

Лабораторна робота №3: ОПЕРАТОР INSERT

Автор: Ясногородський Нікіта Вікторович, ТУ-12-22-Б1ІПЗ

Тема

Оператор **INSERT**, ключове слово **NULL** і обмеження цілісності.

Мета

Виконати різні форми оператора **INSERT**, додаючи записи в схему бази даних Постачальників з використанням різних обмежень цілісності: **AUTOINCREMENT**, **NOT NULL** і **DEFAULT XXX**.

Виконання

Створення таблиці

```
CREATE TABLE IF NOT EXISTS suppliers (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    name VARCHAR(255) NOT NULL,  
    address VARCHAR(255) DEFAULT 'Not provided'  
);
```

Оператор **INSERT** з обмеженням цілісності **AUTOINCREMENT**

```
INSERT INTO suppliers (name, address) VALUES ('Company A', 'Address 1');  
INSERT INTO suppliers (name, address) VALUES ('Company B', 'Address 2');
```




Оператор **INSERT** з обмеженням цілісності **NOT NULL**

```
INSERT INTO suppliers (name, address) VALUES (NULL, 'Address 3');
```

```
main> INSERT INTO suppliers (name, address) VALUES (NULL, 'Address 3')  
[2024-03-30 18:51:01] [19] [SQLITE_CONSTRAINT_NOTNULL] A NOT NULL constraint failed (NOT NULL constraint failed: suppliers.name)
```

Оператор **INSERT** з обмеженням цілісності **DEFAULT XXX**

```
INSERT INTO suppliers (name) VALUES ('Company D');  
SELECT * FROM suppliers WHERE name == 'Company D';
```

	 id	 name	 address
1	4	Company D	Not provided

Висновок

У цій лабораторній роботі ми використовували різні форми оператора **INSERT** для додавання даних до бази даних Постачальників, застосовуючи при цьому різні обмеження цілісності, такі як **AUTOINCREMENT**, **NOT NULL** і **DEFAULT XXX**. Ці обмеження сприяють забезпеченню правильності та цілісності даних в базі даних, дозволяючи зберігати інформацію у відповідності з вимогами та очікуваннями.