Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Інститут прикладної математики та фундаментальних наук

****

**Звіт**

До лабораторної роботи №1

З дисципліни “Організація баз даних та знань”

На тему “Створення схеми БД”

**Виконала:**

Студентка групи ПМ-23

Стецюк Анастасія Русланівна

**Прийняв:**

Кандидат фізико-математичних наук, доцент

Пабирівський Віктор Володимирович

Львів 2023

**Мета:** Навчитися створювати концептуальну та логічну модель предметної області, а також створювати базу даних з використанням різних типів обмежень цілісності.

**Завдання 1**

Створити концептуальну та логічну модель предметної області згідно з описом, отриманим в викладача (відповідно до обраної предметної області). Варіант №9 - Аеропорт

* **planes** - таблиця, що містить інформацію про літаки, з наступними полями:
  + id - унікальний ідентифікатор літака (типу INTEGER);
  + board\_number - бортовий номер літака (типу VARCHAR) (UNIQUE, бо номер завжди має бути унікальним);
  + model - модель літака (типу VARCHAR) (NOT NULL, бо у літака має бути модель);
  + worked\_hours - кількість відпрацьованих годин (типу INTEGER);
  + seats - кількість місць в літаку (типу INTEGER);
  + capacity - вантажопідйомність літака (типу INTEGER).
* **flights** - таблиця, що містить інформацію про рейси, з наступними полями:
  + id - унікальний ідентифікатор рейсу (типу INTEGER);
  + departure\_point - пункт відправлення рейсу (типу VARCHAR) (NOT NULL, бо має існувати точка вильоту);
  + destination\_point - пункт призначення рейсу (типу VARCHAR) (NOT NULL, бо має існувати точка посадки);
  + departure\_time - час вильоту (типу TIMESTAMP) (NOT NULL, бо має існувати час вильоту);
  + landing\_time - час посадки (типу TIMESTAMP) (NOT NULL, бо має існувати час посадки);
  + plane\_id - ідентифікатор літака, на якому виконується рейс (типу INTEGER);
  + sold\_tickets\_number - кількість проданих квитків на рейс (типу INTEGER).
* **crew\_members** - таблиця, що містить інформацію про членів екіпажу, з наступними полями:
  + id - унікальний ідентифікатор члена екіпажу (типу INTEGER);
  + last\_name - прізвище члена екіпажу (типу VARCHAR) (NOT NULL, бо у пілота не може не бути прізвища);
  + birth\_date - дата народження члена екіпажу (типу DATE) (NOT NULL, бо оскільки пілот живий, він має мати дату народження);
  + address - адреса члена екіпажу (типу VARCHAR).
* **pilots** - таблиця, що містить інформацію про пілотів, з наступними полями:
  + id - унікальний ідентифікатор пілота (типу INTEGER);
  + crew\_member\_id - ідентифікатор члена екіпажу, який є пілотом (типу INTEGER (UNIQUE NOT NULL, бо у пілота має бути ID та оскільки пілот такий тільки один, то він також має бути унікальним);
  + allowed\_planes - список дозволених літаків для пілота (типу VARCHAR);
  + last\_flight\_date - дата останнього вильоту пілота (типу TIMESTAMP).
* **flight\_crew** - таблиця, що містить інформацію про екіпажі рейсів для реалізації звʼязку багато до багатьох, з наступними полями:
  + flight\_id - ідентифікатор рейсу (типу INTEGER (NOT NULL, бо потрібен ID рейсу для функціонування цієї таблиці);
  + crew\_member\_id - ідентифікатор члена екіпажу, що входить до складу екіпажу рейсу (типу INTEGER) (NOT NULL, бо потрібен ID члена екіпажу для функціонування цієї таблиці);

|  |  |  |  |
| --- | --- | --- | --- |
| **Елементи даних** | **Обʼєкт** | **Тип даних** | **Умова на значення** |
| **ID літака** | Planes | Цілочисельний (INT) | PK |
| Бортовий номер | Planes | Цілочисельний (INT) | UNIQUE NOT NULL |
| Модель | Planes | Стрічка (VARCHAR) | NOT NULL |
| Кількість відпрацьованих годин | Planes | Цілочисельний (INT) | - |
| Кількість місць | Planes | Цілочисельний (INT) | - |
| Вантажопідйомність | Planes | Цілочисельний (INT) | - |
| **ID рейсу** | Flights | Цілочисельний (INT) | PK |
| Пункт відправлення | Flights | Стрічка (VARCHAR) | NOT NULL |
| Пункт призначення | Flights | Стрічка (VARCHAR) | NOT NULL |
| Час вильоту | Flights | Час (TIMESTAMP) | NOT NULL |
| Час посадки | Flights | Час (TIMESTAMP) | NOT NULL |
| ID літака | Flights | Цілочисельний (INT) | FK |
| Кількість проданих квитків | Flights | Цілочисельний (INT) | - |
| **ID члена екіпажу** | Crew\_members | Цілочисельний (INT) | PK |
| Прізвище | Crew\_members | Стрічка (VARCHAR) | NOT NULL |
| Дата народження | Crew\_members | Дата (DATE) | NOT NULL |
| Адреса | Crew\_members | Стрічка (VARCHAR) | - |
| **ID пілота** | Pilot | Цілочисельний (INT) | PK |
| ID члена екіпажу | Pilot | Цілочисельний (INT) | FK, UNIQUE NOT NULL |
| Список дозволених літаків | Pilot | Стрічка (VARCHAR) | - |
| Дата останнього вильоту | Pilot | Дата (DATE) | - |
| **ID польоту** | Flight\_crew | Цілочисельний (INT) | PK, FK |
| **ID члена екіпажу** | Flight\_crew | Цілочисельний (INT) | PK, FK |



**Завдання 2**

Створити базу даних: створити таблиці, задати всі потрібні обмеження, створити зв’язки між таблицями. Схема бази даних повинна містити приклади використання сурогатних ключів створених за допомогою послідовностей (SEQUENCES). Схема бази даних повинна містити приклади використання всіх можливих способів перевірок обмеження цілісності, зокрема зовнішні ключі, вимоги на унікальність полів, перевірки типу CHECK, перевірки типу NULL/NOT NULL, тощо.

База даних буде створена на основі описаної раніше концептуальної та логічної моделі предметної області. Для створення таблиць будуть використані наступні SQL-запити:

CREATE TABLE IF NOT EXISTS table - створити нову таблицю, якщо такої ще не існує, куди передаються поля та їх властивості;

PRIMARY KEY - первинний ключ;

FOREIGN KEY - зовнішній ключ;

AUTO\_INCREMENT - автоматичне збільшення значення;

UNIQUE - значення завжди має бути унікальним;

NOT NULL - значення не має бути пустим.

Створюю таблиці planes, flights, crew\_members, pilots, flight\_crew:

CREATE TABLE IF NOT EXISTS planes (

id INTEGER PRIMARY KEY AUTO\_INCREMENT,

board\_number VARCHAR(10) UNIQUE NOT NULL,

model VARCHAR(50) NOT NULL,

worked\_hours INTEGER,

seats INTEGER,

capacity INTEGER

);

CREATE TABLE IF NOT EXISTS flights (

id INTEGER PRIMARY KEY AUTO\_INCREMENT,

departure\_point VARCHAR(50) NOT NULL,

destination\_point VARCHAR(50) NOT NULL,

departure\_time TIMESTAMP NOT NULL,

landing\_time TIMESTAMP NOT NULL,

plane\_id INTEGER NOT NULL,

sold\_tickets\_number INTEGER,

FOREIGN KEY (plane\_id) REFERENCES planes(id)

);

CREATE TABLE IF NOT EXISTS crew\_members (

id INTEGER PRIMARY KEY AUTO\_INCREMENT,

last\_name VARCHAR(50) NOT NULL,

birth\_date DATE NOT NULL,

address VARCHAR(100)

);

CREATE TABLE IF NOT EXISTS pilots (

id INTEGER PRIMARY KEY AUTO\_INCREMENT,

crew\_member\_id INTEGER UNIQUE NOT NULL,

allowed\_planes VARCHAR(100),

last\_flight\_date TIMESTAMP,

FOREIGN KEY (crew\_member\_id) REFERENCES crew\_members(id)

);

CREATE TABLE IF NOT EXISTS flight\_crew (

flight\_id INTEGER NOT NULL,

crew\_member\_id INTEGER NOT NULL,

PRIMARY KEY (flight\_id, crew\_member\_id),

FOREIGN KEY (flight\_id) REFERENCES flights(id),

FOREIGN KEY (crew\_member\_id) REFERENCES crew\_members(id)

);

**Завдання 3**

Ввести по декілька записів в кожну таблицю, проаналізувати цілісність даних.

Для додавання записів в таблиці використовую такі SQL-запити:

INSERT INTO table (fields) VALUES (values) - додавання даних в певні поля певної таблиці.

INSERT INTO planes (board\_number, model, worked\_hours, seats, capacity)

VALUES

('AA101', 'Boeing 737', 12000, 150, 5000),

('AA102', 'Boeing 737', 10000, 150, 5000),

('AA103', 'Airbus A320', 8000, 180, 6000),

('AA104', 'Airbus A320', 9000, 180, 6000),

('AA105', 'Boeing 787', 5000, 250, 8000),

('AA106', 'Boeing 787', 6000, 250, 8000),

('AA107', 'Airbus A380', 3000, 500, 10000),

('AA108', 'Airbus A380', 3500, 500, 10000);

INSERT INTO flights (departure\_point, destination\_point, departure\_time, landing\_time, plane\_id, sold\_tickets\_number)

VALUES

('Kyiv', 'Paris', '2023-03-20 15:00:00', '2023-03-20 18:00:00', 1, 100),

('Paris', 'Kyiv', '2023-03-21 12:00:00', '2023-03-21 15:00:00', 1, 90),

('Kyiv', 'Amsterdam', '2023-03-22 10:00:00', '2023-03-22 13:00:00', 3, 120),

('Amsterdam', 'Kyiv', '2023-03-23 17:00:00', '2023-03-23 20:00:00', 3, 110),

('Kyiv', 'New York', '2023-03-24 06:00:00', '2023-03-24 14:00:00', 5, 200),

('New York', 'Kyiv', '2023-03-25 16:00:00', '2023-03-26 00:00:00', 5, 180);

INSERT INTO crew\_members (last\_name, birth\_date, address)

VALUES

('Bull', '1980-05-15', 'Kyiv, Ukraine'),

('Whittaker', '1985-10-22', 'Kharkiv, Ukraine'),

('Rees', '1992-03-05', 'Lviv, Ukraine'),

('Peters', '1995-08-01', 'Odessa, Ukraine'),

('Goodwin', '1998-12-18', 'Dnipro, Ukraine');

INSERT INTO pilots (crew\_member\_id, allowed\_planes, last\_flight\_date)

VALUES

(1, 'Boeing 737, Airbus A320', '2022-02-28 10:30:00'),

(2, 'Boeing 737, Boeing 777', '2022-03-01 15:45:00'),

(3, 'Airbus A320, Airbus A380', '2022-02-27 09:15:00');

INSERT INTO flight\_crew (flight\_id, crew\_member\_id)

VALUES

(1, 1),

(1, 2),

(2, 2),

(3, 3),

(4, 1),

(4, 3),

(5, 1),

(5, 2),

(5, 3);

**Висновок:** Під час виконання даної лабораторної роботи я навчилася створювати концептуальну та логічну модель предметної області та переносити її в реляційну модель бази даних. Я детально вивчила можливості використання різних типів обмежень цілісності, таких як зовнішні ключі, вимоги на унікальність полів, перевірки типу CHECK, перевірки типу NULL/NOT NULL, та була здатна їх використовувати під час створення та наповнення таблиць. Також я створила базу даних для авіакомпанії, яка складається з чотирьох таблиць: planes, flights, crew\_members та pilots. Таблиці мають зв'язки між собою, що дозволяє зберігати та обробляти інформацію про літаки, рейси, екіпажі та пілотів.