

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

Пабірівський 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:
 - **i d** - унікальний ідентифікатор літака (типу INTEGER);
 - **boar d_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);
 - **wor ked_hour s** - кількість відпрацьованих годин (типу INTEGER);
 - **seat s** - кількість місць в літаку (типу INTEGER);
 - **capaci t y** - вантажопідйомність літака (типу INTEGER).
- **flights** - таблиця, що містить інформацію про рейси, з наступними полями:
 - **i d** - унікальний ідентифікатор рейсу (типу INTEGER);
 - **depar t ur e_poi nt** - пункт відправлення рейсу (типу VARCHAR) (NOT NULL, бо має існувати точка вильоту);
 - **dest i nat i on_poi nt** - пункт призначення рейсу (типу VARCHAR) (NOT NULL, бо має існувати точка посадки);
 - **depar t ur e_t i me** - час вильоту (типу TIMESTAMP) (NOT NULL, бо має існувати час вильоту);
 - **l andi ng_t i me** - час посадки (типу TIMESTAMP) (NOT NULL, бо має існувати час посадки);
 - **pl ane_i d** - ідентифікатор літака, на якому виконується рейс (типу INTEGER);
 - **sol d_t i cket s_number** - кількість проданих квитків на рейс (типу INTEGER).
- **crew_members** - таблиця, що містить інформацію про членів екіпажу, з наступними полями:
 - **i d** - унікальний ідентифікатор члена екіпажу (типу INTEGER);
 - **l ast _name** - прізвище члена екіпажу (типу VARCHAR) (NOT NULL, бо у пілота не може не бути прізвища);
 - **bi r t h_dat e** - дата народження члена екіпажу (типу DATE) (NOT NULL, бо оскільки пілот живий, він має мати дату народження);
 - **addr ess** - адреса члена екіпажу (типу VARCHAR).
- **pilots** - таблиця, що містить інформацію про пілотів, з наступними полями:
 - **i d** - унікальний ідентифікатор пілота (типу INTEGER);
 - **cr ew_member _i d** - 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);
 - **al l owed_pl anes** - список дозволених літаків для пілота (типу VARCHAR);
 - **l ast _f l i ght _dat e** - дата останнього вильоту пілота (типу TIMESTAMP).
- **flight_crew** - таблиця, що містить інформацію про екіпажі рейсів для реалізації зв'язку багато до багатьох, з наступними полями:
 - **f l i ght _i d** - ідентифікатор рейсу (типу INTEGER) (NOT NULL, бо потрібен ID рейсу для функціонування цієї таблиці);
 - **cr ew_member _i d** - ідентифікатор члена екіпажу, що входить до складу екіпажу рейсу (типу INTEGER) (NOT NULL, бо потрібен ID члена екіпажу для функціонування цієї таблиці);

Елементи даних	Об'єкт	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 кожену 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.