



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

## КУРСОВА РОБОТА

з дисципліни «Бази даних»

(назва дисципліни)

на тему: База даних для підтримки діяльності компанії, що займається  
морськими вантажними перевезеннями

Студента 2 курсу групи ІТ-02  
спеціальності 121 «Інженерія програмного  
забезпечення»

Котенко Д. О.

(прізвище та ініціали)

Керівник:

доцент Ліщук К. І.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала \_\_\_\_\_

Кількість балів: \_\_\_\_\_ Оцінка ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_

---

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

---

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

---

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

Київ – 2021 рік

**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет Інформатики та обчислювальної техніки  
(повна назва)

Кафедра Інформатики та програмної інженерії  
(повна назва)

Дисципліна Бази даних

Курс   2  

Група   ІТ-02  

Семестр   3  

**З А В Д А Н Н Я  
НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Котенко Дмитро Олександрович

(прізвище, ім'я, по батькові)

1. Тема роботи База даних підтримки діяльності компанії, що  
займається морськими вантажними перевезеннями

керівник роботи Ліщук Катерина Ігорівна, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

2. Строк подання студентом роботи 26.12.2021

3. Вихідні дані до роботи лекційний матеріал, технічне завдання для  
курсової роботи

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно  
розробити):

- 1) Аналіз предметної області
- 2) побудова ER-моделі
- 3) побудова реляційної схеми з ER-моделі
- 4) створення бази даних
- 5) створення та використання збережених тригерів
- 6) імпорт даних в створену базу даних
- 7) створення користувачів бази даних
- 8) створення представлень для запитів
- 9) запити для визначених потреб користувачів
- 10) створення та використання збережених процедур та генераторів
- 11) оптимізація роботи бази даних

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
Розроблена ER-модель, розроблена реляційна схема бази даних

6. Дата видачі завдання 31.10.2021

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання курсового проекту	Строк виконання етапів проекту	Примітка
1	Аналіз предметної області	До 05.11.2021	
2	Побудова ER-моделі	До 12.11.2021	
3	Побудова реляційної схеми з ER-моделі	До 12.11.2021	
4	Створення бази даних	До 25.11.2021	
5	Створення та використання тригерів	До 30.11.2021	
6	Створення користувачів бази даних	До 01.11.2021	
7	Створення представлень для запитів	До 05.12.2021	
8	Запити для визначених потреб користувачів	До 08.12.2021	
9	Запити для визначених потреб користувачів	До 12.12.2021	
10	Створення та використання збережених процедур та генераторів	До 17.12.2021	
11	Оптимізація роботи бази даних	До 20.12.2021	
12	Написання звіту	До 23.12.2021	

Студент

\_\_\_\_\_ Котенко Д.О.  
(підпис) (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ Ліщук К. І.  
(підпис) (прізвище та ініціали)

## **ВСТУП**

### **ТЕОРЕТИЧНА ЧАСТИНА**

1. Аналіз предметної області
  - 1.1. Опис предметної області
  - 1.2. Опис вхідних даних
  - 1.3. Опис вихідних даних
2. Проектування бази даних
  - 2.1. Інфологічна модель бази даних
    - 2.1.1. Опис сутностей
    - 2.1.2. Опис атрибутів
    - 2.1.3. Опис зв'язків
    - 2.1.4. ER-діаграма
  - 2.2. Нормалізація таблиць при проектуванні

### **ПРАКТИЧНА ЧАСТИНА**

1. Створення бази даних за допомогою MySQL
  - 1.1. Створення бази даних
  - 1.2. Створення таблиць бази даних з генераторами
  - 1.3. Створення діаграми бази даних
  - 1.4. Створення користувачів
  - 1.5. Заповнення бази даних даними
  - 1.6. Створення представлень
  - 1.7. Створення процедур та функцій
  - 1.8. Створення тригерів
  - 1.9. Написання DLM запитів
  - 1.10. Оптимізація та її результати

## **ВИСНОВКИ**

## **СПИСОК ЛІТЕРАТУРИ**

## **ДОДАТКИ**

Додаток А. Створення таблиць бази даних

Додаток Б. Створення процедур та функцій

Додаток В. Написання DLM запитів

## ВСТУП

Будь-яка БД є частиною інформаційної системи і призначена для збору, зберігання і маніпулювання інформацією.

У сучасному світі навіть звичайний магазин не обходиться без бази даних. Годі тоді й говорити про величезні міжнародні корпорації.

Особливо важливо мати грамотно спроектувати базу даних для компаній, що займаються морськими перевезеннями, оскільки хоч якась маленька помилка може спричинити до багатомільярдний збитків.

Можемо згадати події 26 березня цього року, коли корабель “Евер Гивен” сів на дно прямо посеред Суецького каналу і провів там аж 12 днів, закривши одну з найбільших в світі торгових артерій. Дана аварія була оцінена в 230 млрд доларів.

Як пізніше виявилось, в базі даних судової компанії виникла помилка і в результаті неї на корабель було погружена надлишковий вантаж, що і стало причиною багато мільярдних збитків.

Тож в даній курсовій роботі я хочу створити досконалу модель бази даних для підтримки діяльності компаній, що займаються морськими перевозками.

## ТЕОРЕТИЧНА ЧАСТИНА

### 1. Аналіз предметної області

#### 1.1. Опис предметної області

Предметна область - частина реального світу, що описує інформаційна система в залежності від її призначення.

В даний час усі організації повинні мати доступ до інформації. Це дуже важливо, тому бази даних, які задовольняють потреби зі зберіганням й управлінням даних, мають неабияку цінність, бо допомагають людям в цій справі.

При розробці реляційної бази даних для підтримки діяльності компанії, що займається морськими вантажними перевезеннями. було проведено дослідження предметної області.

Мною було виявлено, що для реалізації даної теми в базі даних повинно бути дві сторони, а сама компанія перевізник та компанія, яка купує послуги перевезення.

Компанія перевізник повинна розташовуватись в певній країні та мати n-ну кількість кораблів та співробітників. В свою чергу грузові кораблі можуть бути різних типів за вантажопідйомність. А працівники можуть обіймати одну з зазначених в базі даних посад.

Компанія ж, яка купує послуги перевезення може створювати контракти та вказувати в ньому груз, який потрібно перевезти. А в свою чергу компанія перевізник, при бажанні може додати в контракт свій корабель.

Для реалізації всього вищесказаного база даних повинна містити точки призначення кораблів, які можна обрати з наявних в БД портів.

## 1.2. Опис вхідних даних

Знаючи свою предметну область нам потрібно ввести дані, що призначені для введення, щоб наша база даних була потрібною. При розробці реляційної бази даних були виділені наступні вхідні дані:

- інформація про країни
- інформація про компанії перевізники
- інформація про типи кораблів
- інформація про кораблі
- інформація про посади
- інформація про працівників
- інформація про порти
- інформація про компанії клієнти
- інформація про контракти
- інформація про точки призначення
- інформація про продукцію для перевезення

## 1.3. Опис вихідних даних

Вихідні дані - повідомлення і результати, які видаються самою системою. Беруться з постійних даних.

Вихідними даними для бази даних є вихідні запити (табл. 1.1 – 1.11):

Таблиця 1.1 – Довідник Країн (Countries)

Назва поля	Тип поля	Опис поля
Code	VARCHAR	Code країни
Name	VARCHAR	Назва країни

Таблиця 1.2 – Довідник Компаній перевізників (FreightCompanies)

Назва поля	Тип поля	Опис поля
Id	INT	Id компанії
CountryCode	VARCHAR	Code країни
Name	VARCHAR	Назва компанії перевізника
PhoneNumber	VARCHAR	Номер телефон
Email	VARCHAR	Пошта

Таблиця 1.3 – Довідник Типів кораблів (ShipTypes)

Назва поля	Тип поля	Опис поля
Id	INT	Id
Name	VARCHAR	Назва корабля
MaxWeight	INT	Максимальна вага

Таблиця 1.4 – Довідник Кораблів (Ships)

Назва поля	Тип поля	Опис поля
Id	INT	Id
ShipTypeId	INT	Id типу корабля
FreightCompanyId	INT	Id компанії перевізника
Name	VARCHAR	Назва корабля

Таблиця 1.5 – Довідник Посад (Positions)

Назва поля	Тип поля	Опис поля
Id	INT	Id
Name	VARCHAR	Назва посади



MinSalary	INT	Мінімальна зарплата
-----------	-----	---------------------

Таблиця 1.6 – Довідник Працівників (Workers)

Назва поля	Тип поля	Опис поля
Id	INT	Id працівника
PositionId	INT	Id посади
ShipId	INT	Id корабля
FullName	VARCHAR	ПІБ
PhoneNumber	VARCHAR	Номер телефон
Salary	INT	Зарплата

Таблиця 1.7 – Довідник Портів (Ports)

Назва поля	Тип поля	Опис поля
Id	INT	Id
CountryCode	VARCHAR	Code країни
Name	VARCHAR	Назва порту

Таблиця 1.8 – Довідник Компаній замовників / Клієнтів (Clients)

Назва поля	Тип поля	Опис поля
Id	INT	Id
Name	VARCHAR	Назва компанії замовника
PhoneNumber	VARCHAR	Номер телефон
Email	VARCHAR	Пошта

Таблиця 1.9 – Довідник Контрактів (Contracts)

Назва поля	Тип поля	Опис поля
Id	INT	Id
ShipId	INT	Id корабля
ClientId	INT	Id компанії замовника
SigningDate	DATE	дата підпису контракта

Таблиця 1.10 – Довідник Пунктів призначення(Destinations)

Назва поля	Тип поля	Опис поля
Id	INT	Id
PortId	INT	Id порту
ContractId	INT	Id контракту

Таблиця 1.11 – Довідник Продукції для перевезення(Products)

Назва поля	Тип поля	Опис поля
Id	INT	Id
ContractId	INT	Id контракту
Name	VARCHAR	Назва вантажу
Weight	INT	Вага вантажу
Price	INT	Ціна перевезення

## 2. Проектування бази даних

### 2.1. Інфологічна модель бази даних

Проектувальники інфологічної моделі розглядають як модель сутність - зв'язок”.

Призначення інфографічної моделі полягає у наданні найбільш дійсних способів збору та представлення інформації, що зберігається у

базі даних. Ось чому інфологічну модель даних пробують будувати по аналогії з дійсною мовою.

Початковим етап в проектуванні баз даних - це інфологічна модель, яка допомагає в побудові таблиць та зв'язків між ними.

Існують різні способи опису інфологічної моделі, однак на даний момент найбільш популярним з підходів є той, що заснований на діаграмі "сутність-зв'язок". Отже, при створенні інфологічної моделі ми будемо використовувати ER – діаграму. Ця інфологічна модель - це модель предметної області, яка визначає сутності та зв'язки бази даних. Побудова такої моделі є індивідуальною і особливою, тому однієї методики створення інфологічної моделі немає, кожен робить так як хоче, так як саме він бачить цю модель предметної області.

Інфологічна модель – це опис бази даних, яка буде існувати в майбутньому, опис бд поданий за допомогою діаграм, таблиць, дійсної мови. Будувannya цієї 13 моделі є продовженням аналізу предметної області інфологічна модель є представленням БД з точки зору того, хто розробляє її.

Головними конструктивними елементами інфологічної моделі даних є: сутності, їх атрибутів та типи зв'язків.

### 2.1.1. Опис сутностей

Одним із елементом інфологічної моделі "сутність-зв'язок" є сутності.

Сутність - це те, про що накопичується інформація в інформаційній системі і що може бути однозначно унікальне ідентифіковане. При цьому ім'я сутності повинно відображати клас об'єкта або тип об'єкта.

У відповідності з описом предметної області були отримано такі сутності:

- “Країна” — інформація про країни
- “Компанія перевізник” — інформація про компанії перевізники
- “Тип корабля” — інформація про типи кораблів
- “Корабель” — інформація про кораблі
- “Посада” — інформація про посади
- “Працівник” — інформація про працівників
- “Порт” — інформація про порти
- “Клієнт” — інформація про компанії клієнти
- “Контракт” — інформація про контракти
- “Точка призначення” — інформація про точки призначення
- “Продукція” — інформація про продукцію для перевезення

#### 2.1.2. Опис атрибутів

Атрибут - це поіменована характеристика сутності, за допомогою якої моделюється її властивість. Атрибути називають ще інформаційними елементами. У відповідності з описом предметної області були виділені наступні атрибути у кожній сутності:

##### Країни

Code
Name

##### Компанії перевізники

Id
CountryCode
Name

PhoneNumber
Email

## Типи кораблів

Id
Name
MaxWeight

## Кораблі

Id
ShipTypeId
FreightCompanyId
Name

## Посади

Id
Name
MinSalary

## Працівники

Id
PositionId
ShipId
FullName
PhoneNumber
Salary

## Порти

Id
CountryCode

Name
------

## Клієнти

Id
Name
PhoneNumber
Email

## Контракти

Id
ShipId
ClientId
SigningDate

## Точки призначення

Id
PortId
ContractId

## Продукція для перевезення

Id
ContractId
Name
Weight
Price

### 2.1.3. Опис зв'язків

Дві сутності можуть пов'язуватися через зв'язок екземплярів однієї сутності з екземплярами іншої сутності.

Основна вимога бази даних - вміти знаходити одну сутність за

значеннями інших, для чого необхідно встановити зв'язок між ними. Так як часто в базах даних створюють більше 50 сутностей, то між цими сутністями може бути дуже багато зв'язків. Складність інфологічної моделі визначається наявністю великої кількості зв'язків. За допомогою зв'язків показують відношення між сутністю та атрибутами. Зв'язкам можна надати імена для зручності. Термін зв'язок використовується для позначення типу зв'язку і може відображати одноразовість чи багаторазовість.

За направленістю розрізняють такі зв'язки:

1) Однонаправлений зв'язок поділяється на: о однозначний, якщо один екземпляр однієї сутності відповідає точно одному екземпляру іншої сутності, зворотного зв'язку не надається; 17 о багатозначний, якщо екземпляр сутності відповідає декільком екземплярам іншої сутності, зворотного зв'язку теж не надається.

2) Двонаправлений зв'язок поділяється на:

однозначний "один до одного" (1:1) одному екземпляру сутності однієї відповідає лише одному екземпляру другої сутності (навпаки теж працює);

двозначні: "один до багатьох" (1: N) один екземпляр сутності відповідає декільком екземплярам іншої сутності, а кілька екземплярів іншої сутності відповідають лише одному екземпляру сутності "багато до багатьох" (M: N) певна кількість екземплярів однієї сутності відповідає певній кількості екземплярів іншої сутності (навпаки теж працює).

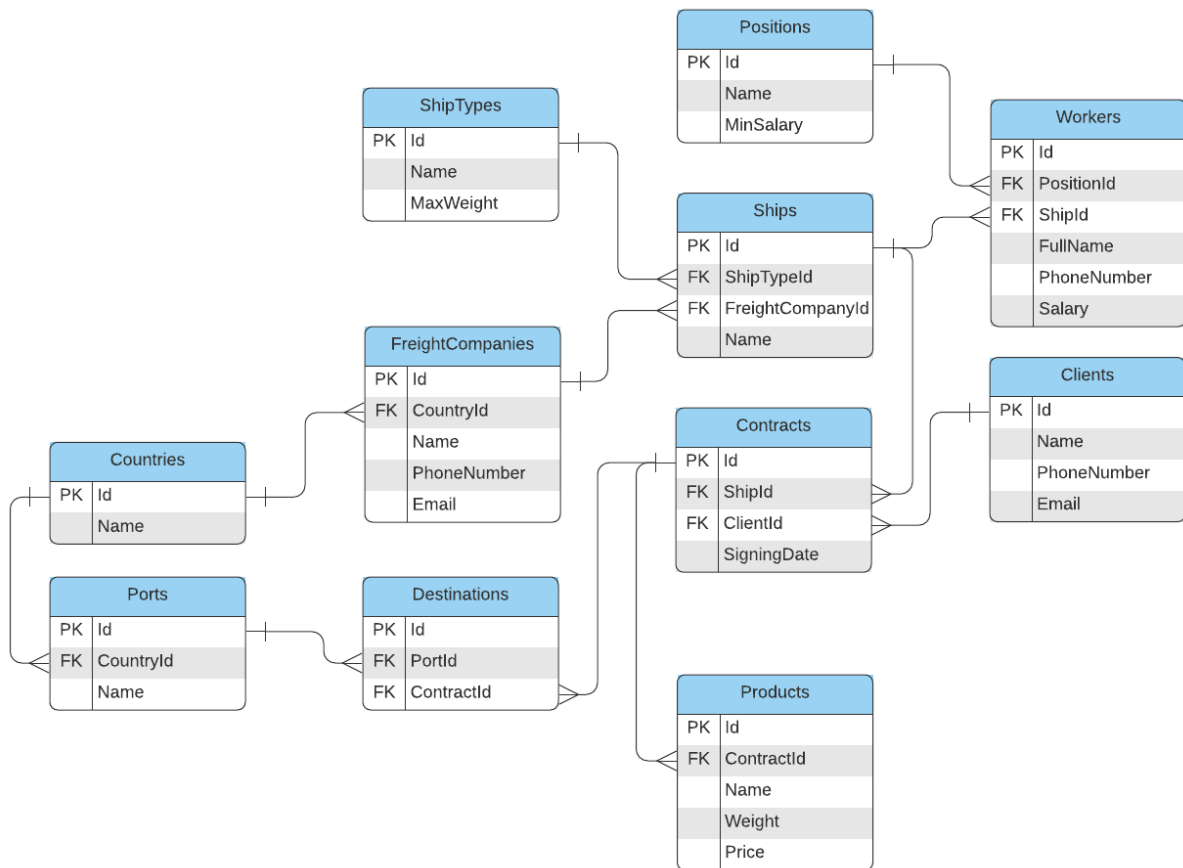
В курсовій роботі були використані наступні типи зв'язків між таблицями:

№ зв'язку	Головна таблиця	Дочірня таблиця	Тип зв'язку
1	Countries	FreightCompanies	1:M
2	Countries	Ports	1:M
3	FreightCompanies	Ships	1:M
4	ShipTypes	Ships	1:M
5	Ships	Contracts	1:M
6	Ships	Contracts	1:M
7	Positions	Workers	1:M
8	Ports	Destinations	1:M
9	Contracts	Destinations	1:M
10	Contracts	Products	1:M
11	Clients	Contracts	1:M

#### 2.1.4. ER-діаграма

На рисунку представлена ER-діаграма бази даних, на якій відображені всі сутності, їх атрибути та зв'язки між сутностями:





## 2.2. Нормалізація таблиць при проектуванні

Нормалізація – це формальна процедура, у ході якої атрибути даних групуються в таблиці, а таблиці групуються в базу даних. Нормалізація бази даних – це одна з важливих вимог БД. Нормалізація потрібна для усунення надмірності даних, тобто видалення повторення даних в різних рядках однієї таблиці.

Перший крок до нормалізації бази даних - це позбавлення від повторень.

Другий крок до нормалізації бази даних – всі поля повинні бути неділімими. Тобто треба розділити усі поля, які можна, щоб в подальшому можна було легше шукати потрібну інформацію і працювати з БД.

Третій крок до нормалізації БД – не повинно бути полів, які можна знайти за допомогою інших полів.

Четвертий крок до нормалізації БД – не потрібні поля в БД, які

позначають різні види одного і того ж.

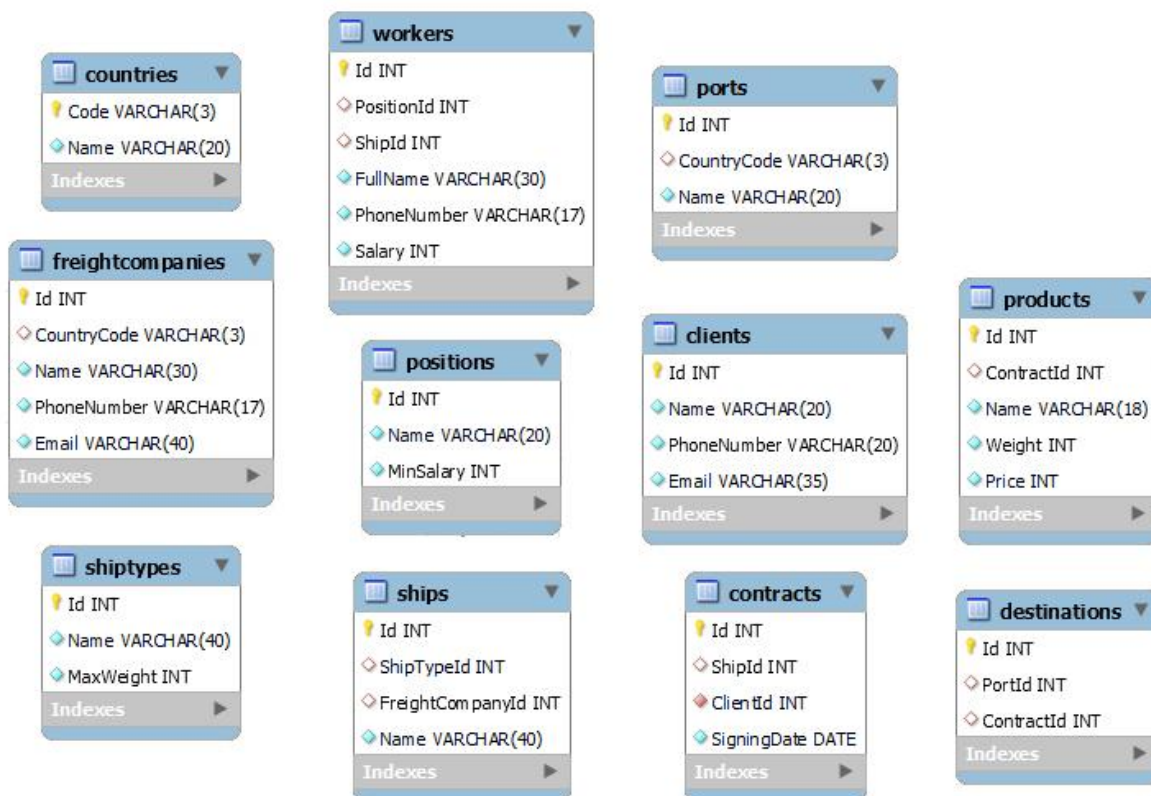
Кожна наступна нормальна форма має кращі властивості, ніж попередня форма. У теорії реляційних БД виділяють п'ять нормальних форми і нормальну форму Бойса-Кодда. Ми будемо розглядати тільки три нормальні форми.

§ Перша нормальна форма вимагає, щоб кожне поле таблиці бази даних було неділимим і не містить повторень.

§ Друга нормальна форма вимагає, щоб всі атрибути таблиці залежали від первинного ключа.

§ Третя нормальна форма вимагає, щоб в таблиці не малося транзитивних залежностей між атрибутами таблиці.

Вигляд після нормалізації таблиць:



## ПРАКТИЧНА ЧАСТИНА

Весь код курсової роботи можна переглянути за посиланням:

<https://github.com/kotenko2002/Shipping-database>

### 1. Створення бази даних за допомогою MySQL

#### 1.1. Створення бази даних

```
CREATE DATABASE shipping;
```

#### 1.2. Створення таблиць бази даних з генераторами

В базі даних *shipping* створюємо наступні таблиці:

- Країни (Countries)
- Компанії перевізники (FreightCompanies)
- Типи кораблів (ShipTypes)
- Кораблі (Ships)
- Посади (Positions)
- Працівники (Workers)
- Порти (Ports)
- Компанії замовники / Клієнти (Clients)
- Контракти (Contracts)
- Пункти призначення (Destinations)
- Продукція для перевезення (Products)

за допомогою команд:

```
-- створення таблиці Країни (Countries)
```

```
CREATE TABLE Countries (  
    Code CHAR(3) PRIMARY KEY,  
    Name VARCHAR(20) NOT NULL  
);
```

```
-- створення таблиці Компанії перевізники (FreightCompanies)
```

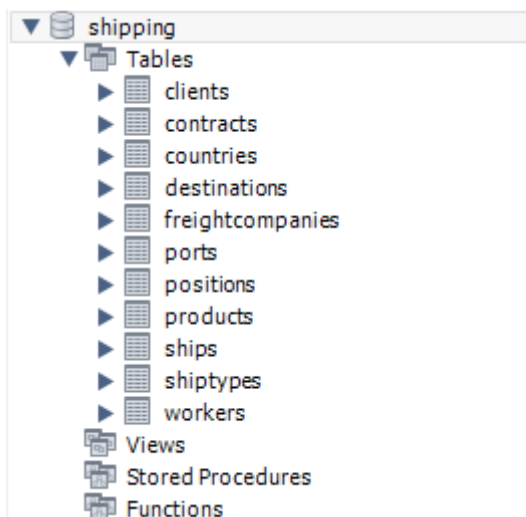
```
CREATE TABLE FreightCompanies (  
    Id INT PRIMARY KEY AUTO_INCREMENT,  
    CountryCode CHAR(3),
```

```

    Name VARCHAR(30) NOT NULL,
    PhoneNumber VARCHAR(17) NOT NULL,
    Email VARCHAR(40) NOT NULL,
    FOREIGN KEY (CountryCode)
        REFERENCES Countries (Code)
);

-- створення таблиці Типи кораблів (ShipTypes)
CREATE TABLE ShipTypes (
    Id INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(40) NOT NULL,
    MaxWeight INT NOT NULL
);
...

```

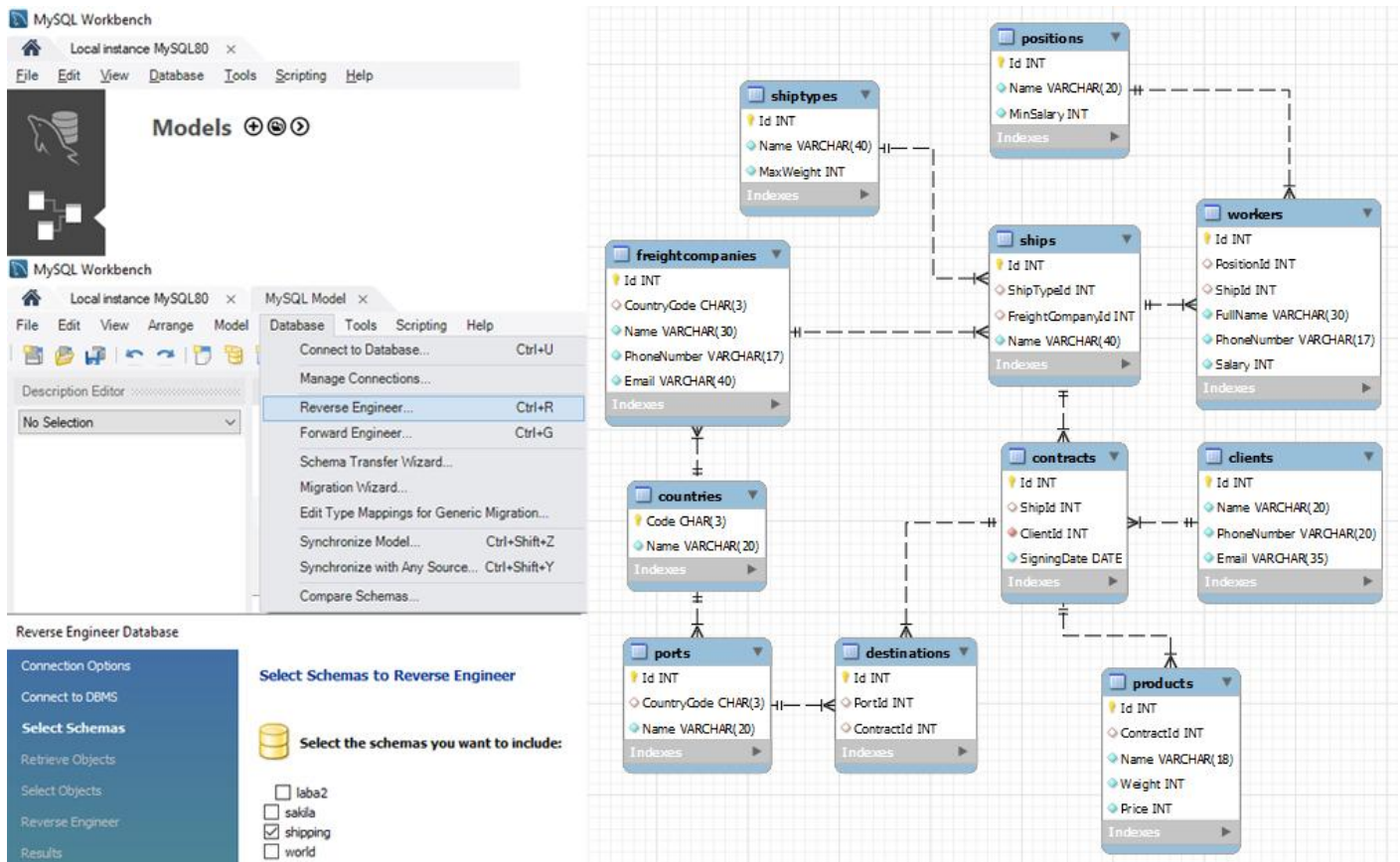


Всі інші таблиці можна переглянути в “Додатку А” або за посиланням:

<https://github.com/kotenko2002/Shipping-database/blob/master/CreateDB.sql>

### 1.3. Створення діаграми бази даних

За допомогою вбудованих в MySQL WorkBench функцій створюємо діаграму бази даних *shipping*



#### 1.4. Створення користувачів

В базі даних *shipping* створюємо наступних користувачів:

- Адміністратор (Admin) - має всі можливі права. Слідкує за цілісністю бази даних, за потреби може вносити перші правки.
- Транспортний менеджер (TransportationManager) - має права додавати значення в таблиці: Кораблі (Ships), Працівники (Workers) та переглядати значення в таблицях: Кораблі (Ships), Працівники (Workers), Компанії перевізники (FreightCompanies), Країни (Countries), Типи кораблів (ShipTypes), Посади (Positions). Слідкує за діяльністю компаній перевізників і у разі потреби додає нові кораблі та працівників до бази даних.
- Менеджер по замовленням (OrderManager) - має права додавати значення в таблиці: Контракти (Contracts), Пункти призначення (Destinations), Продукція для перевезення (Products) та

переглядати значення в таблицях: Контракти (Contracts), Пункти призначення (Destinations), Продукція для перевезення (Products), Компанії замовники / Клієнти (Clients), Порти (Ports), Кораблі (Ships). Слідкує за запитамі компаній, які користуються послугами перевезення та у разі потреби додає нові контракти, пункти призначення та продукцію для перевезення до бази даних.

- Менеджер по географії (GeographyManager) - має права додавати значення в таблиці: Країни (Countries), Порти (Ports) та переглядати значення в таблицях: Країни (Countries), Порти (Ports). Слідкує за ситуацією в світі та у разі появи нових країн/портів додає їх до бази даних.

за допомогою команд:

```
CREATE USER 'Admin'@'localhost' IDENTIFIED BY 'Aqwerty';
GRANT ALL PRIVILEGES ON shipping.* TO 'Admin'@'localhost';
```

```
CREATE USER 'TransportationManager'@'localhost' IDENTIFIED BY 'Tqwerty';
GRANT UPDATE ON shipping.ships TO 'TransportationManager'@'localhost';
GRANT INSERT ON shipping.ships TO 'TransportationManager'@'localhost';
GRANT INSERT ON shipping.workers TO 'TransportationManager'@'localhost';
GRANT SELECT ON shipping.ships TO 'TransportationManager'@'localhost';
GRANT SELECT ON shipping.workers TO 'TransportationManager'@'localhost';
GRANT SELECT ON shipping.freightcompanies TO
'TransportationManager'@'localhost';
GRANT SELECT ON shipping.countries TO 'TransportationManager'@'localhost';
GRANT SELECT ON shipping.shiptypes TO 'TransportationManager'@'localhost';
GRANT SELECT ON shipping.positions TO 'TransportationManager'@'localhost';
```

```
CREATE USER 'OrderManager'@'localhost' IDENTIFIED BY 'Oqwerty';
GRANT INSERT ON shipping.contracts TO 'OrderManager'@'localhost';
GRANT INSERT ON shipping.destinations TO 'OrderManager'@'localhost';
GRANT INSERT ON shipping.products TO 'OrderManager'@'localhost';
GRANT SELECT ON shipping.contracts TO 'OrderManager'@'localhost';
GRANT SELECT ON shipping.destinations TO 'OrderManager'@'localhost';
GRANT SELECT ON shipping.products TO 'OrderManager'@'localhost';
```

```

GRANT SELECT ON shipping.clients TO 'OrderManager'@'localhost';
GRANT SELECT ON shipping.ports TO 'OrderManager'@'localhost';
GRANT SELECT ON shipping.ships TO 'OrderManager'@'localhost';

CREATE USER 'GeographyManager'@'localhost' IDENTIFIED BY 'Gqwerty';
GRANT INSERT ON shipping.countries TO 'GeographyManager'@'localhost';
GRANT SELECT ON shipping.countries TO 'GeographyManager'@'localhost';
GRANT INSERT ON shipping.ports TO 'GeographyManager'@'localhost';
GRANT SELECT ON shipping.ports TO 'GeographyManager'@'localhost';

```

### 1.5. Заповнення бази даних даними

Кількість даних дуже велика (більше 7к), тому я не став додавати їх до додатку. Знайти їх можна за посиланням:

<https://github.com/kotenko2002/Shipping-database/blob/master/AddInfo.sql>

### 1.6. Створення представлень

Створюємо представлення аби полегшити подальшу роботу. Найчастіше використовується комбінації таблиць (Компанії перевізники (FreightCompanies), Кораблі (Ships)) та (Порти (Ports), Пункти призначення(Destinations), Контракти (Contracts)). Тож зробимо з них представлення за допомогою команд:

```

-- вьюшка, зв'язуюча таблиці "Компанії перевізники" з "Кораблі"
DROP VIEW IF EXISTS freightcompanies_ships;
CREATE VIEW freightcompanies_ships
AS
SELECT freightcompanies.Id FC_Id, CountryCode, freightcompanies.Name FC_Name,
PhoneNumber, Email,
ships.Id S_Id, ShipTypeId, FreightCompanyId, ships.Name S_Name
FROM ships
JOIN freightcompanies ON freightcompanies.Id = ships.FreightCompanyId;

-- вьюшка, зв'язуюча таблиці "порти" з "контрактами" через табл "точки
призначення"

```

```

CREATE VIEW ports_contracts
AS
SELECT ports.Id Ports_Id, CountryCode, ports.Name Ports_Name,
contracts.Id Contracts_Id, ShipId, ClientId, SigningDate
FROM destinations
JOIN ports ON ports.Id = destinations.PortId
JOIN contracts ON contracts.Id = destinations.ContractId;

```

## 1.7. Створення процедур та функцій

Створюємо процедури для отримання потрібних даних з БД та функції для отримання для використання в тригерах.

-- функція для знаходження поточної ваги товару на кораблі. Використовується в тригері

```

DELIMITER $$
CREATE FUNCTION get_current_ShipWeight(C_Id INT)
RETURNS INT
BEGIN
DECLARE currentWeight INT;
SELECT SUM(Weight) INTO currentWeight
FROM Products
JOIN Contracts ON Contracts.Id = Products.ContractId
WHERE ShipId = (
SELECT ShipId
FROM Contracts
WHERE Contracts.Id = C_Id
LIMIT 1
) ;
RETURN currentWeight;
END$$
DELIMITER ;

```

-- процедура для знаходження всіх портів за назвою країни

```

DELIMITER $$
CREATE PROCEDURE get_all_Ports_by_CountryName(IN NameOfCountry VARCHAR(20))
BEGIN
SELECT ports.Id, ports.Name FROM countries
JOIN ports ON ports.CountryCode = countries.Code

```



```

WHERE countries.Name = NameOfCountry;
END$$
DELIMITER ;
CALL get_all_Ports_by_CountryName('CША');
...

```

Всі інші процедури та функції можна переглянути в “Додатку Б” або за посиланням:

<https://github.com/kotenko2002/Shipping-database/blob/master/Functions%20and%20Procedures.sql>

### 1.8. Створення тригерів

Для безпомилкової роботи бази даних *shipping* створюємо наступні тригери:

- При INSERT в таблицю Продукція для перевезення(Products) перевіряється, чи є достатньо вільного місця на кораблі, який використовується в контракті, за яким закріплений продукт.
- При INSERT в таблицю Працівники (Workers) перевіряється, чи є зарплата працівника  $\geq$  мінімалці, яка закріплена за кожною посадою.
- При UPDATE таблиці Працівники (Workers) перевіряється, чи є нова зарплата працівника  $\geq$  мінімалці, яка закріплена за кожною посадою.

```

DELIMITER |
CREATE TRIGGER validate_product_Insert
BEFORE INSERT ON Products
FOR EACH ROW BEGIN
IF (get_current_ShipWeight(NEW.ContractId) + NEW.Weight >
get_MaxWeight_by_ContractId(NEW.ContractId))
THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = "Overweight on the ship";

```

```

END IF;
END;

DELIMITER ||
CREATE TRIGGER validate_worker_Insert
BEFORE INSERT ON Workers
FOR EACH ROW BEGIN
IF (NEW.Salary < get_MinSalary_by_PositionId(NEW.PositionId))
THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = "The value 'Salary' is less than the minimum wage";
END IF;
END;

DELIMITER |
CREATE TRIGGER validate_product_Update
BEFORE UPDATE
ON Workers
FOR EACH ROW BEGIN
IF (NEW.Salary < get_MinSalary_by_PositionId(OLD.PositionId))
THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = "The new value of 'Salary' is less than the minimum wage";
END IF;
END;

```

## 1.9. Написання DLM запитів

Створюємо 20 DML запитів в рамках бази даних *shipping* за допомогою коду:

Код всіх 20 DML запитів можна переглянути в “Додатку В” або за посиланням:

<https://github.com/kotenko2002/Shipping-database/blob/master/Requests.sql>

```

-- (1) топ 10 країн з найбільшою кількістю портів
SELECT DISTINCT Countries.Name, Ports.CountryCode, COUNT(Ports.CountryCode)
FROM Ports
JOIN Countries ON Countries.Code = Ports.CountryCode

```

```
GROUP BY CountryCode
ORDER BY COUNT(Ports.CountryCode) DESC
LIMIT 10;
```

	Name	CountryCode	COUNT(Ports.CountryCode)
►	США	USA	14
	КНР	CHN	10
	Велика Британія	GBR	8
	Південна Корея	PRK	7
	Японія	JPN	7
	Індія	IND	6
	Бразилія	BRA	6
	Австралія	AUS	5
	Франція	FRA	3
	Іспанія	ESP	3

```
-- (2) топ 10 портів світу за відвідуваністю
```

	CountryCode	Ports_Name	COUNT(Ports_Id)
►	BRA	Тубарао	27
	PRK	Барнан	27
	SAU	Джидда	26
	PRK	Ульсан	25
	AUS	Ньюкасл	24
	VNM	Сайгон	24
	GBR	Дартмут	23
	IND	Джавахарлал Неру	23
	USA	Лаем Чабанг	22
	USA	Вірджинія	22

```
-- (3). топ 10 країн світу за відвідуваністю
```

	Name	COUNT(Ports_Id)
►	США	214
	КНР	152
	Велика Британія	133
	Японія	122
	Південна Корея	121
	Бразилія	104
	Індія	93
	Австралія	78
	Туреччина	62
	Саудівська Аравія	60

```
-- (4) кількість кораблів у компаній перевізників
SELECT CountryCode, FC_Name, COUNT(FC_Id) numberOfShips
FROM freightcompanies_ships
GROUP BY FC_Id;
```

	CountryCode	FC_Name	numberOfShips
►	USA	A.P. Moller-Maersk Group	15
	ITA	Mediterranean Shipping Company	39
	CHN	COSCO	36
	FRA	CMA CGM Group	26
	UKR	NIKO	13
	USA	Hapag-Lloyd	43
	JPN	Ocean Network Express	23
	TWN	Evergreen Marine Corp	17
	TWN	Yang Ming Marine Corporation	36
	SGP	Pacific International Line	33
	PRK	Hyundai MM	9

-- (5) кількість співробітників компаній перевізників на кораблях

	CountryCode	FC_Name	numberOfWorkers
►	USA	A.P. Moller-Maersk Group	200
	ITA	Mediterranean Shipping Company	439
	CHN	COSCO	446
	FRA	CMA CGM Group	295
	UKR	NIKO	154
	USA	Hapag-Lloyd	571
	JPN	Ocean Network Express	299
	TWN	Evergreen Marine Corp	208
	TWN	Yang Ming Marine Corporation	466
	SGP	Pacific International Line	384
	PRK	Hyundai MM	145

-- (6) контрактна інформація компанії перевізника та компаній, які користуються її послугами

```

SELECT FC_Name Freight_Company_Name, freightcompanies_ships.PhoneNumber
FC_PhoneNumber, freightcompanies_ships.Email FC_Email,
clients.Name Client_Company_Name, clients.PhoneNumber CC_PhoneNumber,
clients.Email CC_Email
FROM contracts
JOIN freightcompanies_ships ON S_Id = contracts.ShipId
JOIN clients ON clients.Id = contracts.ClientId
GROUP BY FC_Id, clients.Id
ORDER BY FC_Id;
```

Freight_Company_Name	FC_PhoneNumber	FC_Email	Client_Company_Name	CC_PhoneNumber	CC_Email
A.P. Moller-Maersk Group	+1-858-842-7616	MollerMaerskGroup@gmail.com	Honda	822(074)792-84-67	queuddemidoidoi-4159@yopmail.com
A.P. Moller-Maersk Group	+1-858-842-7616	MollerMaerskGroup@gmail.com	Toyota	960(9813)475-55-39	brennoussigrumu-5252@yopmail.com
A.P. Moller-Maersk Group	+1-858-842-7616	MollerMaerskGroup@gmail.com	Nestle	0(110)767-17-14	bipruzuyassa-7137@yopmail.com
A.P. Moller-Maersk Group	+1-858-842-7616	MollerMaerskGroup@gmail.com	Coca-Cola	46(934)484-72-01	kovatrepraprou-6788@yopmail.com
A.P. Moller-Maersk Group	+1-858-842-7616	MollerMaerskGroup@gmail.com	Danone	9(575)405-70-44	hoibejugroceu-8942@yopmail.com
A.P. Moller-Maersk Group	+1-858-842-7616	MollerMaerskGroup@gmail.com	ДЕРЕВОРУБ	738(348)607-41-91	xetrojoulleipro-1190@yopmail.com
A.P. Moller-Maersk Group	+1-858-842-7616	MollerMaerskGroup@gmail.com	Wood Work	539(202)230-34-50	trappeikifreipeu-5484@yopmail.com
Mediterranean Shipping ...	+39 038 728 472	MediterraneanShippingCompa...	Amazon	978(4931)859-86-24	neihasseimneunni-1645@yopmail.com
Mediterranean Shipping ...	+39 038 728 472	MediterraneanShippingCompa...	eBay	817(899)129-21-96	cosufroupropoi-3031@yopmail.com
Mediterranean Shipping ...	+39 038 728 472	MediterraneanShippingCompa...	Auchan	2(688)362-46-46	feipaddassaneu-8943@yopmail.com
Mediterranean Shipping ...	+39 038 728 472	MediterraneanShippingCompa...	Alibaba	293(0673)802-78-07	sexauquadere-3549@yopmail.com
Mediterranean Shipping ...	+39 038 728 472	MediterraneanShippingCompa...	Кернел	1(5212)462-54-22	wexoisobatre-4704@yopmail.com
Mediterranean Shipping ...	+39 038 728 472	MediterraneanShippingCompa...	НИБУЛОН	098-4624-324	xeunoipaquoipri-3176@yopmail.com
Mediterranean Shipping ...	+39 038 728 472	MediterraneanShippingCompa...	IGT	4(4194)990-98-64	gradauffepreumme-2771@yopmail.com
Mediterranean Shipping ...	+39 038 728 472	MediterraneanShippingCompa...	УкрЗлеваторПром	388(70)744-29-96	seunnaffeuussukeu-6589@yopmail.com
Mediterranean Shipping ...	+39 038 728 472	MediterraneanShippingCompa...	Nucor Corporation	00(40)644-14-27	bitouhaunifre-4285@yopmail.com
Mediterranean Shipping ...	+39 038 728 472	MediterraneanShippingCompa...	Метинвест	096-3488-091	jerevesewoi-9947@yopmail.com

-- (7) інформація про завантаженість кожного корабля

	Name	CurrentWeight	FreeWeight	MaxWeight
►	The Hateful Terror	23000	2000	25000
	The Disgrace of the Coward	NULL	NULL	320000
	The Shadows of the Executioners	79000	1000	80000
	The Corrupted Galley	120000	200000	320000
	The Liberty Grail	44000	1000	45000
	The Madness of the Disgrace	218000	332000	550000
	The Buccaneers Shark	158000	2000	160000
	The Poseidon's Tortuga	159000	1000	160000
	The Happy Hades	24000	1000	25000
	The Howl of the Doom	41000	119000	160000
	The Bearded Wolf	NULL	NULL	45000
	The Poison Delight	80000	0	80000
	The Revenge Deceit	81000	79000	160000
	The Corsair	20000	5000	25000
	The Revenge Doubloon	37000	8000	45000
	The Victory Secret	43000	2000	45000
	The Black Curse	19000	141000	160000

-- (8) повна інформація про продукт

	ProductName	ClientName	Weight	Price
►	IS-481442-SKTFS566	Amazon	47000	94000
	AB-537863-FXWFX169	Amazon	24000	72000
	PQ-441331-UKREC164	Amazon	14000	70000
	CM-610642-IQUWB497	Amazon	30000	90000
	VS-970631-LSLJB149	Amazon	35000	70000
	MQ-867421-DIPEJ499	Amazon	50000	200000
	RQ-624968-QEXVJ626	Amazon	17000	102000
	NC-021055-CQLWS990	Amazon	8000	8000
	CQ-536223-HDXGI654	Amazon	27000	81000
	XA-492744-XIFUW003	Amazon	10000	40000
	MB-241507-GFUNM272	Amazon	24000	48000
	HK-445055-TJOQL896	Amazon	5000	15000
	UH-181220-CWIJF319	Amazon	48000	48000
	EK-643163-UMIBH676	Amazon	42000	168000
	VX-199093-BBGNN668	Amazon	27000	162000
	TB-489907-ABVMS097	Amazon	6000	36000
	ET-028405-MBJPA939	Amazon	6000	24000

-- (9) середня ціна однієї тонни вантажу кожного клієнта

```
SELECT clients.Name, ROUND(SUM(Price)/SUM(Weight),2)
FROM products
JOIN contracts ON contracts.Id = products.ContractId
JOIN clients ON clients.Id = contracts.ClientId
GROUP BY clients.Id
ORDER BY ROUND(SUM(Price)/SUM(Weight),2) DESC;
```

	Name	ROUND(SUM(Price)/SUM(Weight),2)
►	PepsiCo	4.05
	Auchan	4.00
	Toyota	3.89
	Nucor Corporation	3.87
	Danone	3.87
	eBay	3.84
	Кернел	3.83
	Метинвест	3.81
	Coca-Cola	3.80
	НИБУЛОН	3.71
	IGT	3.63
	Volkswagen	3.61
	Nestle	3.47
	Wood Work	3.45
	ArcelorMittal	3.34
	ДЕРЕВОРУБ	3.33

-- (10) інформація про команду корабля

```
SELECT FullName, positions.Name, PhoneNumber, Salary
FROM workers
JOIN positions ON positions.Id = workers.PositionId
JOIN ships ON ships.Id = workers.ShipId
```

WHERE ships.Name = 'The Hateful Terror';

	FullName	Name	PhoneNumber	Salary
►	Kirk Gutierrez	Капітан	060 439 0732	12887
	Cynthia Stevens	Старший помічник	079 231 2633	8317
	Jerry Sanchez	Другий помічник	031 375 2049	6249
	Tim Shaw	Старший механік	079 067 1489	14127
	Delores Brown	Механік	024 174 2395	7728
	Jose Riley	Боцман	096 804 0828	4436
	Delores Garcia	Матрос	053 026 6723	1209
	John Tucker	Кок	076 274 8847	1485

-- (11) інформація, які порти/країни корабель повинен відвідати за його Id

	Ports_Name	Country_Name
►	Нінбо	КНР
	Гонконг	КНР
	Бендер-Аббас	Іран
	Токіо	Японія
	Берген	Норвегія
	Ботас	Туреччина

-- (12) інформація, про всіх "назва професії" в компанії "назва компанії"

	WorkerName	ShipName	PhoneNumber	Salary
►	Kirk Gutierrez	The Hateful Terror	060 439 0732	12887
	Constance Ramirez	The Disgrace of the Coward	061 561 9802	8140
	Aaron Cruz	The Shadows of the Executioners	078 907 5300	16440
	Melissa Ramirez	The Corrupted Galley	048 300 8157	12305
	Daniel Gonzales	The Liberty Grail	059 530 4678	20967
	Michelle Andrews	The Madness of the Disgrace	044 029 8193	14939
	Sean Brown	The Buccaneers Shark	065 898 9400	15343
	Kimberly Lewis	The Poseidon's Tortuga	040 026 6987	9166
	Kevin Rodriguez	The Happy Hades	028 545 9790	8334
	Roger Carter	The Howl of the Doom	071 460 3439	16052
	Ethel Baker	The Bearded Wolf	083 051 6303	19362
	Joyce Clark	The Poison Delight	032 131 3666	11809
	Theodore Maxwell	The Revenge Deceit	083 087 9773	10558
	Christopher Williams	The Corsair	075 384 9770	15890
	Bertha Stephens	The Revenge Doubloon	051 049 9812	8432

-- (13) компанії, продукцію яких буде перевозити корабель по Id корабля

	Name	SigningDate
►	ДЕРЕВОРУБ	2021-10-16
	ArcelorMittal	2021-08-17
	Wood Work	2021-12-14
	IGT	2021-10-13

-- (14) інформація, в яких країнах побуває працівник корабля по Id працівника

	Name
►	Південна Корея
	Іран
	Бразилія
	Франція
	Норвегія
	Австралія
	ПАР
	Малайзія

```
-- (15) вся продукція, що перевозиться кораблем по Id корабля
SELECT products.Name, Weight, Price
FROM ships
JOIN contracts ON contracts.ShipId = ships.Id
JOIN products ON contracts.Id = products.ContractId
WHERE ships.Id = 3;
```

	Name	Weight	Price
►	RD-008359-GCJFM050	33000	99000
	JT-369911-XJXPB256	41000	164000
	GP-074960-EPHEN240	5000	10000

```
-- (16) інформація про кількість працівників, загальну суму ЗА, та середню
```

	Toatal_workers	Total_Salary	AVG_Salary
►	439	2875068	6549.13

```
-- (17) прибуток компаній перевізників за останні "n" місяців
```

	Name	SUM(products.Price)
►	A.P. Moller-Maersk Group	3546000
	Mediterranean Shipping Company	7981000
	COSCO	6890000
	CMA CGM Group	5139000
	NIKO	2773000
	Hapag-Lloyd	9773000
	Ocean Network Express	5170000
	Evergreen Marine Corp	2466000
	Yang Ming Marine Corporation	8148000
	Pacific International Line	4428000
	Hyundai MM	2717000

```
-- (18) прибутки перевізника за різні місяці по Id компанії
```



	Name	SUM(products.Price)	MONTHNAME(SigningDate)
►	Hapag-Lloyd	792000	August
	Hapag-Lloyd	1981000	September
	Hapag-Lloyd	1140000	October
	Hapag-Lloyd	2771000	November
	Hapag-Lloyd	3089000	December

-- (19) кораблі, що не мають жодного контракту

	Ship_Id	Ship_Name	Type_Name	MaxWeight	Owner_Name
►	2	The Disgrace of the Coward	Вантажне судно типу VL	320000	A.P. Moller-Maersk Group
	11	The Bearded Wolf	Вантажне судно середніх розмірів	45000	A.P. Moller-Maersk Group
	18	The Delight	Вантажне судно типу LR 1	80000	Mediterranean Shipping Company
	24	The Evil Wolf	Вантажне судно середніх розмірів	45000	Mediterranean Shipping Company
	27	The Evil Tainted	Вантажне судно середніх розмірів	45000	Mediterranean Shipping Company
	33	The Hellish Pillager	Вантажне судно типу LR 1	80000	Mediterranean Shipping Company
	41	The Festering Wolf	Вантажне судно загального призначення	25000	Mediterranean Shipping Company
	49	The Howling Murderers	Вантажне судно типу LR 1	80000	Mediterranean Shipping Company
	51	The Good Fear	Вантажне судно загального призначення	25000	Mediterranean Shipping Company
	53	The Fearful Demon	Вантажне судно середніх розмірів	45000	Mediterranean Shipping Company
	54	The Privateers Killers	Вантажне судно типу LR 1	80000	Mediterranean Shipping Company
	58	The Dishonorable Wolf	Вантажне судно загального призначення	25000	COSCO
	59	The Mermaid's Dragon	Вантажне судно типу LR2	160000	COSCO
	60	The Davy Jones Pearl	Вантажне судно середніх розмірів	45000	COSCO
	62	The Cacophonous Trinity	Вантажне судно загального призначення	25000	COSCO
	68	The Murderous Fear	Вантажне судно типу VL	320000	COSCO

-- (20) кораблі, що не мають жодного продукту на борту

	Ship_Name	Type	MaxWeight
►	The Disgrace of the Coward	Вантажне судно типу VL	320000
	The Bearded Wolf	Вантажне судно середніх розмірів	45000
	The Delight	Вантажне судно типу LR 1	80000
	The Evil Wolf	Вантажне судно середніх розмірів	45000
	The Evil Tainted	Вантажне судно середніх розмірів	45000
	The Hellish Pillager	Вантажне судно типу LR 1	80000
	The Festering Wolf	Вантажне судно загального призначення	25000
	The Howling Murderers	Вантажне судно типу LR 1	80000
	The Good Fear	Вантажне судно загального призначення	25000
	The Fearful Demon	Вантажне судно середніх розмірів	45000
	The Privateers Killers	Вантажне судно типу LR 1	80000
	The Dishonorable Wolf	Вантажне судно загального призначення	25000
	The Mermaid's Dragon	Вантажне судно типу LR2	160000
	The Davy Jones Pearl	Вантажне судно середніх розмірів	45000
	The Cacophonous Trinity	Вантажне судно загального призначення	25000
	The Murderous Fear	Вантажне судно типу VL	320000

## 1.10. Оптимізація та її результати

Для оптимізації запитів створимо індекси за допомогою коду:

```
CREATE INDEX shipsName_idx ON ships(Name);
CREATE INDEX freightcompaniesName_idx ON freightcompanies(Name);
CREATE INDEX positionsName_idx ON positions(Name);
```

```
CREATE INDEX contractsSigningDate_idx ON contracts(SigningDate);
CREATE INDEX productsWeight_idx ON products(Weight);
```

### Запити та їхній час виконання до та після оптимізація

			До	Після
✓	566	16:22:53	SELECT DISTINCT Countries.Name, Ports.CountryCode, COUNT(Ports.CountryCode) FROM Ports JOIN Countries ON Countries.Code = Ports...	0.000 sec 0.000 sec
✓	567	16:22:53	SELECT CountryCode, Ports_Name, COUNT(Ports_Id) FROM ports_contracts GROUP BY Ports_Id ORDER BY COUNT(Ports_Id) DESC LIMIT...	0.016 sec 0.016 sec
✓	568	16:22:53	SELECT countries.Name, COUNT(Ports_Id) FROM ports_contracts JOIN countries ON countries.Code = CountryCode GROUP BY countries.Co...	0.015 sec 0.000 sec
✓	569	16:22:53	SELECT CountryCode, FC_Name, COUNT(FC_Id) numberOfShips FROM freightcompanies_ships GROUP BY FC_Id LIMIT 0, 1000	0.000 sec 0.000 sec
✓	570	16:22:53	SELECT CountryCode, FC_Name, COUNT(workers.Id) numberOfWorkers FROM workers JOIN freightcompanies_ships ON S_Id = workers.Shipl...	0.016 sec 0.016 sec
✓	571	16:22:53	SELECT FC_Name Freight_Company_Name, freightcompanies_ships.PhoneNumber FC_PhoneNumber, freightcompanies_ships.Email FC_Email, ...	0.016 sec 0.016 sec
✓	572	16:22:53	SELECT ships.Name, SUM(products.Weight) CurrentWeight, shiptypes.MaxWeight - SUM(products.Weight) FreeWeight, shiptypes.MaxWeight ...	0.031 sec 0.015 sec
✓	573	16:22:54	SELECT products.Name ProductName, clients.Name ClientName, Weight, Price FROM products JOIN contracts ON contracts.Id = products.Con...	0.016 sec 0.000 sec
✓	574	16:22:54	SELECT clients.Name, ROUND(SUM(Price)/SUM(Weight),2) FROM products JOIN contracts ON contracts.Id = products.ContractId JOIN client...	0.000 sec 0.016 sec
✓	575	16:22:54	SELECT FullName, positions.Name, PhoneNumber, Salary FROM workers JOIN positions ON positions.Id = workers.PositionId JOIN ships ON sh...	0.047 sec 0.000 sec
✓	576	16:22:54	SELECT Ports_Name, countries.Name Country_Name FROM ships JOIN ports_contracts ON ships.Id = ShipId JOIN countries ON countries.Cod...	0.032 sec 0.000 sec
✓	577	16:22:54	SELECT FullName WorkerName, ships.Name ShipName, workers.PhoneNumber, Salary FROM freightcompanies JOIN ships ON freightcompanie...	0.000 sec 0.000 sec
✓	578	16:22:54	SELECT clients.Name, SigningDate FROM ships JOIN contracts ON contracts.ShipId = ships.Id JOIN clients ON clients.Id = contracts.ClientId W...	0.000 sec 0.000 sec
✓	579	16:22:54	SELECT DISTINCT countries.Name FROM workers JOIN ships ON ships.Id = workers.ShipId JOIN ports_contracts ON ships.Id = ports_contract...	0.000 sec 0.000 sec
✓	580	16:22:54	SELECT products.Name, Weight, Price FROM ships JOIN contracts ON contracts.ShipId = ships.Id JOIN products ON contracts.Id = products.C...	0.000 sec 0.000 sec
✓	581	16:22:54	SELECT COUNT(workers.Id) Total_workers, SUM(workers.Salary) Total_Salary, ROUND(SUM(workers.Salary)/COUNT(workers.Id),2) AVG_S...	0.015 sec 0.000 sec
✓	582	16:22:54	SELECT freightcompanies.Name, SUM(products.Price) FROM freightcompanies JOIN ships ON ships.FreightCompanyId = freightcompanies.Id JO...	0.015 sec 0.000 sec
✓	583	16:22:54	SELECT freightcompanies.Name, SUM(products.Price), MONTHNAME(SigningDate) FROM freightcompanies JOIN ships ON ships.FreightComp...	0.000 sec 0.000 sec
✓	584	16:22:54	SELECT S_Id Ship_Id, S_Name Ship_Name, shiptypes.Name Type_Name, shiptypes.MaxWeight, FC_Name Owner_Name FROM freightcompa...	0.000 sec 0.000 sec
✓	585	16:22:54	SELECT ships.Name Ship_Name, shiptypes.Name Type, shiptypes.MaxWeight FROM products JOIN contracts ON contracts.Id = products.Cont...	0.016 sec 0.016 sec

## ВИСНОВКИ

Метою курсової роботи було закріплення студентом навичок з проектування, реалізації реляційних баз даних та їх використання для практичних задач.

Тож для виконання курсової роботи були проведені необхідні дослідження, що безпосередньо стосуються предметної області, а саме діяльності компанії, що займається морськими перевезеннями. Спочатку я провів повний аналіз аналіз нашої предметної області, а вже тоді почав її описувати.

Після цього я побудував концептуальну модель для якої я використовував мову ER-опису предметної області, яка базується на концепції, що інформаційна модель предметної області може бути описана із застосування таких понять, як сутність, атрибут, зв'язок. Після побудови ER діаграми, я провів нормалізацію таблиць нашої бази даних.

І тільки після цих дій я перейшов до практичної частини і почав створювати саму базу даних, її таблиць. Потім я створив діаграму, аби перевірити цілісність бази даних і коректність.

Далі я створив користувачів, заповнив базу даних даними, створив представлення, процедури, функції, тригери та 20 DML запитів. І в кінці оптимізував їх.

В даній курсовій роботі була розроблена база даних «База даних підтримки діяльності компанії, що займається морськими вантажними перевезеннями » в системі управління базами даних MySQL WorkBench.

## СПИСОК ЛІТЕРАТУРИ

- Лекційний матеріал з предмету «Бази даних»
- METANIT <https://metanit.com/sql/mysql/1.1.php>
- Офіційна документація MySQL <https://dev.mysql.com/doc/>
- Advanced SQL <https://www.kaggle.com/learn/advanced-sql>

## ДОДАТКИ

### Додаток А. Створення таблиць бази даних

```
-- створення таблиці Країни (Countries)
CREATE TABLE Countries (
    Code CHAR(3) PRIMARY KEY,
    Name VARCHAR(20) NOT NULL
);

-- створення таблиці Компанії перевізники (FreightCompanies)
CREATE TABLE FreightCompanies (
    Id INT PRIMARY KEY AUTO_INCREMENT,
    CountryCode CHAR(3),
    Name VARCHAR(30) NOT NULL,
    PhoneNumber VARCHAR(17) NOT NULL,
    Email VARCHAR(40) NOT NULL,
    FOREIGN KEY (CountryCode)
        REFERENCES Countries (Code)
);

-- створення таблиці Типи кораблів (ShipTypes)
CREATE TABLE ShipTypes (
    Id INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(40) NOT NULL,
    MaxWeight INT NOT NULL
);

-- створення таблиці Кораблі (Ships)
CREATE TABLE Ships (
    Id INT PRIMARY KEY AUTO_INCREMENT,
    ShipTypeId INT,
    FreightCompanyId INT,
    Name VARCHAR(40) NOT NULL,
    FOREIGN KEY (ShipTypeId)
        REFERENCES ShipTypes (Id),
    FOREIGN KEY (FreightCompanyId)
        REFERENCES FreightCompanies (Id)
```

```
);
```

```
-- створення таблиці Посади (Positions)
```

```
CREATE TABLE Positions (  
    Id INT PRIMARY KEY AUTO_INCREMENT,  
    Name VARCHAR(20) NOT NULL,  
    MinSalary INT NOT NULL  
);
```

```
-- створення таблиці Працівники (Workers)
```

```
CREATE TABLE Workers (  
    Id INT PRIMARY KEY AUTO_INCREMENT,  
    PositionId INT,  
    ShipId INT,  
    FullName VARCHAR(30) NOT NULL,  
    PhoneNumber VARCHAR(17) NOT NULL,  
    Salary INT NOT NULL,  
    FOREIGN KEY (PositionId)  
        REFERENCES Positions (Id),  
    FOREIGN KEY (ShipId)  
        REFERENCES Ships (Id)  
);
```

```
-- створення таблиці Порти (Ports)
```

```
CREATE TABLE Ports (  
    Id INT PRIMARY KEY AUTO_INCREMENT,  
    CountryCode CHAR(3),  
    Name VARCHAR(20) NOT NULL,  
    FOREIGN KEY (CountryCode)  
        REFERENCES Countries (Code)  
);
```

```
-- створення таблиці Компанії замовники / Клієнти (Clients)
```

```
CREATE TABLE Clients (  
    Id INT PRIMARY KEY AUTO_INCREMENT,  
    Name VARCHAR(20) NOT NULL,  
    PhoneNumber VARCHAR(20) NOT NULL,  
    Email VARCHAR(35) NOT NULL  
);
```

```
-- створення таблиці Контракти (Contracts)
```

```

CREATE TABLE Contracts (
    Id INT PRIMARY KEY AUTO_INCREMENT,
    ShipId INT NULL,
    ClientId INT NOT NULL,
    SigningDate DATE NOT NULL,
    FOREIGN KEY (ShipId)
        REFERENCES Ships (Id),
    FOREIGN KEY (ClientId)
        REFERENCES Clients (Id)
);

-- створення таблиці Пункти призначення(Destinations)
CREATE TABLE Destinations (
    Id INT PRIMARY KEY AUTO_INCREMENT,
    PortId INT,
    ContractId INT,
    FOREIGN KEY (PortId)
        REFERENCES Ports (Id),
    FOREIGN KEY (ContractId)
        REFERENCES Contracts (Id)
);

-- створення таблиці Продукція для перевезення(Products)
CREATE TABLE Products (
    Id INT PRIMARY KEY AUTO_INCREMENT,
    ContractId INT,
    Name VARCHAR(18) NOT NULL,
    Weight INT NOT NULL,
    Price INT NOT NULL,
    FOREIGN KEY (ContractId)
        REFERENCES Contracts (Id)
);

```

## Додаток Б. Створення процедур та функцій

```

-- функція для знаходження мінімальної зарплати по Id професії.
Використовується в тригері
DELIMITER $$
CREATE FUNCTION get_MinSalary_by_PositionId(PosId INT)
RETURNS INT

```

```

BEGIN
DECLARE minSal INT;
SELECT MinSalary INTO minSal
FROM positions
WHERE positions.Id = PosId;
    RETURN minSal;
END$$

DELIMITER ;
SELECT get_MinSalary_by_PositionId(5);

-- функція для знаходження поточної ваги товару на кораблі. Використовується
в тригері
DELIMITER $$
CREATE FUNCTION get_current_ShipWeight(C_Id INT)
RETURNS INT
BEGIN
DECLARE currentWeight INT;
SELECT SUM(Weight) INTO currentWeight
FROM Products
JOIN Contracts ON Contracts.Id = Products.ContractId
WHERE ShipId = (
SELECT ShipId
FROM Contracts
WHERE Contracts.Id = C_Id
LIMIT 1
) ;
    RETURN currentWeight;
END$$

DELIMITER ;
SELECT get_current_ShipWeight(1);

-- функція для знаходження максимальної вантажопідйомності корабля.
Використовується в тригері
DELIMITER $$
CREATE FUNCTION get_MaxWeight_by_ContractId(C_Id INT)
RETURNS INT
BEGIN
DECLARE maxW INT;
SELECT Shiptypes.MaxWeight INTO maxW
FROM Contracts
JOIN Ships ON Ships.Id = Contracts.ShipId

```

```

JOIN Shiptypes ON Shiptypes.Id = Ships.ShipTypeId
WHERE
Ships.Id = (
SELECT ShipId
FROM Contracts
WHERE Contracts.Id = C_Id
LIMIT 1
)
LIMIT 1;
        RETURN maxW;
END$$
DELIMITER ;
SELECT get_MaxWeight_by_ContractId(500);

-- процедура для знаходження всіх кораблів вказаного типу компаній
перевізників
DELIMITER $$
CREATE PROCEDURE get_all_ships_for_all_FC_by_ShipTypeId(IN typeOfShips INT)
BEGIN
SELECT FC_Name, COUNT(FC_Id) numberOfShips
FROM freightcompanies_ships
WHERE ShipTypeId = typeOfShips
GROUP BY FC_Id;
END$$
DELIMITER ;
CALL get_all_ships_for_all_FC_by_ShipTypeId(2);

-- процедура для знаходження всіх портів за назвою країни
DELIMITER $$
CREATE PROCEDURE get_all_Ports_by_CountryName(IN NameOfCountry VARCHAR(20))
BEGIN
SELECT ports.Id, ports.Name FROM countries
JOIN ports ON ports.CountryCode = countries.Code
WHERE countries.Name = NameOfCountry;
END$$
DELIMITER ;
CALL get_all_Ports_by_CountryName('США');

```

## Додаток В. Написання DLM запитів



```
-- (1) топ 10 країн з найбільшою кількістю портів
SELECT DISTINCT Countries.Name, Ports.CountryCode, COUNT(Ports.CountryCode)
FROM Ports
JOIN Countries ON Countries.Code = Ports.CountryCode
GROUP BY CountryCode
ORDER BY COUNT(Ports.CountryCode) DESC
LIMIT 10;
```

```
-- (2) топ 10 портів світу за відвідуваністю
SELECT CountryCode, Ports_Name, COUNT(Ports_Id)
FROM ports_contracts
GROUP BY Ports_Id
ORDER BY COUNT(Ports_Id) DESC
LIMIT 10;
```

```
-- (3). топ 10 країн світу за відвідуваністю
SELECT countries.Name, COUNT(Ports_Id)
FROM ports_contracts
JOIN countries ON countries.Code = CountryCode
GROUP BY countries.Code
ORDER BY COUNT(Ports_Id) DESC
LIMIT 10;
```

```
-- (4) кількість кораблів компаній перевізників
SELECT CountryCode, FC_Name, COUNT(FC_Id) numberOfShips
FROM freightcompanies_ships
GROUP BY FC_Id;
```

```
-- (5) кількість співробітників компаній перевізників на кораблях
SELECT CountryCode, FC_Name, COUNT(workers.Id) numberOfWorkers
FROM workers
JOIN freightcompanies_ships ON S_Id = workers.ShipId
GROUP BY FC_Id;
```

```
-- (6) контрактна інформація компанії перевізника та компаній, які
користуються її послугами
SELECT FC_Name Freight_Company_Name, freightcompanies_ships.PhoneNumber
FC_PhoneNumber, freightcompanies_ships.Email FC_Email,
clients.Name Client_Company_Name, clients.PhoneNumber CC_PhoneNumber,
clients.Email CC_Email
FROM contracts
```

```
JOIN freightcompanies_ships ON S_Id = contracts.ShipId
JOIN clients ON clients.Id = contracts.ClientId
GROUP BY FC_Id, clients.Id
ORDER BY FC_Id;
```

```
-- (7) інформація про завантаженість кожного корабля
SELECT ships.Name, SUM(products.Weight) CurrentWeight,
shiptypes.MaxWeight - SUM(products.Weight) FreeWeight, shiptypes.MaxWeight
FROM products
JOIN contracts ON contracts.Id = products.ContractId
RIGHT JOIN ships ON ships.Id = contracts.ShipId
JOIN shiptypes ON shiptypes.Id = ships.ShipTypeId
GROUP BY ships.Id
order by ships.Id;
```

```
-- (8) повна інформація про продукт
SELECT products.Name ProductName, clients.Name ClientName, Weight, Price
FROM products
JOIN contracts ON contracts.Id = products.ContractId
JOIN clients ON clients.Id = contracts.ClientId;
```

```
-- (9) середня ціна однієї тонни вантажу кожного клієнта
SELECT clients.Name, ROUND(SUM(Price)/SUM(Weight),2)
FROM products
JOIN contracts ON contracts.Id = products.ContractId
JOIN clients ON clients.Id = contracts.ClientId
GROUP BY clients.Id
ORDER BY ROUND(SUM(Price)/SUM(Weight),2) DESC;
```

```
-- (10) інформація про команду корабля
SELECT FullName, positions.Name, PhoneNumber, Salary
FROM workers
JOIN positions ON positions.Id = workers.PositionId
JOIN ships ON ships.Id = workers.ShipId
WHERE ships.Name = 'The Hateful Terror';
```

```
-- (11) інформація, які порти/країни корабель повинен відвідати
SELECT Ports_Name, countries.Name Country_Name
FROM ships
JOIN ports_contracts ON ships.Id = ShipId
JOIN countries ON countries.Code = CountryCode
```

```
WHERE ships.Name = 'The Hateful Terror';
```

```
-- (12) інформація, про всіх "назва професії" в компанії "назва компанії"  
SELECT FullName WorkerName, ships.Name ShipName, workers.PhoneNumber, Salary  
FROM freightcompanies  
JOIN ships ON freightcompanies.Id = ships.FreightCompanyId  
JOIN workers ON ships.Id = workers.ShipId  
JOIN positions ON positions.Id = workers.PositionId  
WHERE freightcompanies.Name = 'A.P. Moller-Maersk Group' AND positions.Name =  
'Капітан';
```

```
-- (13) компанії, продукцію яких буде перевозити корабель по Id корабля  
SELECT clients.Name, SigningDate FROM ships  
JOIN contracts ON contracts.ShipId = ships.Id  
JOIN clients ON clients.Id = contracts.ClientId  
WHERE ships.Id = 3;
```

```
-- (14) інформація, в яких країнах побуває працівник корабля по Id працівника  
SELECT DISTINCT countries.Name  
FROM workers  
JOIN ships ON ships.Id = workers.ShipId  
JOIN ports_contracts ON ships.Id = ports_contracts.ShipId  
JOIN countries ON countries.Code = CountryCode  
WHERE workers.Id = 347;
```

```
-- (15) вся продукція, що перевозиться кораблем по Id корабля  
SELECT products.Name, Weight, Price  
FROM ships  
JOIN contracts ON contracts.ShipId = ships.Id  
JOIN products ON contracts.Id = products.ContractId  
WHERE ships.Id = 3;
```

```
-- (16) інформація про кількість працівників, загальну суму ЗА, та середню ЗП  
в компанії по Id компанії  
SELECT COUNT(workers.Id) Toatal_workers, SUM(workers.Salary) Total_Salary,  
ROUND(SUM(workers.Salary)/COUNT(workers.Id),2) AVG_Salary  
FROM workers  
JOIN freightcompanies_ships ON S_Id = workers.ShipId  
WHERE FC_Id = 2;
```

```
-- (17) прибуток компаній перевізників за останні "n" місяців
```

```

SELECT freightcompanies.Name, SUM(products.Price) FROM freightcompanies
JOIN ships ON ships.FreightCompanyId = freightcompanies.Id
JOIN contracts ON contracts.ShipId = ships.Id
JOIN products ON contracts.Id = products.ContractId
WHERE MONTH(NOW()) - MONTH(SigningDate) < 5
GROUP BY freightcompanies.Id;

```

```

-- (18) прибутки перевізника за різні місяці по Id компанії
SELECT freightcompanies.Name, SUM(products.Price), MONTHNAME(SigningDate)
FROM freightcompanies
JOIN ships ON ships.FreightCompanyId = freightcompanies.Id
JOIN contracts ON contracts.ShipId = ships.Id
JOIN products ON contracts.Id = products.ContractId
WHERE freightcompanies.Id = 6
group by MONTH(SigningDate)
ORDER BY MONTH(SigningDate);

```

```

-- (19) кораблі, що не мають жодного контракту
SELECT S_Id Ship_Id, S_Name Ship_Name,
shiptypes.Name Type_Name, shiptypes.MaxWeight, FC_Name Owner_Name
FROM freightcompanies_ships
JOIN shiptypes ON shiptypes.Id = ShipTypeId
WHERE NOT EXISTS
(SELECT * FROM contracts WHERE contracts.ShipId = S_Id)
ORDER BY S_Id;

```

```

-- (20) кораблі, що не мають жодного продукту на борту
SELECT ships.Name Ship_Name, shiptypes.Name Type, shiptypes.MaxWeight
FROM products
JOIN contracts ON contracts.Id = products.ContractId
RIGHT JOIN ships ON ships.Id = contracts.ShipId
JOIN shiptypes ON shiptypes.Id = ships.ShipTypeId
GROUP BY ships.Id
HAVING SUM(products.Weight) IS NULL
order by ships.Id;

```