

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Розрахункова робота

з дисципліни

«Організація баз даних та знань»

на тему:

**«Проектування бази даних інформаційної системи
Інтернет-магазин»**

Виконав студент групи КН-207

Шиманський П.С.

Оцінка	Балів	Дата

Керівник

Мельникова Н.І.

Львів – 2019 р.

ЗАВДАННЯ

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

студента групи КН-207

Тема: Проектування бази даних інформаційної системи

«Інтернет-магазин»

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 інтернет-магазину.....	7
3. Розроблення бази даних	8
3.1 Концептуальна модель «сутність-зв'язок».....	8
4. Логічна структура даних.....	9
5. Обмеження та нормалізація.....	10
5.1 Побудова обмежень бази даних.....	10
5.2 Нормалізація.....	10
6. Реалізація бази даних.....	11
Виконання над відношеннями операцій реляційної алгебри.....	11
Висновок.....	14

1. Аналітичний огляд літератури

Проектування баз даних – процес створення схеми бази даних і визначення необхідних обмежень цілісності.

Основними задачами проектування баз даних є забезпечення зберігання в БД всієї необхідної інформації, забезпечення можливості отримання даних по всім необхідним запитам, скорочення надмірності і дублювання даних, забезпечення цілісності бази даних.

Процес проектування БД являє собою процес переходів від неформального словесного опису інформаційної структури предметної області до формалізованого опису об'єктів предметної області в термінах деякої моделі. Кінцевою метою проектування є побудова проектної БД. Очевидно, що процес проектування складний і тому має сенс розділити його на логічно завершені етапи:

- 1) Інфологічне(концептуальне) проектування – частково формалізований опис об'єктів предметної області в термінах деякої семантичної моделі. Така модель створюється без орієнтації на якусь конкретну СУБД і модель даних. Інфологічна модель повинна включати такий формалізований опис предметної області, який легко буде сприйматися не тільки фахівцями в області БД, а й, наприклад, замовником.

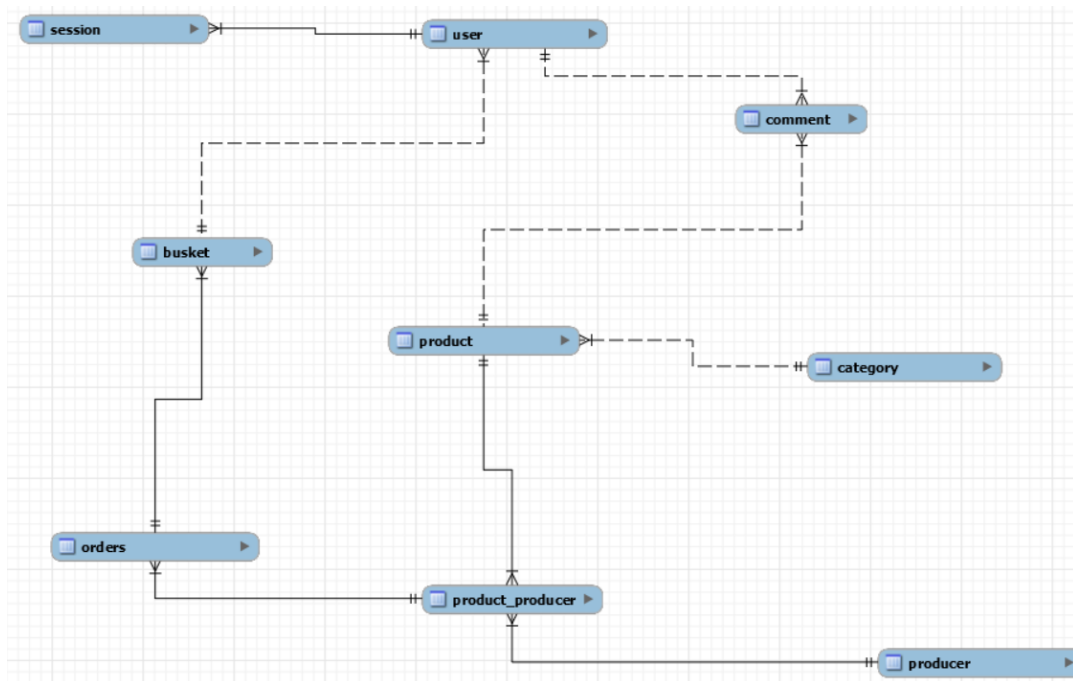


Рис.1.1 Приклад інформаційної схеми

- 2) Логічне проектування – створення схеми бази даних на основі конкретної моделі даних, наприклад реляційної моделі даних. Для реляційної моделі даних даталогічна модель – набір схем відношень, зазвичай із

зазначенням первинних ключів, а також «зв'язків» між відношеннями, що представляють собою зовнішні ключі.

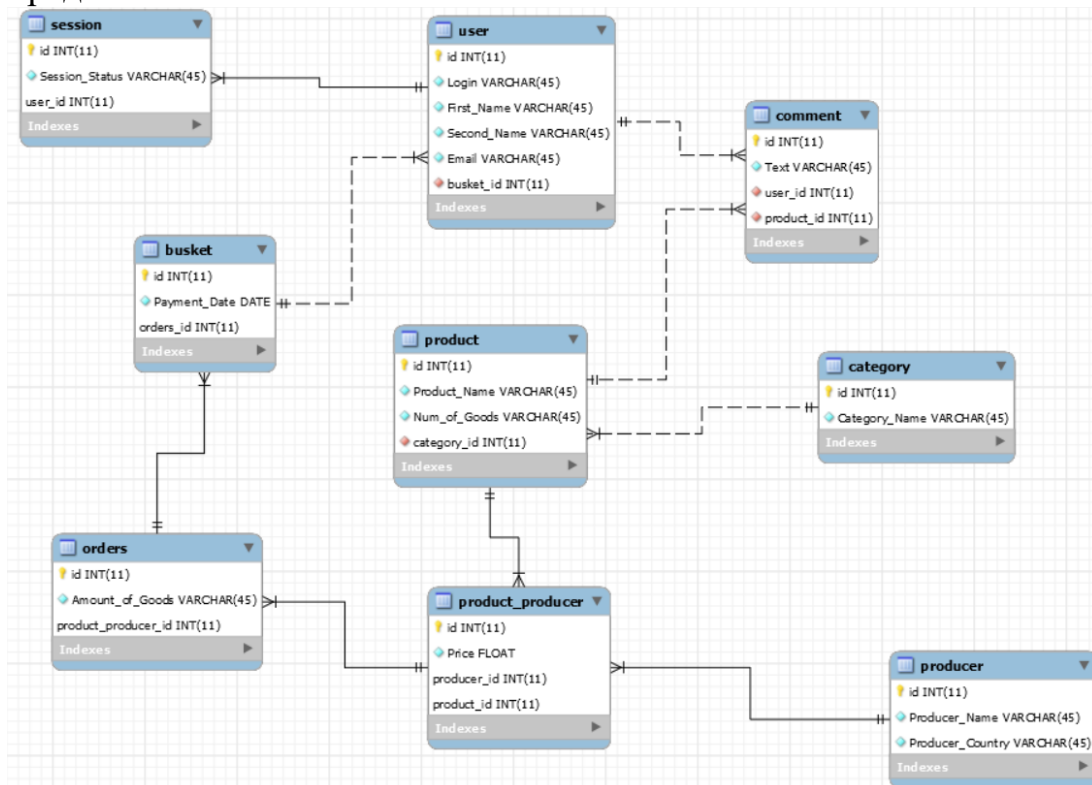
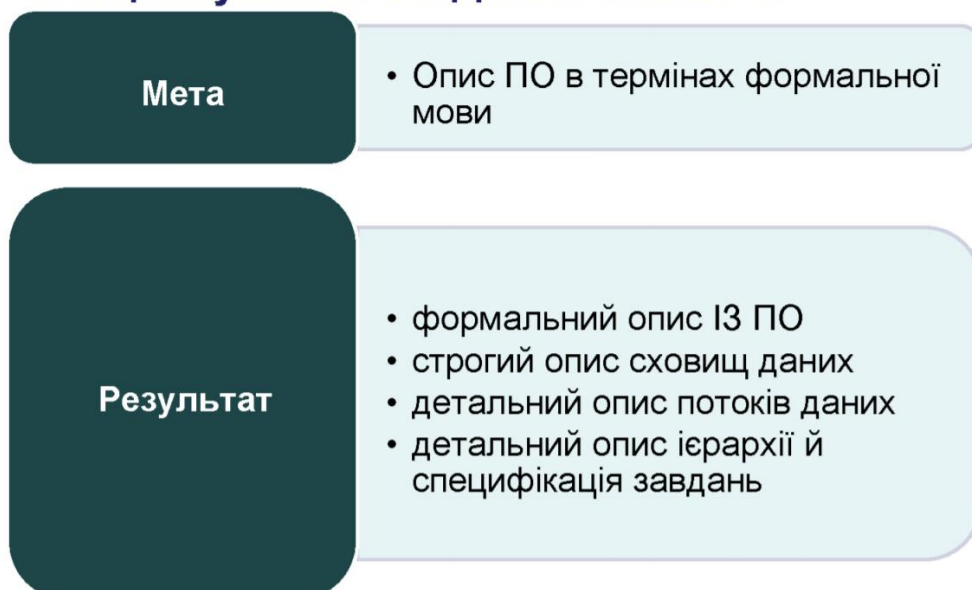


Рис. 1.2 Приклад даталогічної схеми

- 3) Фізичне проектування – створення схеми бази даних для конкретної СУБД. Специфіка конкретної СУБД може містити обмеження на іменування об'єктів бази даних, обмеження на підтримувані типи даних і т.д. Обмеження, наявні в логічній моделі даних, реалізуються різними засобами СУБД, наприклад, за допомогою індексів, декларативних обмежень цілісності, тригерів, процедур. Під схемою баз даних, якщо це реляційна база даних, мають на увазі код MySQL.

Концептуальне моделювання ПО



Логічне та фізичне моделювання ПО



Рис. 1.3 Відмінність рівнів на кожному з етапів проектування

На основі цих знань будемо проектувати базу даних для обраної предметної області – інтернет-магазину.

2. Визначення та опис предметної області.

2.1. Аналіз вимог до баз даних

Однією з основних функцій інтернет-магазинів є продаж товарів різних категорій. Для ефективної роботи інтернет-магазину необхідно спроектувати базу даних, яка повинна містити інформацію про товарів, виробників, види товарів, їх ціну. Таким чином в базі даних повинна зберігатися інформація:

- Про користувача: код користувача, логін, ім'я, прізвище, електронна адреса, код корзини.
- Про сесію користувача: код сесії, стан сесії, код користувача.
- Про коментарії: код коментаря, текст коментаря, код користувача, який написав цей коментар.
- Про товар: код товару, назва товару, кількість товару, код категорії товару.
- Про категорії: код категорії, назва категорії.
- Про виробника: код виробника, назва виробника, назва країни виробника.
- Про замовлення: код замовлення, кількість купленого товару користувачем, код корзини користувача.
- Про корзину: код корзини, дата оплати замовлення, код замовлення.

Користувачів цікавить товари різних категорій, наприклад, телефони, ноутбуки і т.д. Їм потрібна інформація про ці товари, така як – назва товару, його ціна, хто є виробником, детальніша інформація про виробника, чи наявний товар у продажі.

На основі всієї цієї інформації складається специфікація вимог користувачів.

2.2. Створення ділової моделі організації

Функцією інтернет-магазину є деякий суттєвий бізнес-процес, який управляєм своїми ресурсами та здійснює продаж своїх ресурсів з різними користувачами.

Створимо ділову модель Інтернет-магазину

Таблиця 1

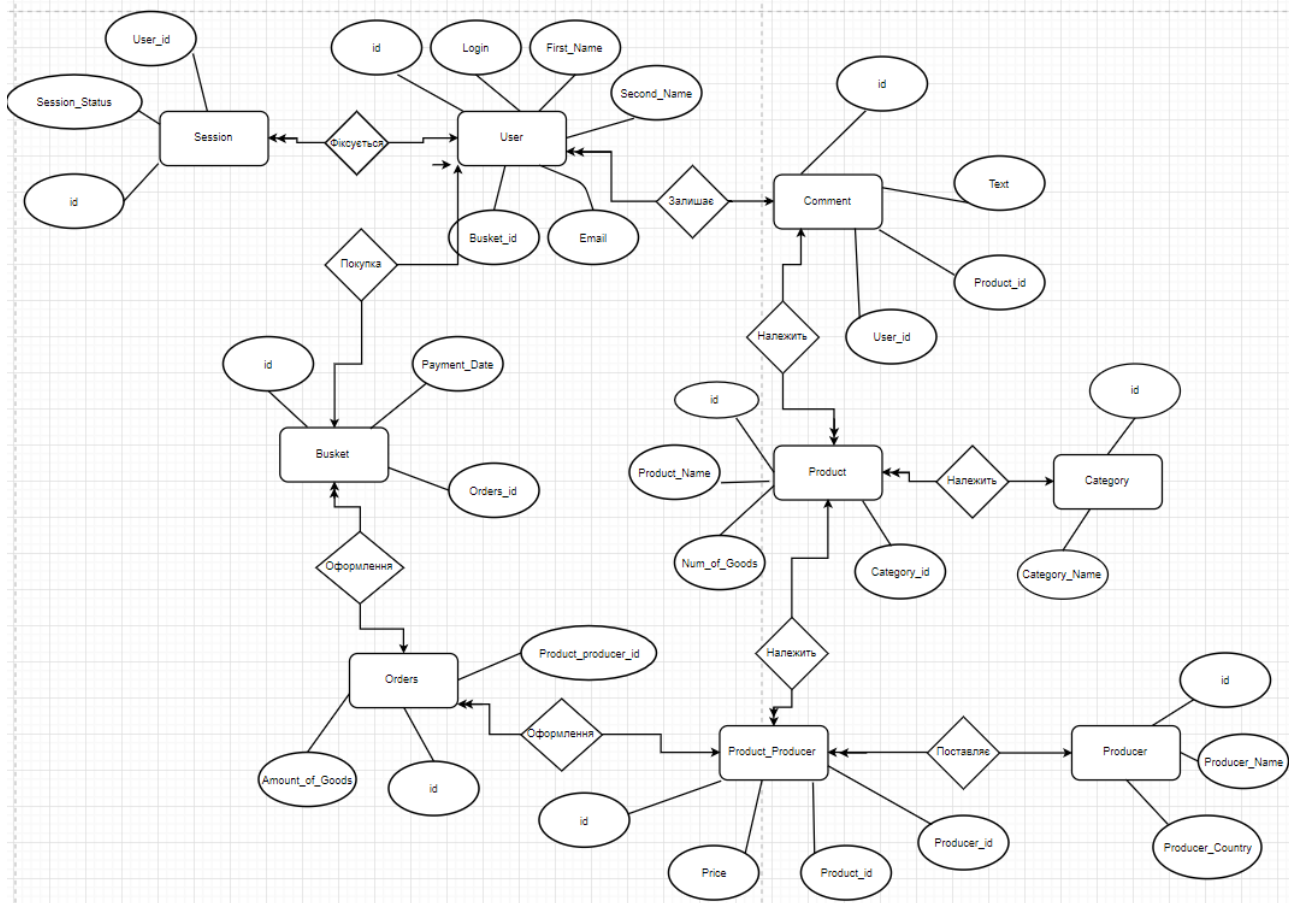
Класи Функції	Сесія	Користувач	Корзина	Замовлення	Продукт	Коментарій	Категорія	Виробник
Стан сесії	*							
Інформація про користувача		*						
Зберігання замовлення			*					
Інформація про покупку			*	*				
Назва продукту та його кількість			*	*	*			
Коментарій		*			*	*		
Клас продукту							*	
Інформація про виробника				*				*

Рис 2.1. Ділова модель організації «Інтернет-магазин»

Зірочки у діловій моделі вказують зв'язки між об'єктами, що забезпечують виконання певної функції.

3. Побудова моделі типу «сутність-зв'язок».

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



4. Побудова логічної структури бази даних.

Для створення бази даних «Інтернет-магазин» у режимі конструктора створимо структури таблиць, встановимо зв'язки між таблицями.

Таблиця 2

Назва таблиці	Ім'я поля	Тип даних
Session	<u>Id</u> Session_Status User_id	Лічильник Текстовий Лічильник (майстер підстановки)
User	<u>Id</u> Login First_Name Second_Name Email	Лічильник Текстовий Текстовий Текстовий Текстовий
Category	<u>Id</u> Category_Name	Лічильник Текстовий
Producer	<u>Id</u>	Лічильник

	Producer_Name Producer_Country	Текстовий Текстовий
Product	<u>Id</u> Product_Name Num_of_Goods Category_id	Лічильник Текстовий Числовий Лічильник (майстер підстановки)
Orders	<u>Id</u> Amount_of_Goods Product_producer_id	Лічильник Числовий Лічильник (майстер підстановки)
Busket	<u>Id</u> Payment_Date Orders_id	Лічильник Дата Лічильник (майстер підстановки)
Comment	<u>Id</u> Text User_id Product_id	Лічильник Текстовий Лічильник (майстер підстановки) Лічильник (майстер підстановки)
Product_Producer	<u>id</u> Price Producer_id Product_id	Лічильник Числовий Лічильник (майстер підстановки) Лічильник (майстер підстановки)

Під час опису структури таблиці визначав первинний ключ.(Підкреслені)

5.Побудова обмежень відношень бази даних. Нормалізація відношень бази даних.

5.1. Побудова обмежень бази даних

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

Всі відношення 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

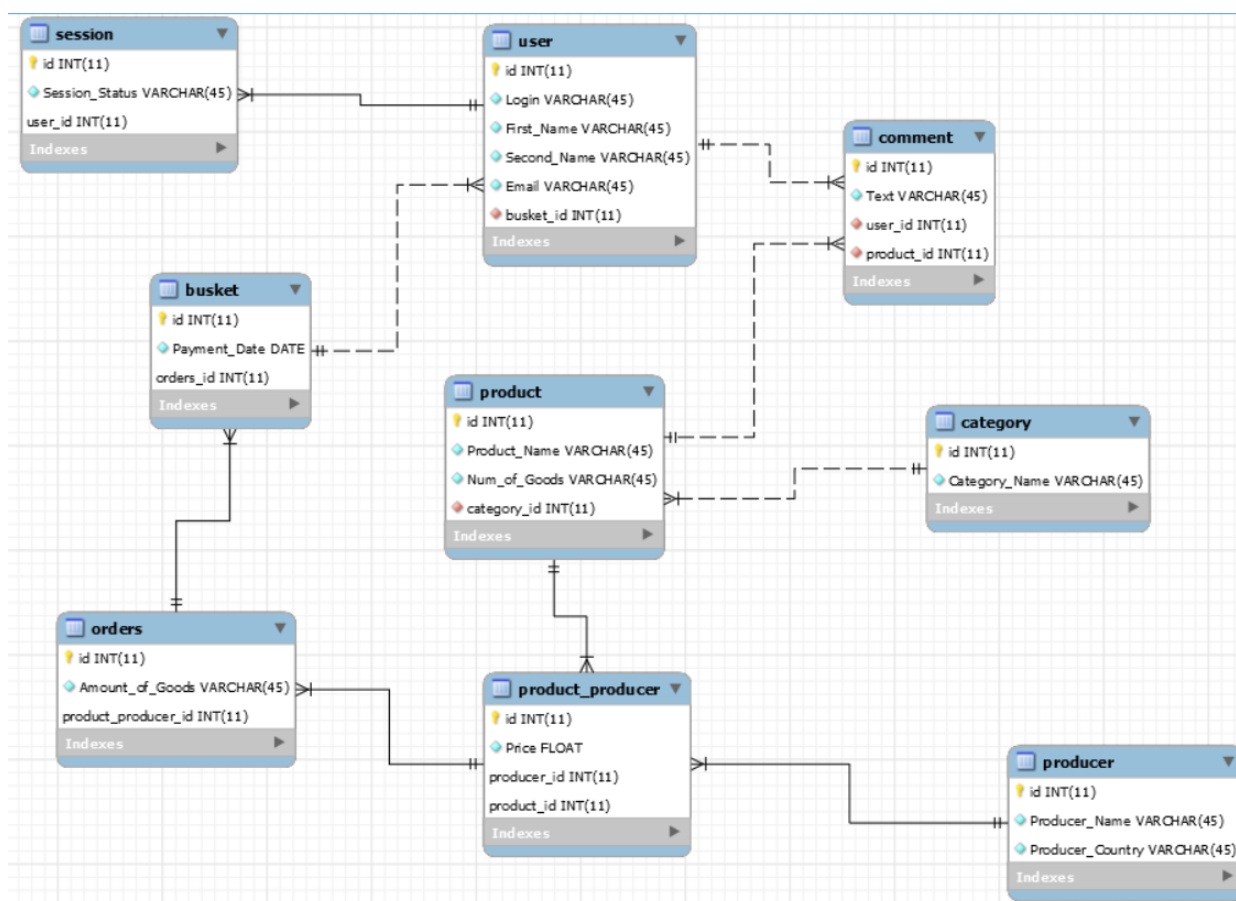


Рис 5.1. Схема БД «Інтернет-магазин»

База даних відноситься до другої нормальної форми, оскільки:

- Кожна таблиця має основний ключ: мінімальний набір колонок, які ідентифікують запис;
- дані, що повторно з'являються в декількох рядках, винесені в окремі таблиці (в нашому випадку – це код продукту та код виробника).
- уникаються повторення груп (категорії даних, що можуть зустрічатися різну кількість разів у різних записах) правильно визначаючи ключові атрибути;

6. Виконання над відношеннями операцій реляційної алгебри.

- 1) Створив запит, який виводить id користувача, його логін та ім'я, номер його корзини і дату оплати замовлення. Однією з умов запиту було те, що потрібно було зробити їх об'єднання та вивести лише тих користувачів номер яких менший за цифру «5». Для цього було створено дві **проекції** user1 та basket1 та було використано команду **Select**:

```
SELECT user1.id, user1.Login, user1.First_Name,  
basket1.id, basket1.Payment_Date  
FROM user1, basket1  
WHERE user1.id = basket1.id AND user1.id < 5 AND basket1.id < 5;
```

```
mysql> create view user1 as select distinct id, Login, First_Name  
-> from user;  
Query OK, 0 rows affected (0.10 sec)  
mysql> create view basket1 as select distinct id, Payment_Date  
-> from basket;  
Query OK, 0 rows affected (0.17 sec)  
mysql> select user1.id, user1.Login, user1.First_Name,  
-> basket1.id, basket1.Payment_Date  
-> from user1, basket1  
-> where user1.id = basket1.id and user1.id < 5 and basket1.id < 5;  
+-----+-----+-----+-----+-----+  
| id | Login   | First_Name | id | Payment_Date |  
+-----+-----+-----+-----+-----+  
| 1 | Bender  | John       | 1 | 2005-06-20   |  
| 2 | Wanderer | Anthony    | 2 | 2001-07-20   |  
| 3 | Fray    | Ivan       | 3 | 2005-05-20   |  
| 4 | Darker  | Anastasia  | 4 | 2012-09-20   |  
+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

Рис 6.1. Результат виконання запиту користувачів, в яких значення коду користувачів менше за 5.

- 2) Створив запит, котрий виводить номер користувача, його логін та електронну пошту, а також текст його коментаря і номер цього самого коментаря. Для цього було використано **Left Join**:

```
SELECT user.id, user.Login, user.Email, comment.id, comment.Text  
FROM user LEFT JOIN comment ON  
user.id = comment.id;
```

```
mysql> select user.id, user.Login, user.Email, comment.id, comment.Text
-> from user left join comment on
-> user.id = comment.id;
```

id	Login	Email	id	Text
1	Bender	bender@gmail.com	1	Good product
2	Wanderer	wanderer@gmail.com	2	Nice
3	Fray	fray@gmail.com	3	I have a lot problems
4	Darker	darker@gmail.com	4	Okay, looks good
5	Vampire	vampire@gmail.com	5	Not bad
6	Ymir	ymir@gmail.com	6	I don't like it

6 rows in set (0.00 sec)

Рис 6.2. Результат виконання запиту користувачів, які написали коментарі.

- 3) Виконав запит, який визначив, котрі продукти взагалі не купували. Для цього використав **Select**:

```
SELECT product.Product_Name FROM product
WHERE not exists
(SELECT * FROM product_producer WHERE product_producer.id =
product.id);
```

```
mysql> select product.Product_Name from product
-> where not exists
-> (select * from product_producer where product_producer.id = product.id);
```

Product_Name
HP
Samsung

2 rows in set (0.00 sec)

Рис 6.3. Результат виконання запиту, який виводить назви продуктів, які ніхто не купував.

- 4) Виконав запит, який вивів номер користувача, його логін та його ім'я. Умова запиту полягала в тому, що потрібно було вивести лише тих користувачів в іменах котрих присутня буква «і». Для цього було використано **Select** та **LIKE**:

```
SELECT id, Login, First_Name FROM user WHERE First_Name LIKE
'%i%'
ORDER BY First_Name ASC;
```

```
mysql> select id, Login, First_Name from user where First_Name LIKE '%i%'
-> order by First_Name ASC;
+----+-----+-----+
| id | Login | First_Name |
+----+-----+-----+
| 6 | Ymir | Anakin |
| 4 | Darker | Anastasia |
| 3 | Fray | Ivan |
+----+-----+-----+
3 rows in set (0.00 sec)
```

Рис 6.4. Результат виконання запиту, який виводить лише тих користувачів в імені яких присутня буква «і».

- 5) Виконав запит на виконання декартового добутку двох проекцій, а саме user1 та basket1:

```
SELECT * FROM user1, basket1;
```

```
mysql> select * from user1, basket1;
+----+-----+-----+----+-----+
| id | Login | First_Name | id | Payment_Date |
+----+-----+-----+----+-----+
| 1 | Bender | John | 1 | 2005-06-20 |
| 2 | Wanderer | Anthony | 1 | 2005-06-20 |
| 3 | Fray | Ivan | 1 | 2005-06-20 |
| 4 | Darker | Anastasia | 1 | 2005-06-20 |
| 5 | Vampire | Ben | 1 | 2005-06-20 |
| 6 | Ymir | Anakin | 1 | 2005-06-20 |
| 1 | Bender | John | 2 | 2001-07-20 |
| 2 | Wanderer | Anthony | 2 | 2001-07-20 |
| 3 | Fray | Ivan | 2 | 2001-07-20 |
| 4 | Darker | Anastasia | 2 | 2001-07-20 |
| 5 | Vampire | Ben | 2 | 2001-07-20 |
| 6 | Ymir | Anakin | 2 | 2001-07-20 |
| 1 | Bender | John | 3 | 2005-05-20 |
| 2 | Wanderer | Anthony | 3 | 2005-05-20 |
| 3 | Fray | Ivan | 3 | 2005-05-20 |
| 4 | Darker | Anastasia | 3 | 2005-05-20 |
| 5 | Vampire | Ben | 3 | 2005-05-20 |
| 6 | Ymir | Anakin | 3 | 2005-05-20 |
| 1 | Bender | John | 4 | 2012-09-20 |
| 2 | Wanderer | Anthony | 4 | 2012-09-20 |
| 3 | Fray | Ivan | 4 | 2012-09-20 |
| 4 | Darker | Anastasia | 4 | 2012-09-20 |
| 5 | Vampire | Ben | 4 | 2012-09-20 |
| 6 | Ymir | Anakin | 4 | 2012-09-20 |
| 1 | Bender | John | 5 | 2003-03-20 |
| 2 | Wanderer | Anthony | 5 | 2003-03-20 |
| 3 | Fray | Ivan | 5 | 2003-03-20 |
| 4 | Darker | Anastasia | 5 | 2003-03-20 |
| 5 | Vampire | Ben | 5 | 2003-03-20 |
| 6 | Ymir | Anakin | 5 | 2003-03-20 |
| 1 | Bender | John | 6 | 2001-02-20 |
| 2 | Wanderer | Anthony | 6 | 2001-02-20 |
| 3 | Fray | Ivan | 6 | 2001-02-20 |
| 4 | Darker | Anastasia | 6 | 2001-02-20 |
| 5 | Vampire | Ben | 6 | 2001-02-20 |
| 6 | Ymir | Anakin | 6 | 2001-02-20 |
+----+-----+-----+----+-----+
36 rows in set (0.00 sec)
```

Рис 6.5. Результат виконання декартового добутку

- 6) Запит на виконання натурального з'єднання у проекціях user1 та basket1:

```
SELECT user1.id, user1.Login, user1.First_Name, basket1.id,  
basket1.Payment_Date  
FROM user1, basket1  
WHERE user1.id = basket1.id;
```

```
mysql> select user1.id, user1.Login, user1.First_Name, basket1.id, basket1.Payment_Date  
-> from user1, basket1  
-> where user1.id = basket1.id;
```

id	Login	First_Name	id	Payment_Date
1	Bender	John	1	2005-06-20
2	Wanderer	Anthony	2	2001-07-20
3	Fray	Ivan	3	2005-05-20
4	Darker	Anastasia	4	2012-09-20
5	Vampire	Ben	5	2003-03-20
6	Ymir	Anakin	6	2001-02-20

```
6 rows in set (0.00 sec)
```

Рис 6.6. Результат виконання натурального з'єднання

Висновок:

Після виконання даної розрахункової роботи я оволодів навичками побудови концептуальних моделей «сутність-зв'язок» та створенням бази даних. Також навчився накладати на обмеження і нормалізувати бази даних та створювати логічну структуру бази даних. Також після створення бази даних, я відповідно до своєї предметної області, навчився виконувати над відношеннями різні операції реляційної алгебри.