(book.performer_idperformer) as Performers
172   from performer inner join book on book.performer_idperformer = performer.idperformer
173   group by name;
```

Pname	Performers
Vasiliy	3
Kolya	2

17. Визначимо сумарну кількість копій замовлення.

```
173 • select performer.name as Pname, sum(book.amount) as Numbook
174     from performer inner join book
175     on book.performer_idperformer = performer.idperformer
176     group by performer.name with rollup;
```

	Pname	Numbook
▶	Kolya	20
	Vasiliy	30
	NULL	50

18. Визначити середню кількість копій яку має зробити виконавець не враховуючи повторів.

```
173 • select performer.name as Pname, avg(distinct book.amount) as Numbook
174     from performer inner join book
175     on book.performer_idperformer = performer.idperformer
176     group by performer.name;
```

	Pname	Numbook
▶	Kolya	10.0000
	Vasiliy	10.0000

Висновок: виконавши дану розрахункову роботу я оволодів навичками створення бази даних, побудови концептуальних моделей «сутність-зв'язок». Також навчився створювати логічну структуру бази даних, накладати на неї обмеження і нормалізувати. Після створення бази даних відповідно до своєї предметної області я навчився виконувати над відношеннями операції реляційної алгебри.