

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА“**

Кафедра систем штучного інтелекту

Розрахунково-графічна робота

з дисципліни

«Організація баз даних та знань»

Предметна область:

«Туристична компанія»

Виконав:

Студент групи КН-214

Добрій Назарій

Прийняла:

Мельникова.Н.І.

Львів – 2021р.

ЗМІСТ

1. Вступ.....	3
2. Аналітичний огляд літератури.....	4
3. Визначення та опис предметної області.....	5
3.1 Аналіз вимог до бази даних	5
3.2 Створення ділової моделі організації.....	5
4. Побудова моделі «сутність-зв'язок»	7
5. Побудова логічної структури бази даних.....	7
6. Побудова обмежень бази даних. Нормалізація.....	8
7. Створення запитів різного характеру.....	15
8. Висновок.....	17
9. Список використаної літератури.....	18

ВСТУП

База даних – сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами; ця сукупність підтримує щонайменше одну з областей застосування . В загальному випадку база даних містить схеми, таблиці, збережені процедури та інші об'єкти. Дані у базі організовують відповідно до моделі організації даних. Таким чином, сучасна база даних, крім саме даних, містить їх опис та може містити засоби для їх обробки.

СУБД - система управління базами даних - сукупність програм, призначених для управління БД і можливості отримання користувачами необхідної інформації з бази. До завдань СУБД входять наступні завдання:

- 1) Формування і підтримка БД
- 2) Обробка інформації
- 3) Прийом запитів
- 4) Надання інформації користувачам
- 5) Забезпечення цілісності та реорганізації цінностей БД
- 6) Організація спільної роботи користувачів

1. Визначення та опис предметної області.

1.1. Аналіз вимог до бази даних:

В якості предметної області для бази даних я вибрав сферу туристична компанія. Проектована база даних може бути застосована для підприємства, яке займається туристичними послугами. Користувачем туристичних послуг може бути людина різного фінансового становища, оскільки є тури з різним ціновим діапазоном. Транзакція подорожі – це ціна туру в таблиці “Tour”, яка формується сумою ціни квитка в таблиці “Ticket” та ціни готелю в таблиці “Hotel”.

В базі даних буде зберігатися інформація про такі об’єкти:

- Клієнти
- Тур
- Країна
- Готель
- Аеропорт
- Літак
- Квиток
- Вантаж

Об’єкт “Клієнти” складається з ім’я, прізвища, номера мобільного телефону, пошти та унікального ідентифікатора id. Об’єкт “Тури” складається з локації, людини, яка замовила тур, готелю, аеропорта, ціни туру, кількість днів відпочинку. Об’єкту “Країна” складається з імені країни, континенту, де ця країна розташована, опису країни та ідентифікатора id країни. Об’єкт “Готель” складається з імені готелю, ціни за їжу, ціну за ніч, кількість зірок у готелі, телефонний номер готелю та ідентифікатора id готелю. Об’єкт “Аеропорт” складається з імені, локації аеропорту, ідентифікатора квитка та ідентифікатора id аеропорту. Об’єкт “Квиток” складається з імені локації, ціну квитка, ідентифікатора вантажу, дата відправлення та ідентифікатора id квитка. Об’єкт “Вантаж” складається з типу вантажу, кількість речей у вантажі, вага вантажу та ідентифікатора id вантажу.

Створені таблиці у базі даних:

- Customers – таблиця, яка зберігає дані про клієнтів
- Tours – таблиця, яка зберігає дані про тури
- Country – таблиця, яка зберігає дані про країни
- Hotel – таблиця, яка зберігає дані про готель
- Airport – таблиця, яка зберігає дані про аеропорт
- Plane – таблиця, яка зберігає дані про літак

- Ticket – таблиця, яка зберігає дані про квиток
- Cargo – таблиця, яка зберігає дані про вантаж

1.2. Створення ділового моделі організації:

Функції\Класи	Клієнти	Тур	Країна	Готель	Квиток	Літак	Аеро.	Багаж
Вибір туру		*	*	*	*			
Вибір місця відпочинку			*	*				
Оформлення документів на подорож	*	*						
Встановлення маршруту			*				*	
Пакування валіз	*							*
Встановлення кількості днів відпочинку		*			*			
Покупка квитків	*				*	*	*	

Рис. 1. Ділова модель організації «Туристична компанія»

2. Розроблення бази даних.

Першим етапом процесі проектування і створення бази даних, є розробка ER-моделі.

ER - діаграми використовуються для розробки даних і являють собою стандартний спосіб визначення даних і відносин між ними. Таким чином, здійснюється деталізація сховищ даних. Основними поняттями ER-діаграми є сутність, зв'язок, атрибут.

Побудуємо ER – діаграму бази даних відповідно обраної теми:

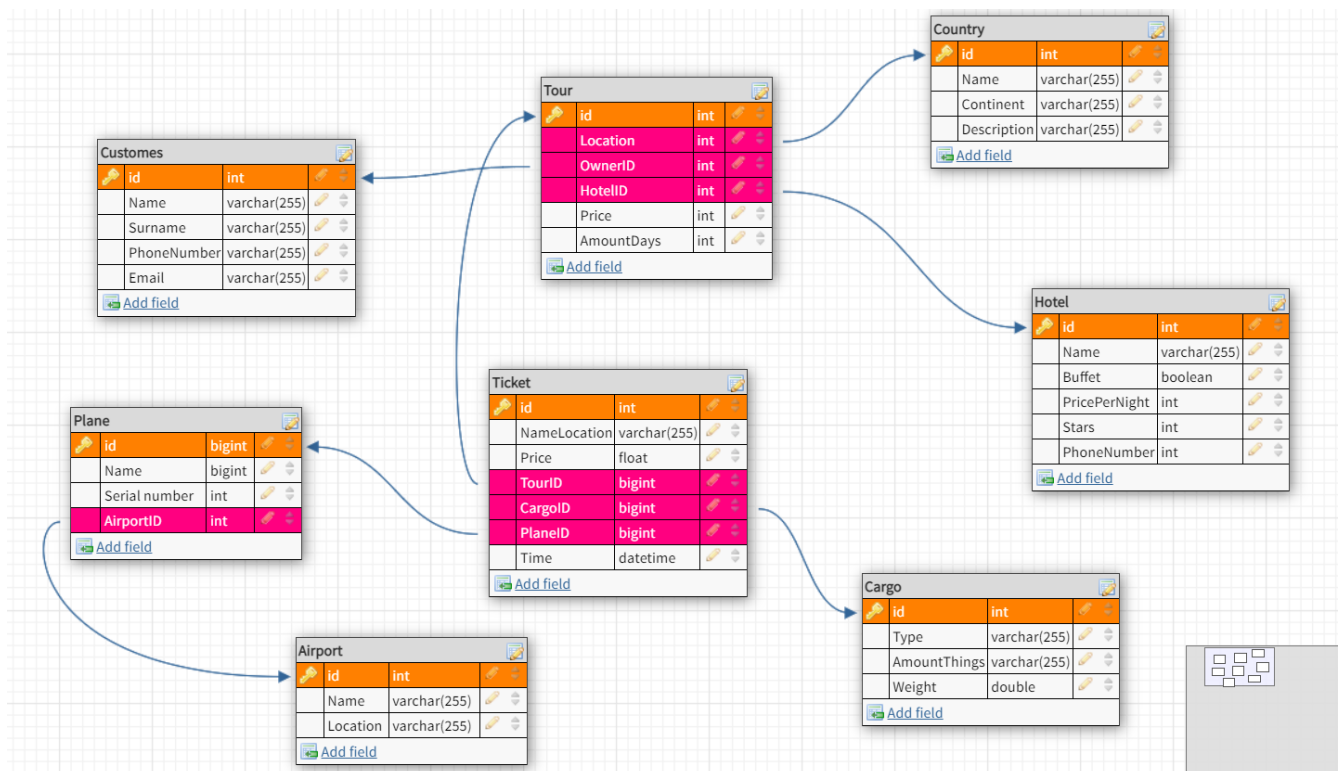


Рис. 2. ER – діаграма «Туристична компанія»

2.2. Логічна структура бази даних:

Назва таблиці	Ім'я поля	Тип даних
Customers	<u>id</u>	Лічильник
	FirstName	Текстовий
	SurName	Текстовий
	Phone	Текстовий
	Email	Текстовий
	<u>id</u>	Лічильник
	<u>Location</u>	Числовий

Tour	<u>OwnerID</u> <u>HotedID</u> Price AmountDays	(Майстер підстановки) Числовий (Майстер підстановки) Числовий (Майстер підстановки) Числовий Числовий
Country	<u>id</u> Name Continent Description	Лічильник Текстовий Текстовий Текстовий
Hotel	<u>id</u> Name Buffet PricePerNight Stars Phone	Лічильник Текстовий Числовий Числовий Числовий Текстовий
Tiket	<u>id</u> NameLocation Price <u>TourID</u> <u>CargoID</u> <u>PlaneID</u> Time	Лічильник Текстовий Числовий Числовий (Майстер підстановки) Числовий (Майстер підстановки) Числовий (Майстер підстановки) Числовий
Cargo	<u>id</u> Type AmountThings Weight	Лічильник Текстовий Числовий Числовий
Plane	<u>id</u> Name SerialNumber AirportID	Лічильник Текстовий Числовий Числовий (Майстер підстановки)
Airport	<u>id</u> Name Location	Лічильник Текстовий Текстовий

2.3. Побудова обмежень бази даних:

Таблиця Customers:

```
CREATE TABLE Customers(  
    id int auto_increment not null,  
    FirstName varchar(20) not null,  
    SecondName varchar(20) not null,  
    PhoneNumber varchar(10) unique not null,  
    Email varchar(255) unique not null,  
    primary key(id));
```

Таблица Airport:

```
CREATE TABLE Airport(  
    id int auto_increment not null,  
    NameAirport varchar(80) not null,  
    Location varchar(20) not null,  
    primary key(id));
```

Таблица Plane:

```
CREATE TABLE Plane(  
    id int auto_increment not null,  
    NamePlane varchar(20) not null,  
    SerialNumber int not null,  
    AirportID int,  
    primary key(id),  
    foreign key(AirportID) references Airport(id));
```

Таблица Cargo:

```
CREATE TABLE Cargo(  
    id int auto_increment not null,  
    AmountThings int not null,  
    Weight int not null,  
    primary key(id));
```

Таблица Hotel:


```
CREATE TABLE Hotel(  
    id int auto_increment not null,  
    NameHotel varchar(30) not null,  
    PricePerNight int not null,  
    Stars int not null,  
    PhoneNumber varchar(15) unique not null,  
    primary key(id));
```

Таблица Country:

```
CREATE TABLE Country(  
    id int auto_increment not null,  
    NameCountry varchar(30) unique not null,  
    Continent varchar(20) not null,  
    Description varchar(255) not null,  
    primary key(id));
```

Таблица Tour:

```
CREATE TABLE Tour(  
    id int auto_increment not null,  
    CountryID int,  
    OwnerID int,  
    HotelID int,  
    Price int not null,  
    AmountDays int not null,  
    primary key(id),  
    constraint country_tour foreign key(CountryID) references Country(id),  
    constraint customers_tour foreign key(OwnerID) references Customers(id),  
    constraint hote_tour foreign key(HotelID) references Hotel(id));
```

Таблиця Ticket:

```
CREATE TABLE Ticket(  
    id int auto_increment not null,  
    NameLocation varchar(20) not null,  
    price float not null,  
    TourID int,  
    CargoID int,  
    PlaneID int,  
    TimeDeparture datetime,  
    primary key(id),  
    constraint ticket_tour foreign key(TourID) references Tour(id),  
    constraint cargo_tour foreign key(CargoID) references Cargo(id),  
    constraint plane_tour foreign key(PlaneID) references Plane(id));
```

Створення тригерів:

1. Тригер для таблиці Customers, запис ім'я користувача з малої літери:

```
drop trigger name_lower;  
create trigger name_lower before  
insert on customers for each row  
set new.FirstName= lower(new.FirstName);
```

Перевіримо роботу тригера, додавши нового користувача:

id	FirstName	SecondName	PhoneNumber	Email
1	FirstName_1	SecondName_1	0971128448	address_1@gmail.com
2	FirstName_2	SecondName_2	0978123558	address_2@gmail.com
3	FirstName_3	SecondName_3	0961123559	address_3@gmail.com
4	FirstName_4	SecondName_4	0961123555	address_4@gmail.com
5	FirstName_5	SecondName_5	0927123354	address_5@gmail.com
6	FirstName_6	SecondName_6	0938223551	address_6@gmail.com
7	FirstName_7	SecondName_7	0930173556	address_7@gmail.com
8	FirstName_8	SecondName_8	0938153555	address_8@gmail.com
9	FirstName_9	SecondName_9	0932223558	address_9@gmail.com
10	FirstName_10	SecondName_10	0942223558	address_10@gmail.com
11	firstname_11	SecondName_11	0938223661	address_11@gmail.com

2. Тригер для таблиці Plane, при видаленні у літака аеропорту, то він автоматично встановлює значення plane.AirportID = 3:

```
drop trigger type_delete;
CREATE TRIGGER type_delete BEFORE DELETE
ON airport FOR EACH ROW
UPDATE plane SET plane.AirportID = 3
WHERE plane.AirportID = OLD.id;
```

Перевіримо роботу тригера, видаливши аеропорт з індексом 4:

1	NamePlane_1	693472	2
2	NamePlane_2	615835	3
3	NamePlane_3	868450	8
4	NamePlane_4	901508	3
5	NamePlane_5	532075	6
6	NamePlane_6	34965	3
7	NamePlane_7	119637	1
8	NamePlane_8	65071	7
9	NamePlane_9	785874	6
10	NamePlane_10	271237	10

3. Тригер для таблиці Tour, встановлює tour.AmountDays = 10:

```
drop trigger hotel_price;
CREATE TRIGGER hotel_price BEFORE update
ON hotel FOR EACH ROW
UPDATE tour SET tour.AmountDays = 10
where tour.HotelID = OLD.id;
```

Перевіримо роботу тригера, оновивши кількість зірок у готеля з індексом 5:

id	CountryID	OwnerID	HotelID	Price	AmountDays
1	2	2	5	410	10
2	1	7	3	315	15
3	1	6	2	610	11
4	7	3	6	119	4
5	8	10	4	55	6
6	4	2	1	178	17
7	9	5	9	4560	50
8	7	3	8	1440	10
9	1	6	6	431	7
10	10	8	2	209	7

4. Тригер для таблиці Customers, при додаванні нового email, шифрує його:

```
drop trigger email_password;  
CREATE TRIGGER email_password BEFORE  
INSERT ON customers FOR EACH ROW  
SET NEW.Email = AES_ENCRYPT(NEW.Email, 'key');
```

Перевіримо роботу тригера, додавши нового користувача:

id	FirstName	SecondName	PhoneNumber	Email
1	FirstName_1	SecondName_1	0971128448	address_1@gmail.com
2	FirstName_2	SecondName_2	0978123558	address_2@gmail.com
3	FirstName_3	SecondName_3	0961123559	address_3@gmail.com
4	FirstName_4	SecondName_4	0961123555	address_4@gmail.com
5	FirstName_5	SecondName_5	0927123354	address_5@gmail.com
6	FirstName_6	SecondName_6	0938223551	address_6@gmail.com
7	FirstName_7	SecondName_7	0930173556	address_7@gmail.com
8	FirstName_8	SecondName_8	0938153555	address_8@gmail.com
9	FirstName_9	SecondName_9	0932223558	address_9@gmail.com
10	FirstName_10	SecondName_10	0942223558	address_10@gmail.com
11	firstname_11	SecondName_11	0938223661	address_11@gmail.com

5. Тригер для таблиці Plane, при створенні нового аеропорту автоматично додає новий літак з цим аеропортом:

```
drop trigger default_plane;  
create trigger default_plane after insert  
on airport for each row  
insert into plane values(null, 'No name', 213, NEW.id);
```

Перевіримо роботу тригера, додавши новий аеропорт:

2.4. Нормалізація бази даних:

Нормалізація - процес зменшення надмірності інформації в таблицях реляційної БД і, як наслідок, побудови оптимальної структури таблиць і зв'язків.

Можна виділити такі основні правила, якими слід керуватися при проектуванні та подальшій нормалізації таблиць бази даних:

Перша нормальна форма:

- 1) Не повинно бути полів, які повторюються.
- 2) Всі атрибути є простими.
- 3) Всі значення є скалярними.

Друга нормальна форма:

- 1) Усі обмеження першої нормальної форми.
- 2) У таблиці повинен бути один первинний ключ.
- 3) Всі атрибути повинні описувати первинний ключ цілісно, а не якусь його частину.

Третя нормальна форма:

- 1) Усі обмеження першої нормальної форми.
- 2) Усі обмеження другої нормальної форми.
- 3) Не повинно бути залежностей одних не ключових атрибутів від інших, всі атрибути залежать тільки від первинного ключа.

Таблиця Customers:

id	Name and Surname		Phone and email	
1	FirstName_1	SecondName_1	0971128448	address_1@gmail.com
2	FirstName_2	SecondName_2	0978123558	address_2@gmail.com
3	FirstName_3	SecondName_3	0961123559	address_3@gmail.com
4	FirstName_4	SecondName_4	0961123555	address_4@gmail.com
5	FirstName_5	SecondName_5	0927123354	address_5@gmail.com
6	FirstName_6	SecondName_6	0938223551	address_6@gmail.com
7	FirstName_7	SecondName_7	0930173556	address_7@gmail.com
8	FirstName_8	SecondName_8	0938153555	address_8@gmail.com
9	FirstName_9	SecondName_9	0932223558	address_9@gmail.com
10	FirstName_10	SecondName_10	0942223558	address_10@gmail.com

Приведемо цю таблицю до 1НФ:

	id	FirstName	SecondName	PhoneNumber	Email
►	1	FirstName_1	SecondName_1	0971128448	address_1@gmail.com
	2	FirstName_2	SecondName_2	0978123558	address_2@gmail.com
	3	FirstName_3	SecondName_3	0961123559	address_3@gmail.com
	4	FirstName_4	SecondName_4	0961123555	address_4@gmail.com
	5	FirstName_5	SecondName_5	0927123354	address_5@gmail.com
	6	FirstName_6	SecondName_6	0938223551	address_6@gmail.com
	7	FirstName_7	SecondName_7	0930173556	address_7@gmail.com
	8	FirstName_8	SecondName_8	0938153555	address_8@gmail.com
	9	FirstName_9	SecondName_9	0932223558	address_9@gmail.com
	10	FirstName_10	SecondName_10	0942223558	address_10@gmail.com

Таблиця Tour:

id	CountryID	OwnerID	HotelID	Price	AmountDays	NameCountry
1	2	2	5	410	7	Brazil
2	1	7	3	315	15	Brazil
3	1	6	2	610	11	Brazil
4	7	3	6	119	4	Brazil
5	8	10	4	55	6	Brazil
6	4	2	1	178	17	Brazil
7	9	5	9	4560	50	Brazil
8	7	3	8	1440	10	Brazil
9	1	6	6	431	7	Brazil
10	10	8	2	209	7	Brazil
1	2	2	5	410	7	China

Приведемо цю таблицю до 2НФ:

	id	CountryID	OwnerID	HotelID	Price	AmountDays
►	1	2	2	5	410	7
	2	1	7	3	315	15
	3	1	6	2	610	11
	4	7	3	6	119	4
	5	8	10	4	55	6
	6	4	2	1	178	17
	7	9	5	9	4560	50
	8	7	3	8	1440	10
	9	1	6	6	431	7
	10	10	8	2	209	7

	id	NameCountry	Continent	Description
►	1	Poland	Eurasia	awesome
	2	Germany	Eurasia	wonderful
	3	Great Britain	Eurasia	nice
	4	USA	North America	beautiful
	5	Brazil	South America	not bad
	6	Mexico	North America	good
	7	Egypt	Africa	wonderful
	8	Tunisia	Africa	hot
	9	China	Eurasia	nice
	10	Congo	Africa	warm

Таблиця Hotel:

	id	NameHotel	FirstName	Customer number
►	1	NameHotel_1	FirstName_1	0971128448
	2	NameHotel_2	FirstName_1	0971128448
	3	NameHotel_3	FirstName_1	0971128448
	4	NameHotel_4	FirstName_1	0971128448
	5	NameHotel_5	FirstName_1	0971128448
	6	NameHotel_6	FirstName_1	0971128448
	7	NameHotel_7	FirstName_1	0971128448
	8	NameHotel_8	FirstName_1	0971128448
	9	NameHotel_9	FirstName_1	0971128448
	10	NameHotel_10	FirstName_1	0971128448

Приведемо цю таблицю до 3НФ:

	id	NameHotel	CustomerID
►	1	NameHotel_1	5
	2	NameHotel_2	5
	3	NameHotel_3	5
	4	NameHotel_4	5
	5	NameHotel_5	5
	6	NameHotel_6	5
	7	NameHotel_7	5
	8	NameHotel_8	5
	9	NameHotel_9	5
	10	NameHotel_10	5

	id	FirstName	Customer number
►	1	FirstName_1	0971128448
	2	FirstName_2	0978123558
	3	FirstName_3	0961123559
	4	FirstName_4	0961123555
	5	FirstName_5	0927123354
	6	FirstName_6	0938223551
	7	FirstName_7	0930173556
	8	FirstName_8	0938153555
	9	FirstName_9	0932223558
	10	FirstName_10	0942223558

Таблиця Ticket:

	id	NameLocation	price	TourID	CargoID	PlaneID	TimeDeparture	NamePlane
►	1	NameLocation_1	654	3	3	1	2021-05-02 00:00:00	NamePlane_1
	2	NameLocation_2	700	1	2	7	2021-05-12 00:00:00	NamePlane_1
	3	NameLocation_3	112	2	10	9	2021-05-18 00:00:00	NamePlane_1
	4	NameLocation_4	81	3	3	3	2021-05-17 00:00:00	NamePlane_1
	5	NameLocation_5	74	5	10	9	2021-05-02 00:00:00	NamePlane_1
	1	NameLocation_1	654	3	3	1	2021-05-02 00:00:00	NamePlane_2
	2	NameLocation_2	700	1	2	7	2021-05-12 00:00:00	NamePlane_2
	3	NameLocation_3	112	2	10	9	2021-05-18 00:00:00	NamePlane_2
	4	NameLocation_4	81	3	3	3	2021-05-17 00:00:00	NamePlane_2
	5	NameLocation_5	74	5	10	9	2021-05-02 00:00:00	NamePlane_2
	1	NameLocation_1	654	3	3	1	2021-05-02 00:00:00	NamePlane_3
	2	NameLocation_2	700	1	2	7	2021-05-12 00:00:00	NamePlane_3
	3	NameLocation_3	112	2	10	9	2021-05-18 00:00:00	NamePlane_3

Приведемо цю таблицю до 2НФ:

	id	NameLocation	price	TourID	CargoID	PlaneID	TimeDeparture
►	1	NameLocation_1	654	3	3	1	2021-05-02 00:00:00
	2	NameLocation_2	700	1	2	7	2021-05-12 00:00:00
	3	NameLocation_3	112	2	10	9	2021-05-18 00:00:00
	4	NameLocation_4	81	3	3	3	2021-05-17 00:00:00
	5	NameLocation_5	74	5	10	9	2021-05-02 00:00:00

	id	NamePlane
►	1	NamePlane_1
	2	NamePlane_2
	3	NamePlane_3
	4	NamePlane_4
	5	NamePlane_5

3. Реалізація бази даних.

3.1. Виконання над відношеннями операцій реляційної алгебри:

Перед виконанням завдання потрібно сформулювати дві таблиці з однаковими множинами атрибутів. Візьмемо за основу таблицю користувачів Ticket і виконаємо вибір двох множин записів, які перетинаються. Результати збережемо у таблицях Ticket1 та Ticket2.

Таблиця Ticket:

1	NameLocation_1	654	3	3	1	2021-05-02 00:00:00
2	NameLocation_2	700	1	2	7	2021-05-12 00:00:00
3	NameLocation_3	112	2	10	9	2021-05-18 00:00:00
4	NameLocation_4	81	3	3	3	2021-05-17 00:00:00
5	NameLocation_5	74	5	10	9	2021-05-02 00:00:00
6	NameLocation_6	456	9	4	6	2021-05-01 00:00:00
7	NameLocation_7	221	3	4	1	2021-05-21 00:00:00
8	NameLocation_8	410	1	1	1	2021-05-19 00:00:00
9	NameLocation_9	65	8	4	7	2021-05-20 00:00:00
10	NameLocation_10	654	10	7	4	2021-05-13 00:00:00

create table Ticket1 as select id, NameLocation, price, TimeDeparture from Ticket where TimeDeparture < '2021-05-17';

Таблиця Ticket1:

	id	NameLocation	price	TimeDeparture
►	1	NameLocation_1	654	2021-05-02 00:00:00
	2	NameLocation_2	700	2021-05-12 00:00:00
	5	NameLocation_5	74	2021-05-02 00:00:00
	6	NameLocation_6	456	2021-05-01 00:00:00
	10	NameLocation_10	654	2021-05-13 00:00:00

create table Ticket2 as select id, NameLocation, price, TimeDeparture from Ticket where TimeDeparture < '2021-05-05';

Таблиця Ticket2:

	id	NameLocation	price	TimeDeparture
►	1	NameLocation_1	654	2021-05-02 00:00:00
	5	NameLocation_5	74	2021-05-02 00:00:00
	6	NameLocation_6	456	2021-05-01 00:00:00

1. Запит на виконання об'єднання Ticket1 та Ticket2:

select * from Ticket1 union select * from Ticket2;

	id	NameLocation	price	TimeDeparture
►	1	NameLocation_1	654	2021-05-02 00:00:00
	2	NameLocation_2	700	2021-05-12 00:00:00
	5	NameLocation_5	74	2021-05-02 00:00:00
	6	NameLocation_6	456	2021-05-01 00:00:00
	10	NameLocation_10	654	2021-05-13 00:00:00

2. Запит на виконання перетину:

select * from Ticket1 where id in (select id from Ticket2);

	id	NameLocation	price	TimeDeparture
►	1	NameLocation_1	654	2021-05-02 00:00:00
	5	NameLocation_5	74	2021-05-02 00:00:00
	6	NameLocation_6	456	2021-05-01 00:00:00

3. Запит на виконання різниці Ticket1 та Ticket2:

select * from Ticket1 where id not in (select id from Ticket2);

	id	NameLocation	price	TimeDeparture
►	2	NameLocation_2	700	2021-05-12 00:00:00
	10	NameLocation_10	654	2021-05-13 00:00:00

4. Запит на виконання декартового добутку двох таблиць:

```
select * from Ticket1, Ticket2;
```

	id	NameLocation	price	TimeDeparture		id	NameLocation	price	TimeDeparture
	1	NameLocation_1	654	2021-05-02 00:00:00	5	NameLocation_5	74	2021-05-02 00:00:00	
	1	NameLocation_1	654	2021-05-02 00:00:00	6	NameLocation_6	456	2021-05-01 00:00:00	
	2	NameLocation_2	700	2021-05-12 00:00:00	1	NameLocation_1	654	2021-05-02 00:00:00	
	2	NameLocation_2	700	2021-05-12 00:00:00	5	NameLocation_5	74	2021-05-02 00:00:00	
	2	NameLocation_2	700	2021-05-12 00:00:00	6	NameLocation_6	456	2021-05-01 00:00:00	
	5	NameLocation_5	74	2021-05-02 00:00:00	1	NameLocation_1	654	2021-05-02 00:00:00	
	5	NameLocation_5	74	2021-05-02 00:00:00	5	NameLocation_5	74	2021-05-02 00:00:00	
	5	NameLocation_5	74	2021-05-02 00:00:00	6	NameLocation_6	456	2021-05-01 00:00:00	
	6	NameLocation_6	456	2021-05-01 00:00:00	1	NameLocation_1	654	2021-05-02 00:00:00	
	6	NameLocation_6	456	2021-05-01 00:00:00	5	NameLocation_5	74	2021-05-02 00:00:00	
	6	NameLocation_6	456	2021-05-01 00:00:00	6	NameLocation_6	456	2021-05-01 00:00:00	
	10	NameLocation_10	654	2021-05-13 00:00:00	1	NameLocation_1	654	2021-05-02 00:00:00	
	10	NameLocation_10	654	2021-05-13 00:00:00	5	NameLocation_5	74	2021-05-02 00:00:00	
	10	NameLocation_10	654	2021-05-13 00:00:00	6	NameLocation_6	456	2021-05-01 00:00:00	

Висновок: я навчився створювати ділову та концептуальну модель бази даних, визначати логічну структуру кожної таблиці бази даних, навився прописувати обмежування для кожної таблиці бази даних, нормалізувати таблиці до 1 та 2 НФ та виконувати різні види запитів між таблицями такі, як: об'єднання, перетин, різниця, декартовий добуток.

СПИСОК ЛІТЕРАТУРИ

- 1) <https://www.mysqltutorial.org>
- 2) https://uk.wikipedia.org/wiki/Нормалізація_баз_даних
- 3) <https://dev.mysql.com/doc/>
- 4) <https://sql-academy.org>