

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА  
ФРАНКА  
ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ**

**Кафедра цифрової економіки та бізнес-аналітики**

**КУРСОВА РОБОТА**

**з навчальної дисципліни „Проектування та адміністрування БД і СД”**

на тему:

**„Інформаційна система для управління рейсами літаків”**

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

**Спеціальність:** 051 «Економіка»

**Спеціалізація:** «Інформаційні технології в бізнесі»

**Освітній ступінь:** бакалавр

**Науковий керівник:**

к.ф.-м.н., доц. Депутат Б.Я.

(науковий ступінь, посада, прізвище,  
ініціали)

\_\_\_\_\_ “\_\_\_” травня

2021 р. (підпис)

**Виконавець:**

Студент групи УФЕ-31с

Особа Є.Ю.

\_\_\_\_\_ “\_\_\_” травня

2021 р. (підпис)

**Загальна кількість балів** \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

(підпис, ПІП членів комісії)

**ЛЬВІВ 2021**

## ЗМІСТ

<b>ВСТУП.....</b>	<b>3</b>
<b>РОЗДІЛ І. АНАЛІЗ ВИМОГ .....</b>	<b>6</b>
<b>1.1Постановка завдання.....</b>	<b>6</b>
<b>1.2Розробка моделі варіантів використання бази даних; .....</b>	<b>7</b>
<b>1.3Аналіз засобів реалізації (техніко-економічне обґрунтування вибору) .....</b>	<b>13</b>
<b>РОЗДІЛ ІІ. РОЗРОБКА БАЗИ ДАНИХ .....</b>	<b>17</b>
<b>2.1 Опис моделі даних.....</b>	<b>17</b>
<b>2.1.1 Побудова ER-діаграми .....</b>	<b>19</b>
<b>2.3Визначення типів даних.....</b>	<b>23</b>
<b>2.4 Реалізація SQL-скрипту.....</b>	<b>24</b>
<b>ВИСНОВКИ .....</b>	<b>28</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>29</b>
<b>ДОДАТКИ.....</b>	<b>30</b>

## ВСТУП

**Актуальність теми дослідження.** На сьогоднішній день неможливо уявити наше життя без авіаперельотів. Вони відіграють ключову роль в житті будь якої країни. З роками кількість авіаперельотів стрімко збільшилася, тобто збільшились самі пасажиропотоки, як наслідок, авіаперевезення стало в ролі громадського транспорту.

Упродовж 2019 року пасажирські та вантажні перевезення здійснювали 29 вітчизняних авіакомпаній, якими загалом виконано 103,3 тис. комерційних рейсів (за 2018 рік - 100,2 тис. рейсів). У 2019 році ринок пасажирських авіаперевезень продовжував демонструвати позитивну динаміку. За статистичними даними кількість пасажирів, що скористались послугами українських авіакомпаній, збільшилась на 9,4 відсотка та склала 13705,8 тис. чоловік. Пасажирські перевезення протягом року здійснювали 18 вітчизняних авіаперевізників, серед яких найбільші обсяги виконано авіакомпаніями «Міжнародні авіалінії України», «Азур Ейр Україна», «Скайап», «Роза вітрів» та «Буковина». За звітний рік п'ятьма провідними авіакомпаніями загалом перевезено 13306,7 тис. чол., що на 22,4 відсотка більше, ніж за 2018 рік та складає 97 відсотків від загальних обсягів пасажирських перевезень українських авіакомпаній.

Більше половини (51,9 відсотка) всіх пасажирських перевезень вітчизняних авіакомпаній складають міжнародні регулярні перевезення. У 2019 році міжнародні регулярні пасажирські перевезення відповідно до розкладу руху здійснювали 10 вітчизняних авіакомпаній до 46 країн світу. Кількість пасажирів, які скористались послугами українських компаній, зросла на 4,6 відсотка та становила 7107,2 тис. чоловік, при цьому середній відсоток пасажирського завантаження міжнародних регулярних рейсів збільшився на 2,1% та склав 80,9%.

Водночас було потрібно розширення діяльності на українському ринку іноземних авіакомпаній, послугами яких скористались 9422,5 тис. пасажирів, що на 37,4 відсотка перевищує показник 2018 року та складає 57 відсотків від загальних обсягів регулярних пасажирських перевезень між Україною та країнами світу. Загалом регулярні пасажирські перевезення до України виконували 40 іноземних авіакомпаній (у тому числі чотири нові – австрійська авіакомпанія «Laudamotion, французька «Aigle Azur» (здійснювала польоти до вересня 2019 року), ізраїльська «Israir Airlines» та норвезька «Scandinavian Airlines System») з 37 країн світу. Протягом року іноземними авіаперевізниками було відкрито 29 нових маршрутів, з них 21 новий маршрут - авіакомпаніями «Ryanair» та «Wizz Air Hungary».

На сьогоднішній день з стрімким розвитком прогресу аеропорти збільшили свою пропускну здатність пасажиропотоку на доволі значний відсоток. Багатомільйонний потік пасажирів повинен бути чітко розподілений на загальноприйнятій підгрупи для забезпечення оптимальної швидкості і якості обслуговування кожного клієнта до відповідних встановлених правил. Інформаційна система для управління рейсами літаків дає змогу потенційному клієнту вибрати для себе найбільш вигідний спосіб добирання до назначеного пункту призначення. Оцінити усі можливі варіанти перельоту, включаючи ціну, час та швидкість.

**Мета і завдання дослідження.** Метою курсової роботи є дослідження та аналіз актуальності інформаційної системи для управління рейсами літаків.

Для досягнення мети в роботі поставлено й вирішено такі теоретичні та практичні завдання:

- визначити особливості створення бази даних для управління рейсами літаків;
- розробити інформаційну систему, що дає змогу вибрати найбільш вигідний варіант перельоту від пункту А до пункту В;

- перевірити програмний продукт на наявність дефектів та знайти шляхи покращення.

*Об'єктом дослідження* виступає конкретний аеропорт, для якого створюється інформаційна система та процес розробки відповідного програмного забезпечення.

*Предметом* є теоретичні, методичні та практичні аспекти розробки програмного забезпечення мовою запитів: MySQL, для створення інформаційної системи для управління рейсами літаків.

Практичне значення отриманих результатів. Програма може використовуватись фізичною особою чи особами, які хочуть придбати авіаквиток з певною метою.

Використане програмне забезпечення. Для створення програмного продукту використовувалось середовище розробки(MySql) та певні мови програмування.

Структура роботи. Курсова робота складається з трьох розділів («Аналіз вимог» та «Розробка бази даних» висновків, списку використаних джерел та додатків. Загальний обсяг роботи – 30 сторінок.

## РОЗДІЛ I. АНАЛІЗ ВИМОГ

### 1.1 Постановка завдання

Основне завдання в даній роботі полягає в тому, створення бази даних для управління рейсами літаків необхідно проаналізувати вхідні й вихідні документи, щоб виділити сутності, зв'язку між ними й користувачів. Під вхідними документами розуміються накладні, по яких вивозять продукцію для виробництва товару. Під вихідними документами розуміються накладні, по яких партнери одержують товар.

Користувача цієї інформаційної системи будуть комірник, директор, менеджер. Інформацію зі складу буде вносити комірник, не маючи при цьому допуску до інших даних. Інформацію про партнерів, виробників і продукцію буде вносити менеджер. Директор може тільки переглядати всю інформацію.

Для розробки інформаційної системи «Літак» необхідні наступні дані:

- Інформація про продукцію: назва, ціна, тип.
- Інформація про партнерів: назва магазину, адреса, місто, телефон.
- Інформація про виготовлювача: виготовлювач.

Головною перевагою БД є швидкість внесення та використання потрібної інформації. Завдяки спеціальним алгоритмам, які використовуються для баз даних, можна легко знаходити необхідні дані всього за декілька секунд.

Головною задачею є те, щоб клієнт міг швидко та зручно забронювати квиток на бажаний рейс. В базах даних повинно бути представлено чітко і правильно сформоване замовлення клієнта з усіма додатковими даними обраними клієнтом. Перелік усіх доступних авіарейсів повинен відображатися автоматично за усіма установленими нормами. Клієнт повинен отримати усі дані про замовлений ним рейс такі як:

- номер рейсу
- точка відправлення
- точка прибуття
- час відправлення
- час прибуття
- ціна квитка

Також клієнт повинен бути проінформований про запізнення або відміну рейсу. Потенційний клієнт сам вибирає коли замовляти або скасовувати рейс. Усі дані в інформаційній базі даних повинні відповідати установленим умовам, а саме:

- чіткості
- правильності написання
- точності
- компактності

## **1.2 Розробка моделі варіантів використання бази даних;**

За технологією обробки даних бази даних поділяються на централізовані й розподілені. Централізована база даних зберігається у пам'яті однієї обчислювальної системи.

Класифікація БД за моделлю даних:

- Ієрархічні - бази даних можуть бути представлені як дерево, що складається з об'єкти даних можуть бути представлені як дерево,

що складається з об'єктів різних рівнів. Верхній рівень займає один об'єкт, другий - об'єкти другого рівня і т.д.

Між об'єктами існують зв'язки, кожен об'єкт може включати в себе декілька об'єктів більш низького рівня. Такі об'єкти перебувають у відношенні предка (об'єкт більш близький до кореня) до нащадка (об'єкт більш низького рівня), при цьому можлива ситуація, коли об'єкт-предок не має нащадків або має їх декілька, тоді як у об'єкта-нащадка обов'язково тільки один предок. Об'єкти, що мають загального предка, називаються близнюками. Об'єкти різних рівнів. Верхній рівень займає один об'єкт, другий - об'єкти другого рівня і т.д.

- Мережеві - бази даних подібні до ієрархічних, за винятком того, що в них є покажчики в обох напрямках, які з'єднують споріднену інформацію.

До основних понять мережевої моделі бази даних відносяться: рівень, елемент (вузол), зв'язок.

- Вузол - це сукупність атрибутів даних, що описують деякий об'єкт. На схемі ієрархічного дерева вузли представляються вершинами графа. У мережній структурі кожен елемент може бути пов'язаний з будь-яким іншим елементом.

Незважаючи на те, що ця модель вирішує деякі проблеми, пов'язані з ієрархічною моделлю, виконання простих запитів залишається досить складним процесом.



Також, оскільки логіка процедури вибірки даних залежить від фізичної організації цих даних, то ця модель не є повністю незалежною від програми. Іншими словами, якщо необхідно змінити структуру даних, то потрібно змінити і додаток.

- Реляційні - модель орієнтована на організацію даних у вигляді двовимірних таблиць. Кожна реляційна таблиця являє собою двовимірний масив і має наступні властивості:
  - кожен елемент таблиці - один елемент даних;
  - всі осередки в стовпчику таблиці однорідні, тобто всі елементи в стовпчику мають однаковий тип (числовий, символічний тощо);
  - кожен стовпчик має унікальне ім'я;
  - однакові рядки в таблиці відсутні;
  - порядок проходження рядків і стовпчиків може бути довільним.
- об'єктні - СУБД ідеально підходить для інтерпретації складних даних, на відміну від реляційних СУБД, де додавання нового типу даних досягається ціною втрати продуктивності або за рахунок різкого збільшення термінів і вартості розробки додатків. Об'єктна база, на відміну від реляційної, не вимагає модифікації ядра при додаванні нового типу даних. Новий клас і його екземпляри просто надходять у зовнішні структури бази даних. Система управління ними залишається без змін.
- об'єктно-орієнтовані - база даних (ООБД) - база даних, в якій дані оформлені у вигляді моделей об'єктів, що включають прикладні програми, які управляються зовнішніми подіями. Результатом поєднання можливостей (особливостей) баз даних і можливостей об'єктно-орієнтованих мов програмування є об'єктно-орієнтовані

системи управління базами даних (ООСУБД). ООСУБД дозволяють працювати з об'єктами баз даних також, як з об'єктами у програмуванні в об'єктно-орієнтованих мовах програмування. ООСУБД розширює мови програмування, прозоро вводячи довготривалі дані, управління паралелізмом, відновлення даних, асоційовані запити й інші можливості.

- Об'єктно-орієнтовані бази даних звичайно рекомендовані для тих випадків, коли потрібна високопродуктивна обробка даних, які мають складну структуру.

Система, яка забезпечує об'єктну інфраструктуру і набір реляційних розширювачів, називається "об'єктно-реляційною".

- об'єктно-реляційні - системи поєднують переваги сучасних об'єктно-орієнтованих мов програмування з такими властивостями реляційних систем як множинні представлення даних і високорівневі непроцедурні мови запитів.

Під час роботи використовувалась реляційна база даних, оскільки робота з нею дає змогу сформувати двовимірну таблицю, яка являє собою двовимірний масив, дає змогу встановити зв'язки між таблицями за допомогою з'єднання, а саме:

- один-до-одного - має місце, коли одному запису в батьківській таблиці відповідає один запис в дочірній. Це відношення зустрічається набагато рідше, ніж відношення «один-до-багатьох». Його використовують, якщо не хочуть, щоб таблиця БД

«розпухала» від другорядної інформації, проте для читання зв'язаної інформації в декількох таблицях доводиться проводити ряд операцій читання замість однієї, коли дані зберігаються в одній таблиці.

- один-до-багатьох - має місце, коли одному запису батьківської таблиці може відповідати декілька записів дочірньої.

Зв'язок один-до-багатьох іноді називають зв'язком багато-до-одного. І у тому, і в іншому випадку суть зв'язку між таблицями залишається незмінною. Зв'язок один-до-багатьох є найпоширенішим для реляційних баз даних. Вона дозволяє моделювати також ієрархічні структури даних.

- багато-до-багатьох - застосовується в наступних випадках:
  1. одному запису в батьківській таблиці відповідає більш за один запис в дочірній;
  2. одному запису в дочірній таблиці відповідає більш за один запис в батьківській.

Будь-який зв'язок багато-до-багатьох в реляційній базі даних необхідно замінити на зв'язок один-до-багатьох (одну або більш) за допомогою введення додаткових таблиць.

Процес розробки бази даних являє собою складний процес проектування відображення опису предметної області у схему внутрішньої моделі даних. Перебіг цього процесу є послідовністю більш простих процесів проектування менш складних відображень. Ця послідовність у процесі проектування весь час уточнюється, вдосконалюється таким чином, щоб були визначені об'єкти, їх властивості та зв'язки, які будуть потрібні майбутнім користувачам системи.

Для реалізації серверної частини було обрано СУБД MySQL. Для управління базами даних використовується програма PHPMyAdmin.

PHPMysqlAdmin – це сукупність скриптів написаних на PHP, що забезпечують повноцінну, в тому числі віддалену роботу з базами даних MySQL, через вебінтерфейс. Основним етапом створення таблиць вважається вибір кодування. Адже від нього залежить правильне відображення даних на веб-сторінці.

У процесі розробки проекту у базі даних «Store» створено таблиці:

PROVIDER – таблиця, яка містить дані про постачальників деталей

DETAILS – таблиця, яка містить дані про наявні деталі

EMPLOYEES – таблиця, яка містить дані про працівників фірми

CLIENTS – таблиця, яка містить дані про клієнтів фірми

SERVICE – таблиця, яка містить дані про послуги які надає фірма

На даному рисунку(1.1) зображена реляційна база даних

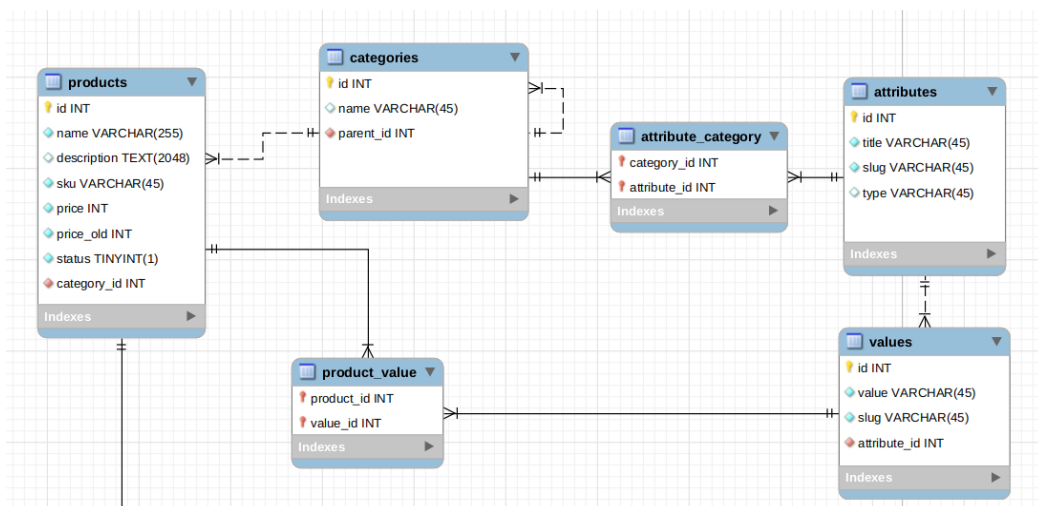


Рисунок 1.1. Реляційна база даних

### 1.3 Аналіз засобів реалізації (техніко-економічне обґрунтування вибору)

Даний проект було виконано на представлених технологіях, а саме MySQL.

MySQL — вільна система керування реляційними базами даних, яка була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

Характеристики MySQL:

- Клієнтська програма MySQL являє собою утиліту командного рядка. Ця програма підключається до сервера по мережі. Команди, що виконуються сервером, зазвичай пов'язані з читанням і записом даних на жорсткому диску.
- Клієнтські програми можуть працювати не тільки в режимі командного рядка. Є і графічні клієнти, наприклад MySQL GUI, phpMyAdmin і ін.
- MySQL взаємодіє з базою даних на мові, званому SQL (Structured Query Language - мова структурованих запитів).
- SQL призначений для маніпуляції даними, які зберігаються в тих системах керування базами даних (RDBMS). SQL має команди, за допомогою яких дані можна отримувати, сортувати, оновлювати, видаляти і додавати. Стандарти мови SQL визначає ANSI (American

National Standards Institute). В даний час діє стандарт, прийнятий в 2003 році (SQL-3).

- SQL можна використовувати з такими RDBMS як MySQL, mSQL, PostgreSQL, Oracle, Microsoft SQL Server, Access, Sybase, Ingres. Ці системи RDBMS підтримують всі важливі і загальноприйняті оператори SQL, проте кожна з них має безліч своїх власних патентованих операторів і розширень.
- SQL є спільною мовою запитів для декількох баз даних різних типів. Даний курс розглядає систему MySQL, яка є RDBMS с відкритим вихідним кодом, доступною для завантаження на сайті MySQL.com.

Основні переваги MySQL. Серед основних переваг MySQL відзначають наступні:

- Масштабованість. MySQL може підтримувати роботу БД значних розмірів, що підтверджують її реалізації в Yahoo!, Google, HP, Associated Press. Згідно документації, що додається до MySQL, деякі БД, що використовуються компанією MySQL AB (розробником MySQL), зберігають до 50 млн. записів.
- Переносність. MySQL працює на різних платформах, серед яких Unix, Linux, Windows, OS/2, Solaris, Mac OS. Окрім того, MySQL працює на різних платформах.

- Зв'язаність. MySQL має мережеву структуру. До MySQL можна одержувати доступ із будь-якої точки Internet кільком користувачам одночасно. MySQL має цілий ряд програмних інтерфейсів додатків (Application Programming Interface –API ), які дозволяють встановлювати з'єднання з MySQL із додатків, написаних на таких мовах як C, C++, Perl, PHP, Java, Python.
- Безпека. MySQL має систему контролю доступу до даних, забезпечує шифрування даних при передаванні.
- Швидкість функціонування.
- Зручність експлуатації. MySQL досить зручно встановлюється та реалізується, легко адмініструється.
- Відкритий код.

На рисунку(1.2) зображено меню MySql





## РОЗДІЛ II. РОЗРОБКА БАЗИ ДАНИХ

### 2.1 Опис моделі даних

Проектування бази даних складається з декількох етапів і починається з попередньої структуризації предметної області. Насамперед, необхідно виділити всі об'єкти, які будуть використатися в базі даних, указати їхньої властивості (характеристики) і встановити зв'язку між ними. Цей етап називають концептуальним проектуванням бази даних.

Початковою стадією проектування системи баз даних є побудова семантичної моделі предметної області, що базується на аналізі властивостей і природи об'єктів предметної області.

Для опису предметної області використовують три основних конструктивних елементи сутність, атрибут і зв'язок.

Сутність – засіб подання предметної області систем, орієнтованих на обробку фактографічної інформації. Сутність визначається своїм унікальним ім'ям і переліком атрибутів, що характеризують властивості сутності.

Атрибут – це поійменована характеристика сутності, що приймає значення з деякої безлічі припустимих значень.

Зв'язок – це графічно зображувана асоціація, призначена для позначення виділеного відношення між двома або більше сутностями.

У даній інформаційній системі основними об'єктами є:

- літак
- рейс
- точка прибуття
- робочі дні

В MySql даний запис “літак” зображено на рисунку(2.1)

```

INSERT INTO PLANES(ID_P, NAME, SIZE, ID_R)
VALUES(1, 'BOEING 737', '250', 1)
VALUES(2, 'A-321', '250', 2)
VALUES(3, 'TY-214', '250', 3)
VALUES(4, 'AH-140', '250', 4)
VALUES(5, 'A-130', '250', 5)
VALUES(6, 'B730', '250', 6)
VALUES(7, 'A320', '250', 7)
VALUES(8, 'CRJ9', '250', 8)
VALUES(9, 'E195', '250', 9)
VALUES(10, 'B738', '250', 10)

```

Рисунок 2.1. Дані про літак

В даній таблиці можна вносити корективи або зміни відповідно до вимог за для дотримання коректного зв'язку між таблицями і проведення чіткої структурованої взаємодії між відповідними даними.

В MySql даний запис “рейс” зображено на рисунку(2.2)

```

INSERT INTO RACE(ID_R, CODE, ID_P)
VALUES(1, 'PS33', 1)
VALUES(2, 'PS561', 2)
VALUES(3, 'PS45', 3)
VALUES(4, 'SU4435', 4)
VALUES(5, 'PS57', 5)
VALUES(6, 'S7121', 6)
VALUES(7, 'SU1805', 7)
VALUES(8, 'KC402', 8)
VALUES(9, 'PS35', 9)
VALUES(10, 'PQ8145', 10)

```

Рисунок 2.2. Дані про рейс

В MySql даний запис “точка прибуття” зображено на рисунку(2.3)

```

INSERT INTO SCHEDULES(ID_R,DAYS, TYPE)
VALUES(1, 'MONDAY', 'ARRIVE')
VALUES(2, 'WEDNESDAY', 'ARRIVE')
VALUES(3, 'THURSDAY', 'ARRIVE')
VALUES(4, 'FRIDAY', 'ARRIVE')
VALUES(5, 'FRIDAY', 'ARRIVE')
VALUES(6, 'TUESDAY', 'ARRIVE')
VALUES(7, 'THURSDAY', 'ARRIVE')
VALUES(8, 'FRIDAY', 'ARRIVE')
VALUES(9, 'SUNDAY', 'ARRIVE')
VALUES(10, 'SATURDAY', 'ARRIVE')

```

Рисунок 2.3. Дані про точку прибуття

В MySQL даний запис “робочі дні” зображено на рисунку(2.4)

```

INSERT INTO DIRECTION(ID_D, NAME, ID_R,ID_P)
VALUES(1, 'LVIV', 1,1)
VALUES(2, 'PARIS', 2,2)
VALUES(3, 'KYIV', 3,3)
VALUES(4, 'VENICE', 4,4)
VALUES(5, 'BERLIN', 5,5)
VALUES(6, 'MOSCOW', 6,6)
VALUES(7, 'BARCELONA', 7,7)
VALUES(8, 'MADRID', 8,8)
VALUES(9, 'MILAN', 9,9)
VALUES(10, 'LONDON', 10,10)

```

Рисунок 2.4. Дані про робочі дні

### 2.1.1 Побудова ER-діаграми

Найпоширенішою формою зображення інформаційно-логічної моделі служать ER-діаграми, які повинні доповнюватися докладною інформацією, а також давати можливість керувати наочним поданням деталей моделі. Для цього використовуються наступні позначення:

1. Сутності зображуються прямокутниками.
2. Атрибути позначаються овалами.
3. Зв'язки зображуються ромбами.

## 2.2 Нормалізація відношень

Нормалізація схеми бази даних — покроковий процес розбиття одного відношення (на практиці: таблиці) відповідно до алгоритму нормалізації на декілька відношень на базі функціональних залежностей.

Нормальна форма — властивість відношення в реляційній моделі даних, що характеризує його з точки зору надмірності, яка потенційно може призвести до логічно помилкових результатів вибірки або зміни даних. Нормальна форма визначається як сукупність вимог, яким має задовольняти відношення.

Таким чином, схема реляційної бази даних переходить у першу, другу, третю і так далі нормальні форми. Якщо відношення відповідає критеріям нормальної форми  $n$  та всіх попередніх нормальних форм, тоді вважається, що це відношення знаходиться у нормальній формі рівня  $n$ .

- Перша нормальна форма

Перша нормальна форма (1НФ, 1NF) утворює ґрунт для структурованої схеми бази даних:

Кожна таблиця повинна мати основний ключ: мінімальний набір колонок, які ідентифікують запис.

Уникнення повторень груп (категорії даних, що можуть зустрічатись різну кількість разів в різних записах) правильно визначаючи неключові атрибути.

Атомарність: кожен атрибут повинен мати лише одне значення, а не множину значень.

- Друга нормальна форма

Друга нормальна форма (2НФ, 2NF) вимагає, аби дані, що зберігаються в таблицях із композитним ключем, не залежали лише від частини ключа:

Схема бази даних повинна відповідати вимогам першої нормальної форми.

Дані, що повторно з'являються в декількох рядках, виносяться в окремі таблиці.

- Третя нормальна форма

Третя нормальна форма (3НФ, 3NF) вимагає, аби дані в таблиці залежали винятково від основного ключа:

Схема бази даних повинна відповідати всім вимогам другої нормальної форми.

Будь-яке поле, що залежить від основного ключа та від будь-якого іншого поля, має виноситись в окрему таблицю.

Нормальна форма Бойса — Кодда

Відношення знаходиться в НФБК тоді і лише тоді, коли детермінант кожної функціональної залежності є потенційним ключем. Якщо це правило не виконується, то, щоб привести вказане відношення до НФБК, його слід розділити на два відношення шляхом двох операцій проєкції на кожну функціональну залежність, детермінант якої не є потенційним ключем:

Проекція без атрибутів залежної частини такої функціональної залежності;

Проекція на всі атрибути цієї функціональної залежності.

Визначення НФБК не потребує жодних умов попередніх нормальних форм. Якщо проводити нормалізацію послідовно, то в переважній більшості випадків при досягненні 3НФ автоматично будуть задовольнятися вимоги

НФБК. 3НФ не збігається з НФБК лише тоді, коли одночасно виконуються такі 3 умови:

Відношення має 2 або більше потенційних ключів.

Ці потенційні ключі складені (містять більш ніж один атрибут)

Ці потенційні ключі перекриваються, тобто мають щонайменше один спільний атрибут.

- Четверта нормальна форма (4НФ, 4NF) потребує, аби в схемі баз даних не було нетривіальних багатозначних залежностей множин атрибутів від будь чого, окрім надмножини ключа-кандидата. Вважається, що таблиця знаходиться у 4НФ тоді і лише тоді, коли вона знаходиться в НФБК та багатозначні залежності є функціональними залежностями. Четверта нормальна форма усуває небажані структури даних — багатозначні залежності.
- П'ята нормальна форма

П'ята нормальна форма (5НФ, 5NF, PJ/NF) вимагає, аби не було нетривіальних залежностей об'єднання, котрі б не витікали із обмежень ключів. Вважається, що таблиця в п'ятій нормальній формі тоді і лише тоді, коли вона знаходиться в 4НФ та кожна залежність об'єднання зумовлена її ключами-кандидатами.

Нормальна форма домен/ключ

Докладніше: Доменно-ключова нормальна форма

Ця нормальна форма вимагає, аби в схемі не було інших обмежень окрім ключів та доменів.

- Шоста нормальна форма

Таблиця знаходиться у 6NF, якщо вона знаходиться у 5NF та задовольняє вимозі відсутності нетривіальних залежностей. Зазвичай 6NF ототожнюють з DKNF.

## 2.3 Визначення типів даних

Тип даних — характеристика, яку явно чи неявно надано об'єкту (змінній, функції, полю запису, константі, масиву тощо). Тип даних визначає множину припустимих значень, формат їхнього збереження, розмір виділеної пам'яті та набір операцій, які можна робити над даними.

На рисунку 2.6. зображено взаємозв'язок типів даних.

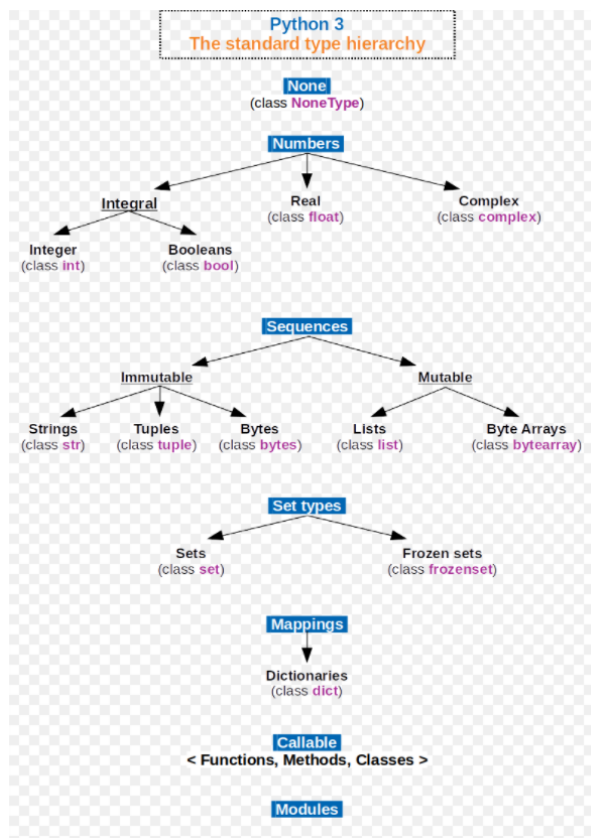


Рисунок 2.6. Модель типів даних

## Діапазони числових типів даних

Кожний числовий тип даних має мінімальне та максимальне значення, яке називають діапазон значень. Важливо знати діапазон значень, особливо, коли працюєш з «маленькими» типами даних, оскільки у них можна зберігати лише значення у вузькому діапазоні. Спроба внести число, більше за доступний діапазон може призвести до помилок періоду компіляції/виконання, або до неправильних підрахунків (через відкидання) залежно від мови програмування, яка використовується.

Діапазон змінних оснований на кількості байтів відведених для збереження значення. Цілі типи даних розрядністю  $n$  бітів зазвичай здатні зберігати  $2n$  значень. Для інших типів даних (напр. дійсних чисел) діапазон заплутаніший, та залежить від методу зберігання інформації у ньому. Існують також типи даних, які не використовують весь байт. Наприклад, для булевого типу, який є двійковим значенням, достатньо лише один біт, але на практиці використовується весь байт, тобто 7 бітів залишаються невикористаними.

## 2.4 Реалізація SQL-скрипту

```
CREATE DATABASE RACES
```

```
USE RACES
```

Створення таблиць

```
CREATE TABLE PLANES(ID_P INT, NAME VARCHAR(30), SIZE  
VARCHAR(30) , ID_R INT, PRIMARY KEY (ID_P))
```



```
CREATE TABLE RACE (ID_R INT, CODE VARCHAR(30), ID_P INT,  
PRIMARY KEY (ID_R))
```

```
CREATE TABLE SHEDULES (ID_R INT, DAYS VARCHAR(30), TYPE  
VARCHAR(30))
```

```
CREATE TABLE DIRECTION (ID_D INT, NAME VARCHAR(30), ID_R  
INT, ID_P INT, PRIMARY KEY(ID_D))
```

Інформація про літак, на якому буде здійснюватися переліт

```
INSERT INTO PLANES(ID_P, NAME, SIZE,ID_R)
```

```
VALUES(1,'BOEING 737', '250', 1)
```

```
VALUES(2,'A-321', '250', 2)
```

```
VALUES(3,'TY-214', '250', 3)
```

```
VALUES(4,'AH-140', '250', 4)
```

```
VALUES(5,'A-130', '250', 5)
```

```
VALUES(6,'B730', '250', 6)
```

```
VALUES(7,'A320', '250', 7)
```

```
VALUES(8,'CRJ9', '250', 8)
```

```
VALUES(9,'E195', '250', 9)
```

```
VALUES(10,'B738', '250', 10)
```

Інформація про рейс перельоту

```
INSERT INTO RACE(ID_R, CODE, ID_P)
VALUES(1,'PS33', 1)
VALUES(2,'PS561', 2)
VALUES(3,'PS45', 3)
VALUES(4,'SU4435', 4)
VALUES(5,'PS57', 5)
VALUES(6,'S7121', 6)
VALUES(7,'SU1805', 7)
VALUES(8,'KC402', 8)
VALUES(9,'PS35', 9)
VALUES(10,'PQ8145', 10)
```

Робочі дні потенційного літака

```
INSERT INTO SHEDULES(ID_R,DAYS, TYPE)
VALUES(1,'MONDAY','ARRIVE')
VALUES(2,'WEDNESDAY','ARRIVE')
VALUES(3,'THURSDAY','ARRIVE')
VALUES(4,'FRIDAY','ARRIVE')
VALUES(5,'FRIDAY','ARRIVE')
```

VALUES(6,'TUESDAY','ARRIVE')

VALUES(7,'THURSDAY','ARRIVE')

VALUES(8,'FRIDAY','ARRIVE')

VALUES(9,'SUNDAY','ARRIVE')

VALUES(10,'SATURDAY','ARRIVE')

Місто, в яке прибуває літак

INSERT INTO DIRECTION(ID\_D, NAME, ID\_R,ID\_P)

VALUES(1,'LVIV',1,1)

VALUES(2,'PARIS',2,2)

VALUES(3,'KYIV',3,3)

VALUES(4,'VENICE',4,4)

VALUES(5,'BERLIN',5,5)

VALUES(6,'MOSCOW',6,6)

VALUES(7,'BARCELONA',7,7)

VALUES(8,'MADRID',8,8)

VALUES(9,'MILAN',9,9)

VALUES(10,'LONDON',10,10)

## ВИСНОВКИ

В процесі виконання поставленого завдання, а саме:

- визначити особливості створення бази даних для управління рейсами літаків;
- розробити інформаційну систему, що дає змогу вибрати найбільш вигідний варіант перельоту від пункту А до пункту В;
- перевірити програмний продукт на наявність дефектів та знайти шляхи покращення.

Поставлена мета була досягнена. Розроблена інформаційна для управління рейсами літаків.

Вибрані засоби для роботи, а саме:

- MySql

Проявили всі позитивні можливості. Починаючи із компактності коду і закінчуючи простотою використання.

Враховуючи швидкість прогресу в авіаційній сфері, усі аеропорти повинні мати власну базу даних з переліком послуг і детальною інформацією. Провівши детальний аналіз опрацьованого матеріалу можна зробити висновок, що інформаційна система працює. Інформація – це основа знання, що дасть змогу потенційним клієнтам зробити правильний вибір.

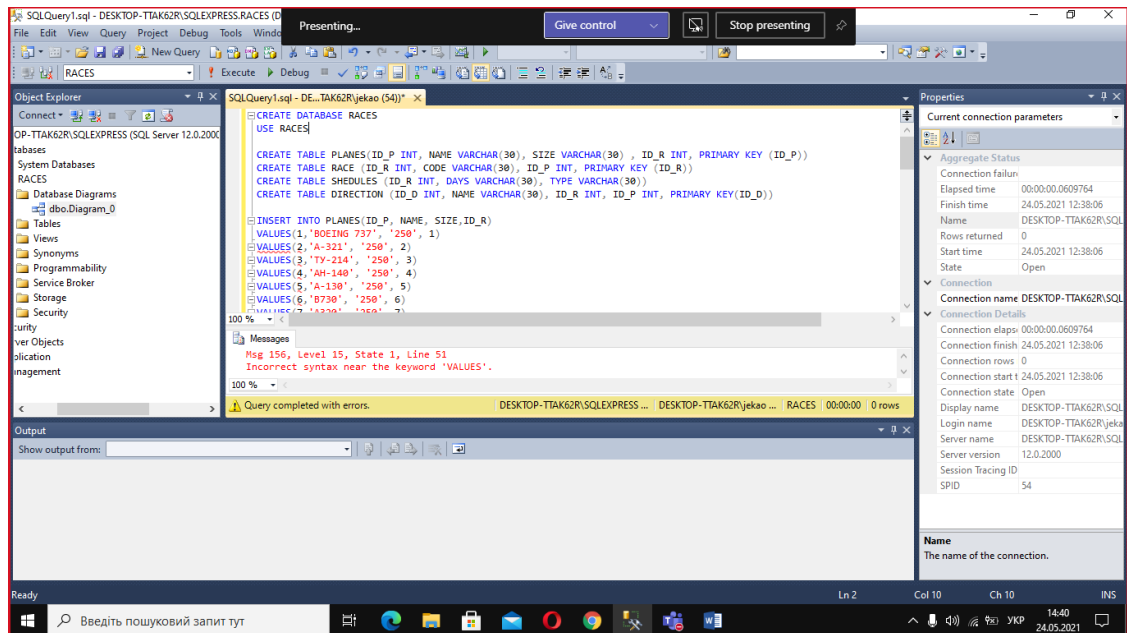
## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- Переваги MySQL [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/560354/page:3/> .
- Типи даних MySQL [Електронний ресурс]. – Режим доступу: [https://uk.wikibooks.org/wiki/SQL/%D0%A2%D0%B8%D0%BF%D0%B8\\_%D0%B4%D0%B0%D0%BD%D0%B8%D%85\\_MySQL](https://uk.wikibooks.org/wiki/SQL/%D0%A2%D0%B8%D0%BF%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D%85_MySQL).
- НОРМАЛІЗАЦІЯ ВІДНОШЕНЬ [Електронний ресурс]. – Режим доступу: [https://life-prog.ru/ukr/1\\_331\\_normalizatsiya-vidnoshen.html](https://life-prog.ru/ukr/1_331_normalizatsiya-vidnoshen.html)
- Переваги PHP [Електронний ресурс]. – Режим доступу: <https://blog.ithillel.ua/ua/articles/chomu-vybyraiut-7-perevah-movuprohramuvannia-php>.
- Моделі даних [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%96\\_%D0%B1%D0%B0%D0%B7\\_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%96_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85).
- НОРМАЛІЗАЦІЯ БАЗ ДАНИХ — ВІКІПЕДІЯ [Електронний ресурс]. – Режим доступу: <https://cutt.ly/tyTCqT8>
- Мартин Грабер :SQL; Видавництво «Лори», 2016. -643 с.

## ДОДАТКИ

Додаток(А)

Код MySql



Додаток(В)

Зв'язки між таблицями

