DDL

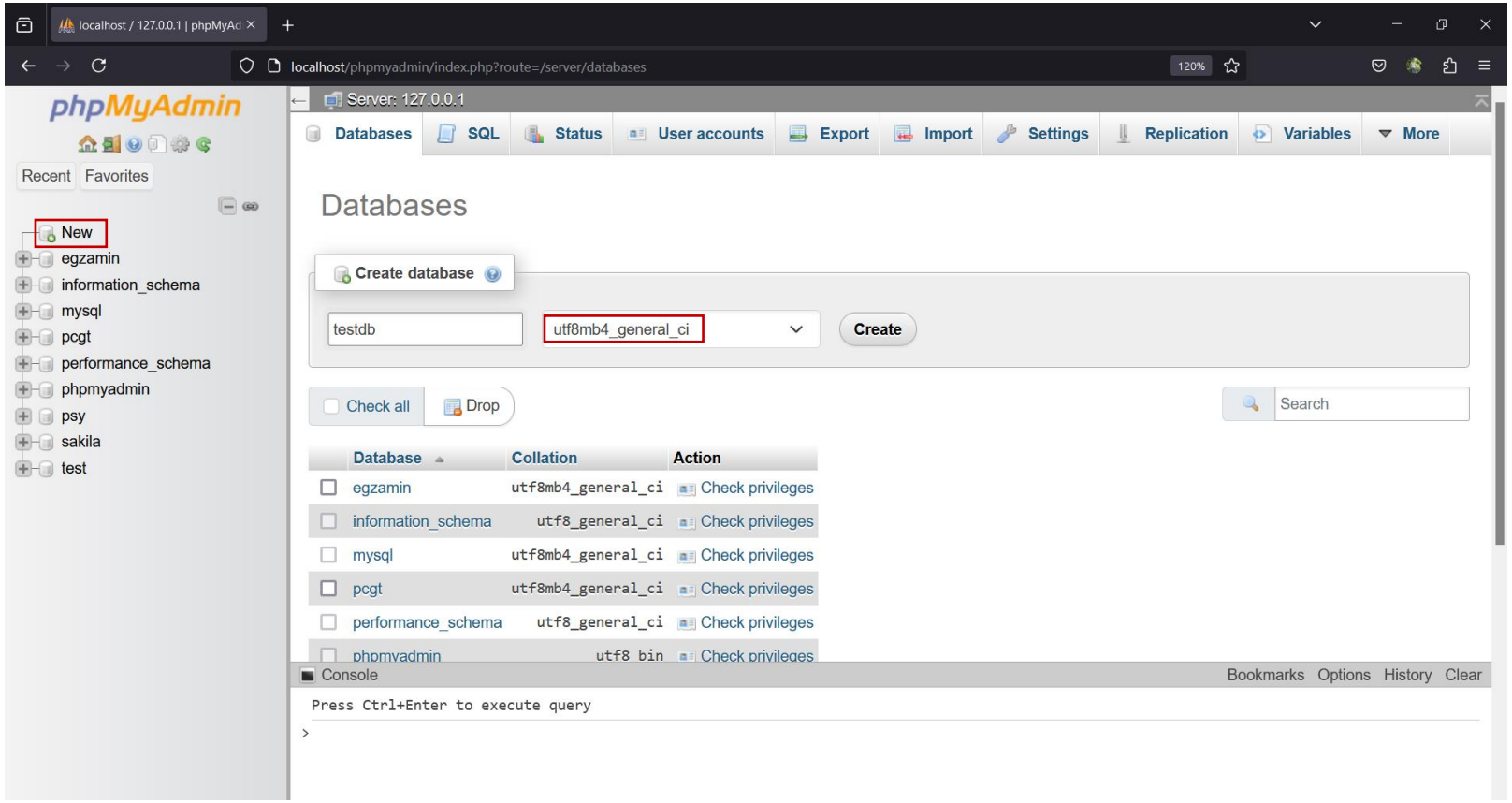
(Data Definition Language)

dimon.work/kurs.html

DDL

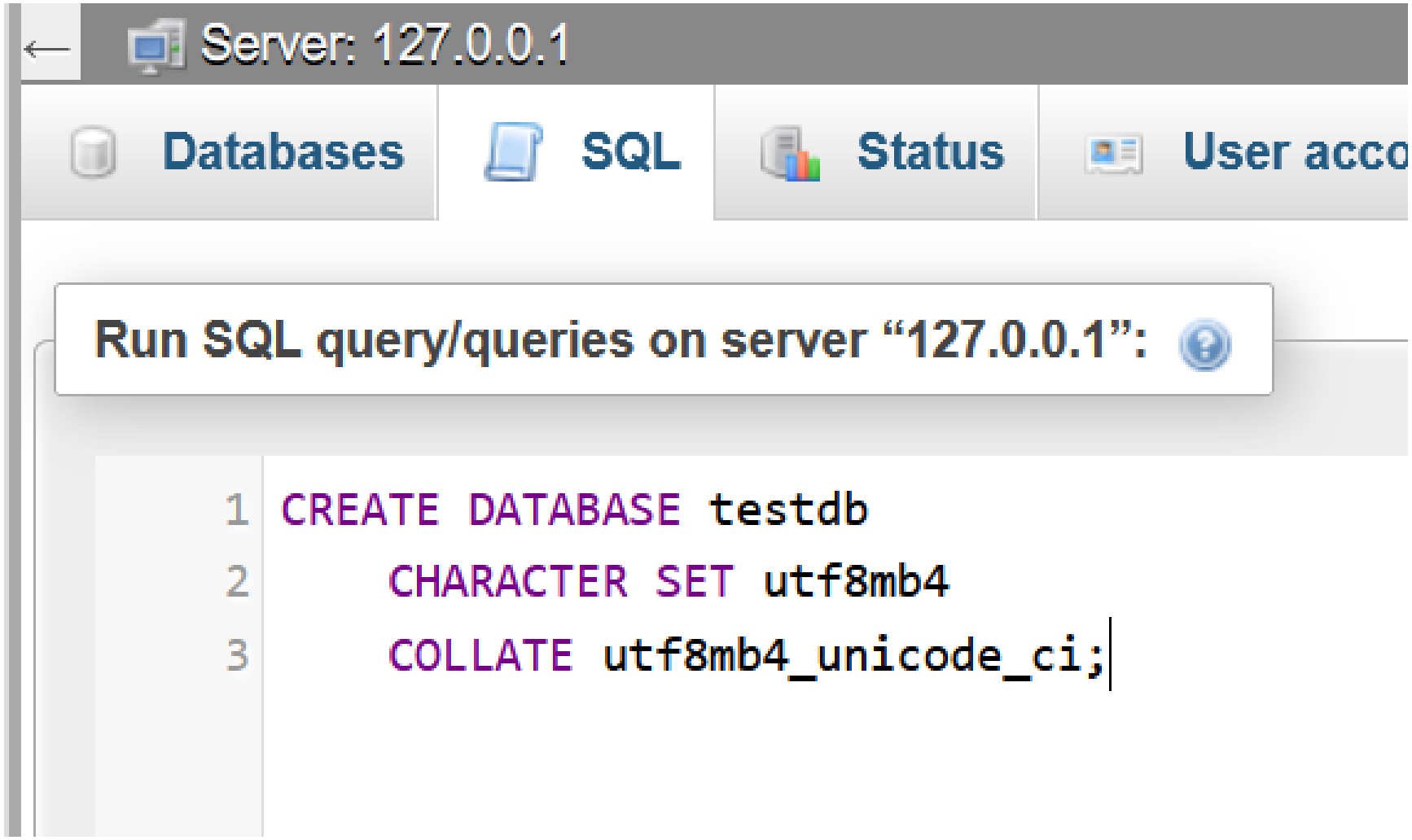
- **Data Definition Language**, **DDL** – grupa instrukcji w języku SQL, które służą do definiowania struktur danych. Możemy do nich zaliczyć polecenia takie jak CREATE, ALTER, DROP. Za pomocą instrukcji DDL nie manipulujemy bezpośrednio danymi, a ich strukturą. Można zdefiniować kolumny tabel, zmienić typy danych, czy usunąć obiekt taki jak widok czy tabela.

Jak stworzyć BD

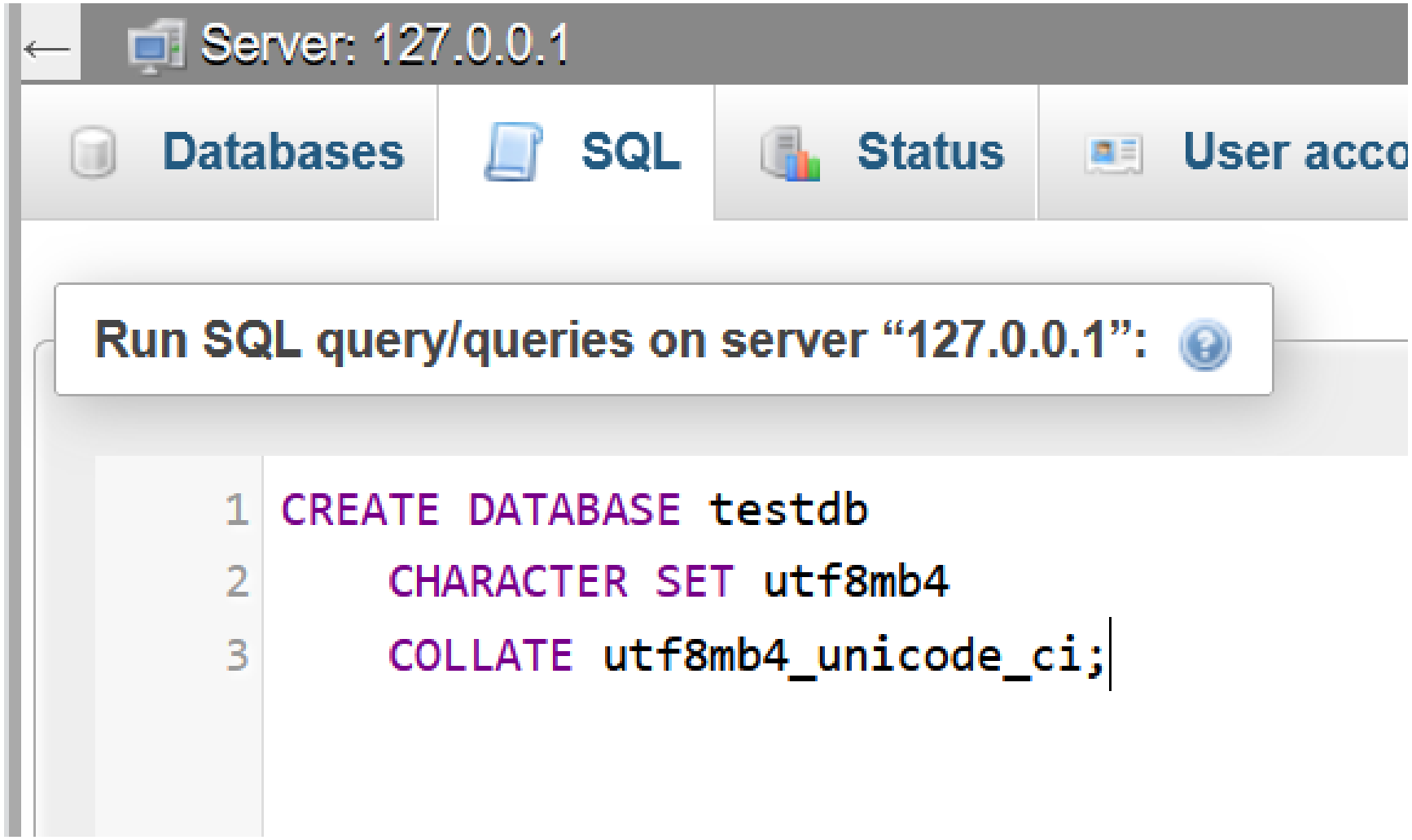


- **utf8mb4_unicode_ci** jest oparty na standardzie Unicode do sortowania i porównywania ciągów, który sortuje ciągi dokładniej w szerokim zakresie języków/alfabetach.
- **utf8mb4_general_ci** nie implementuje wszystkich reguł sortowania Unicode, co często prowadzi do niepożądanych wyników w niektórych sytuacjach dla niektórych języków/znaków.

Jak stworzyć BD

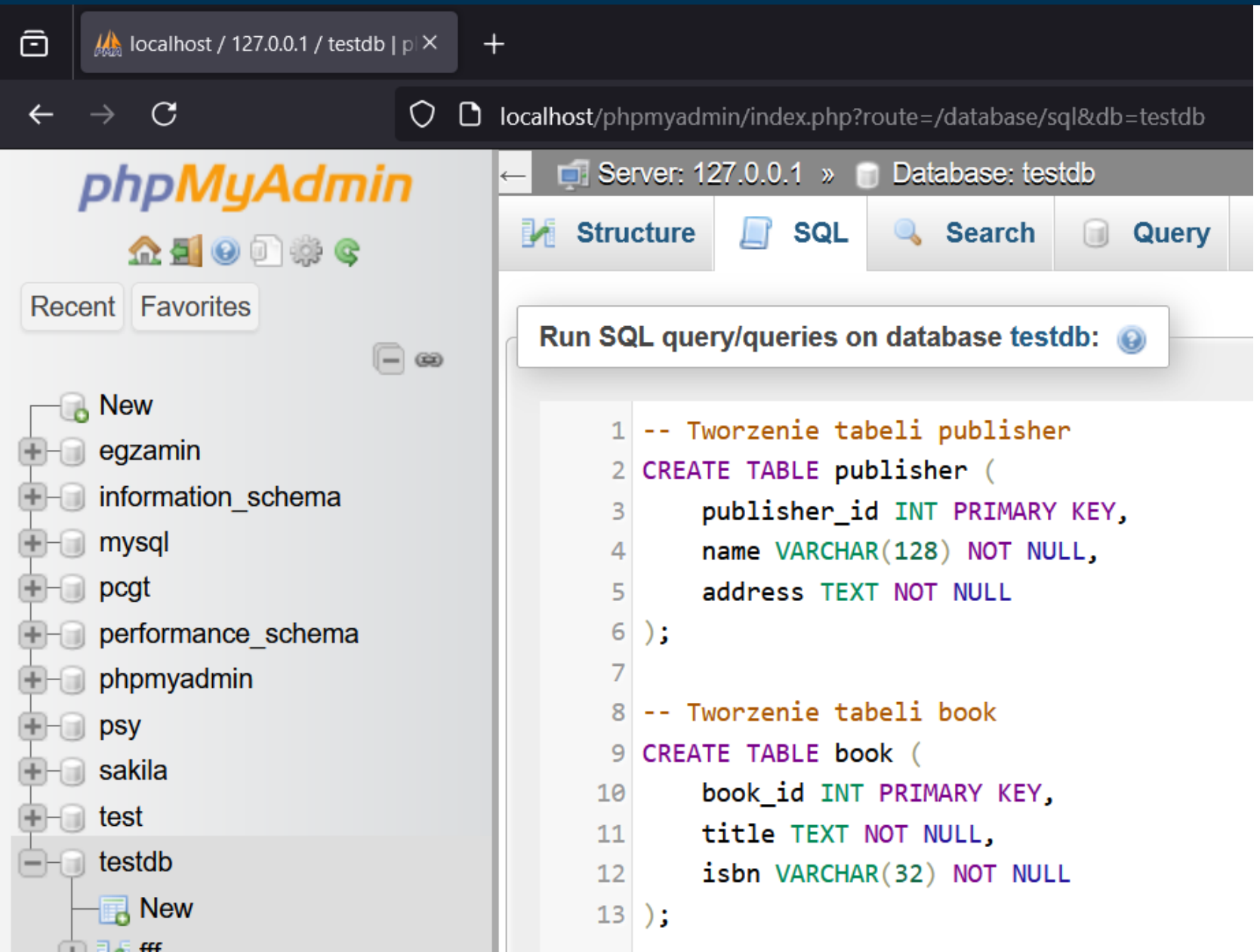


Jak stworzyć BD



Jak dodać tabelę w BD

dimon.work



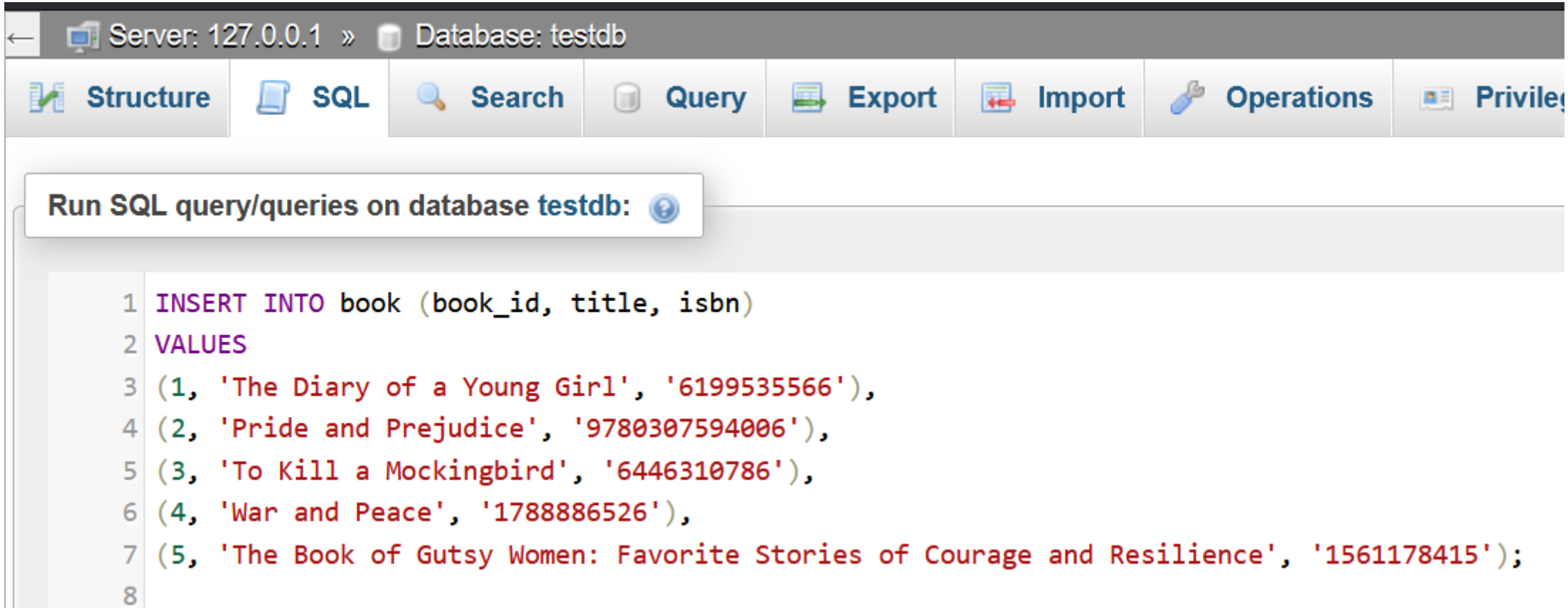
Dla tabeli publisher :

- **publisher_id INT PRIMARY KEY** - jest to klucz główny, który jednoznacznie identyfikuje każdego wydawcę.
- **name VARCHAR(128) NOT NULL** - nazwa wydawcy z limitem 128 znaków.
- **address TEXT NOT NULL** - pole tekstowe do przechowywania adresu.

Dla tabeli book:

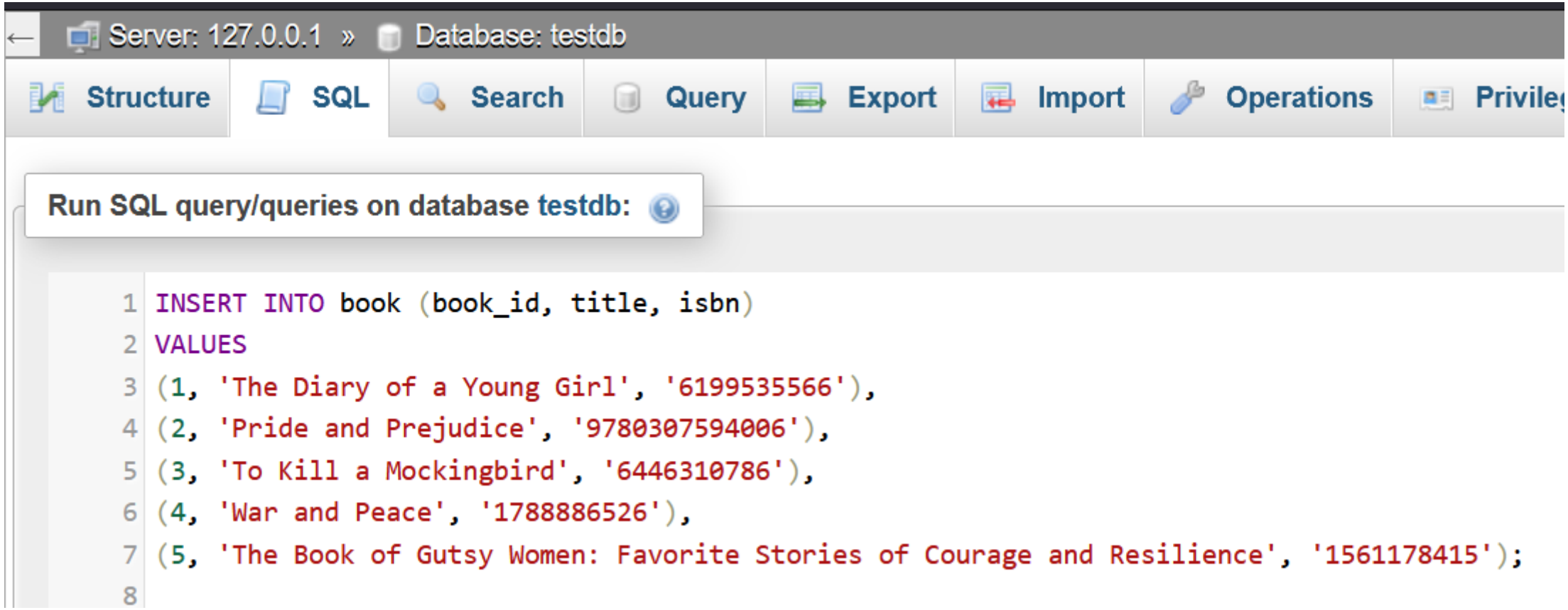
- **book_id INT PRIMARY KEY** - klucz główny dla tabeli book.
- **title TEXT NOT NULL** - tytuł książki.
- **isbn VARCHAR(32) NOT NULL** - pole na numer ISBN z limitem 32 znaków.

Wypełnienie tabel danymi



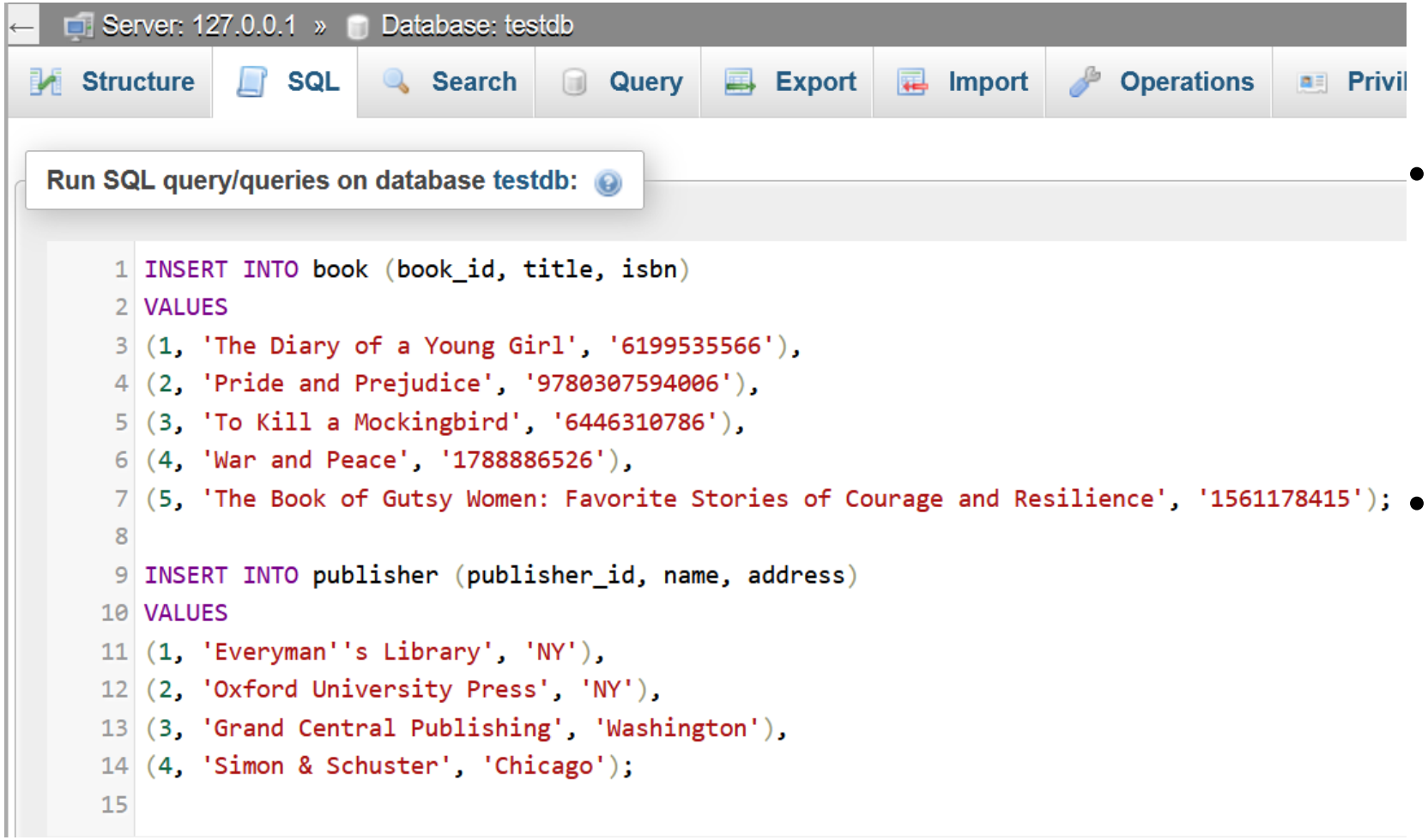
To polecenie SQL wstawia pięć rekordów (książek) do tabeli book, określając identyfikator książki (book_id), tytuł (title) i ISBN (isbn). Każdy rekord zawiera unikalny identyfikator i powiązane informacje o książce.

Wypełnienie tabel danymi



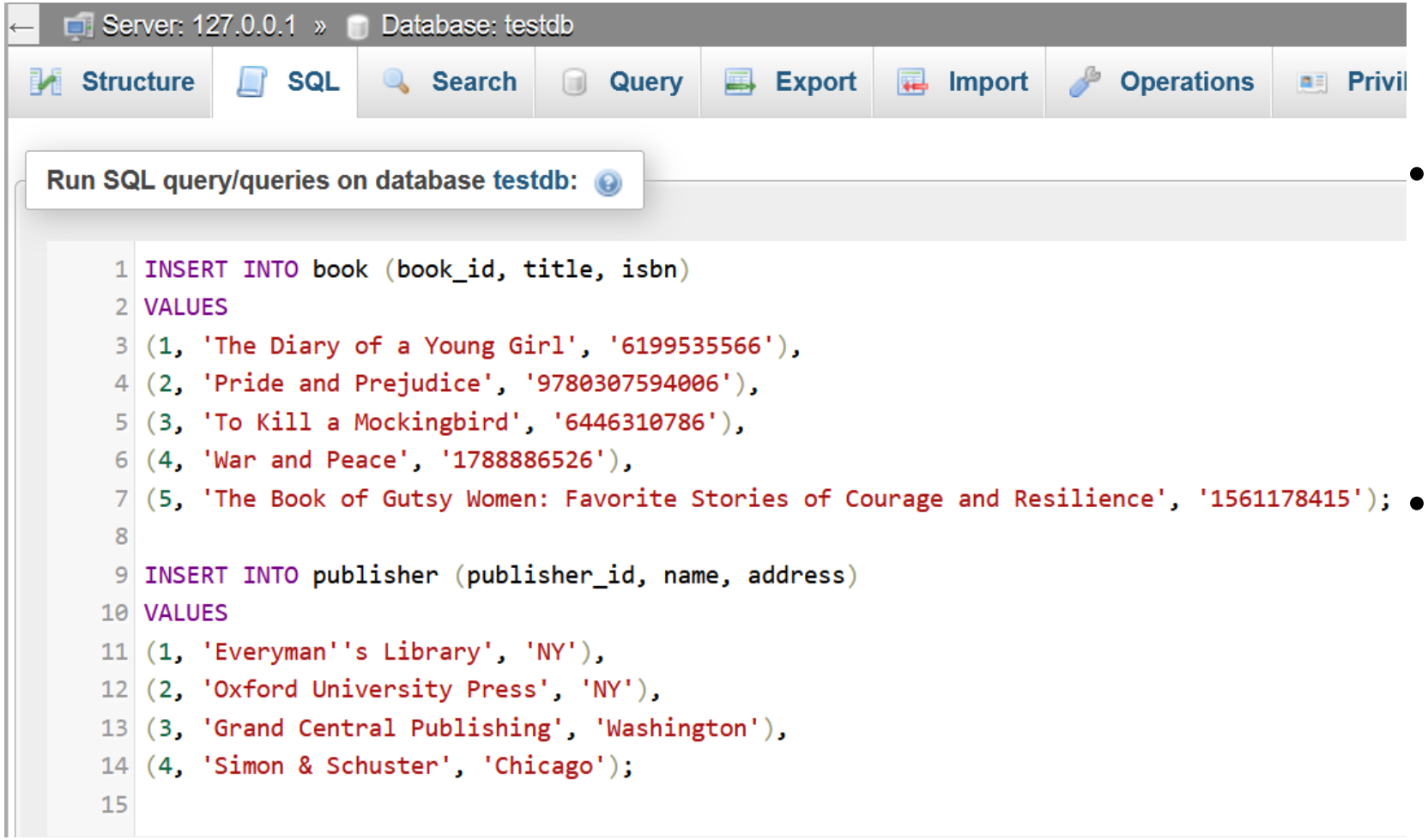
To polecenie SQL wstawia pięć rekordów (książek) do tabeli book, określając identyfikator książki (book_id), tytuł (title) i ISBN (isbn). Każdy rekord zawiera unikalny identyfikator i powiązane informacje o książce.

Wypełnienie tabel danymi



- Pojedynczy apostrof w łańcuchach SQL musi być uniknięty podwójnym apostrofem: 'Everyman''s Library’.
- Każdy rekord jest oddzielony przecinkiem, a średnik (;) jest używany tylko na końcu całego zapytania.

Wypełnienie tabel danymi



- Pojedynczy apostrof w łańcuchach SQL musi być uniknięty podwójnym apostrofem: 'Everyman''s Library’.
- Każdy rekord jest oddzielony przecinkiem, a średnik (;) jest używany tylko na końcu całego zapytania.

Relacja 1 do wielu

Publisher

id	name	address
1	Everyman"s Library	NY
2	Oxford University Press	NY
3	Grand Central Publishing	Washington
4	Simon & Schuster	Chicago

Book

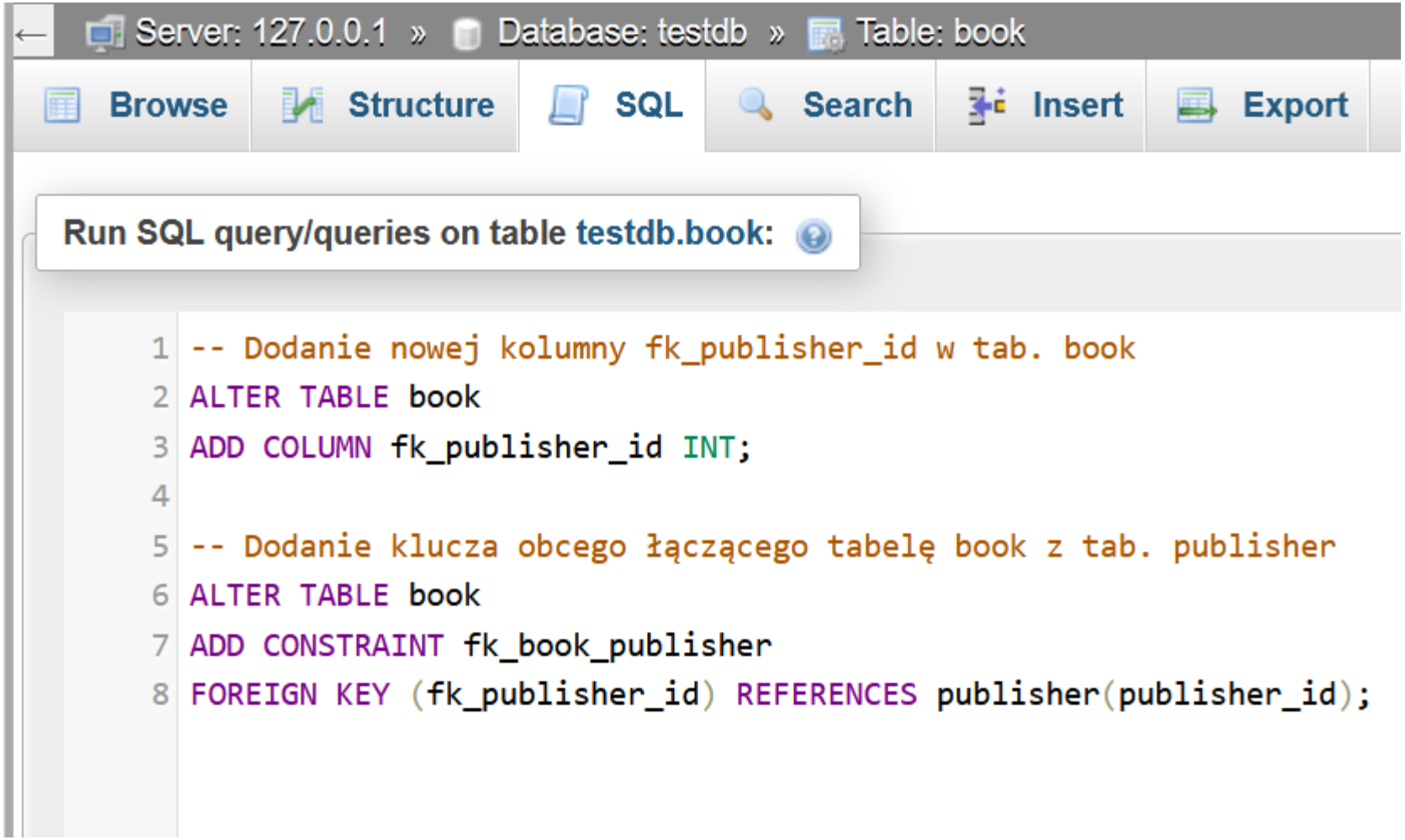
id	title	ISBN
1	The Diary of a Young Girl	199535566
2	Pride and Prejudice	9780307594006
3	To Kill a Mockingbird	0446310786
4	The Book of Gutsy Women	1501178415
5	War and Peace	1788886526

1 publisher can publish many books

id	name	address	title	ISBN
2	Oxford University Press	NY	To Kill a Mockingbird	0446310786
2	Oxford University Press	NY	The Book of Gutsy Women	1501178415
2	Oxford University Press	NY	War and Peace	1788886526

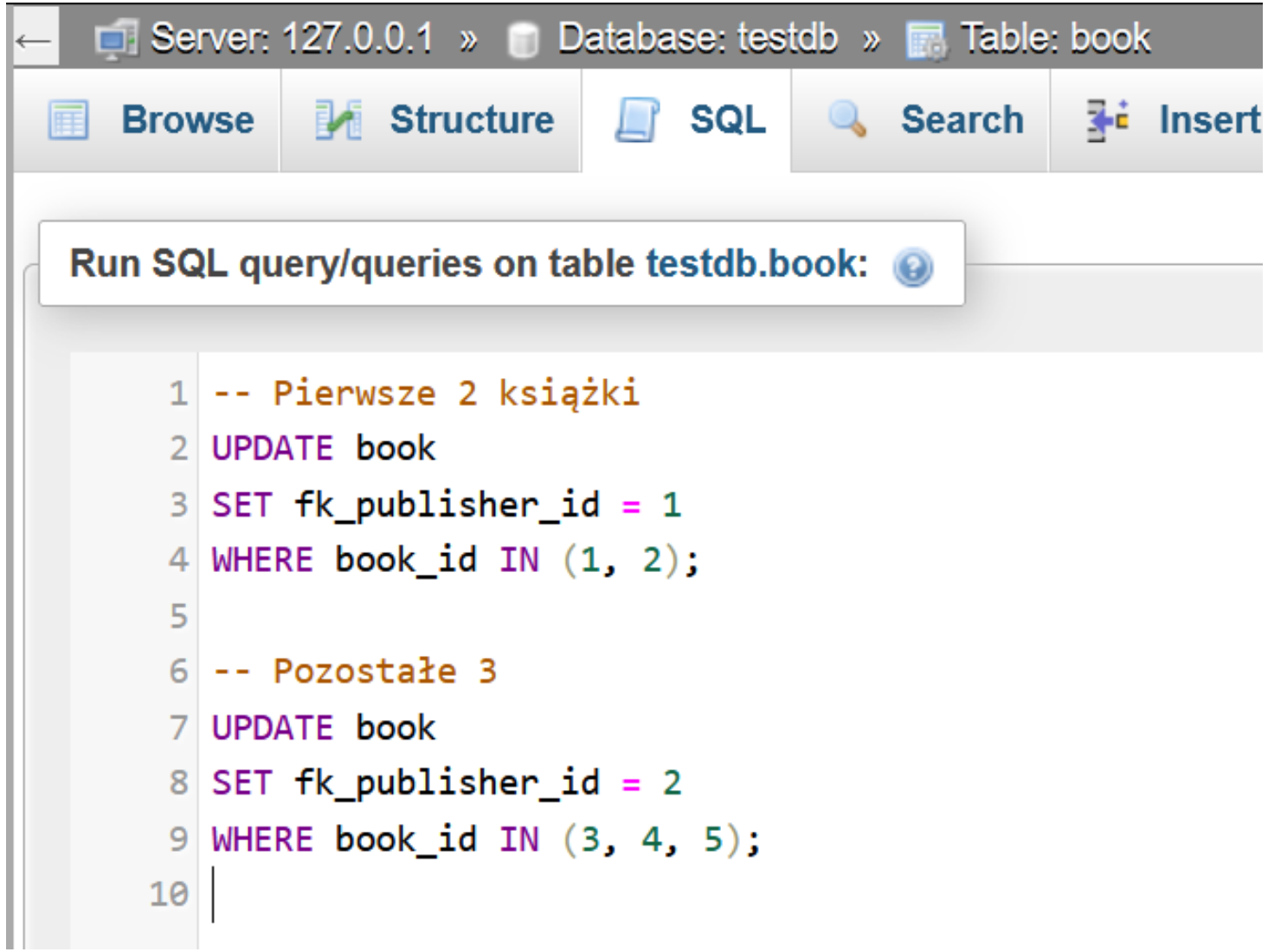
Dla połączenia tabel wykorzystamy **klucz obcy**. Dodamy w tabelę book dodatkową kołumnę z indeksem wydania

Books & Publishers



- **ADD COLUMN fk_publisher_id INT**; - do tabeli book dodawana jest nowa kolumna fk_publisher_id typu INT, która posłuży jako klucz obcy do połączenia z tabelą publisher.
- **ADD CONSTRAINT fk_book_publisher** - dodawane jest ograniczenie klucza obcego, które łączy kolumnę fk_publisher_id tabeli book z kolumną publisher_id tabeli publisher.

Books & Publishers



Books & Publishers

book_id	title	isbn	fk_publisher_id
1	The Diary of a Young Girl	6199535566	1
2	Pride and Prejudice	9780307594006	1
3	To Kill a Mockingbird	6446310786	2
4	War and Peace	1788886526	2
5	The Book of Gutsy Women: Favorite Stories of Coura...	1561178415	2

Books & Publishers

book_id	title	isbn	fk_publisher_id
1	The Diary of a Young Girl	6199535566	1
2	Pride and Prejudice	9780307594006	1
3	To Kill a Mockingbird	6446310786	2
4	War and Peace	1788886526	2
5	The Book of Gutsy Women: Favorite Stories of Coura...	1561178415	2

Relacja 1 do 1

Person

id	first_name	last_name
1	John	Snow
2	Ned	Stark
3	Rob	Baratheon

Passport

serial_number	registration	fk_person_id
123456	Winterfell	1
789012	Winterfell	2
345678	King's Landing	3

Полная выборка из объединённых таблиц

id	first_name	last_name	serial_number	registration
1	John	Snow	123456	Winterfell
2	Ned	Stark	789012	Winterfell
3	Rob	Baratheon	345678	King's Landing

Person & Passport

dimon.work

Server: 127.0.0.1 » Database: testdb

Structure | SQL | Search | Query | Export | Import

Run SQL query/queries on database testdb:

```sql
-- Tworzenie tabeli person
CREATE TABLE person (
    person_id INT PRIMARY KEY,
    first_name VARCHAR(64) NOT NULL,
    last_name VARCHAR(64) NOT NULL
);

-- Tworzenie tabeli passport
CREATE TABLE passport (
    passport_id INT PRIMARY KEY,
    serial_number INT NOT NULL,
    fk_person_id INT,
    FOREIGN KEY (fk_person_id) REFERENCES person(person_id)
);
```

Server: 127.0.0.1 » Database: testdb

Structure | SQL | Search | Query | Export

Run SQL query/queries on database testdb:

```sql
INSERT INTO person VALUES (1, 'John', 'Snow');
INSERT INTO person VALUES (2, 'Ned', 'Stark');
INSERT INTO person VALUES (3, 'Rob', 'Baratheon');
```

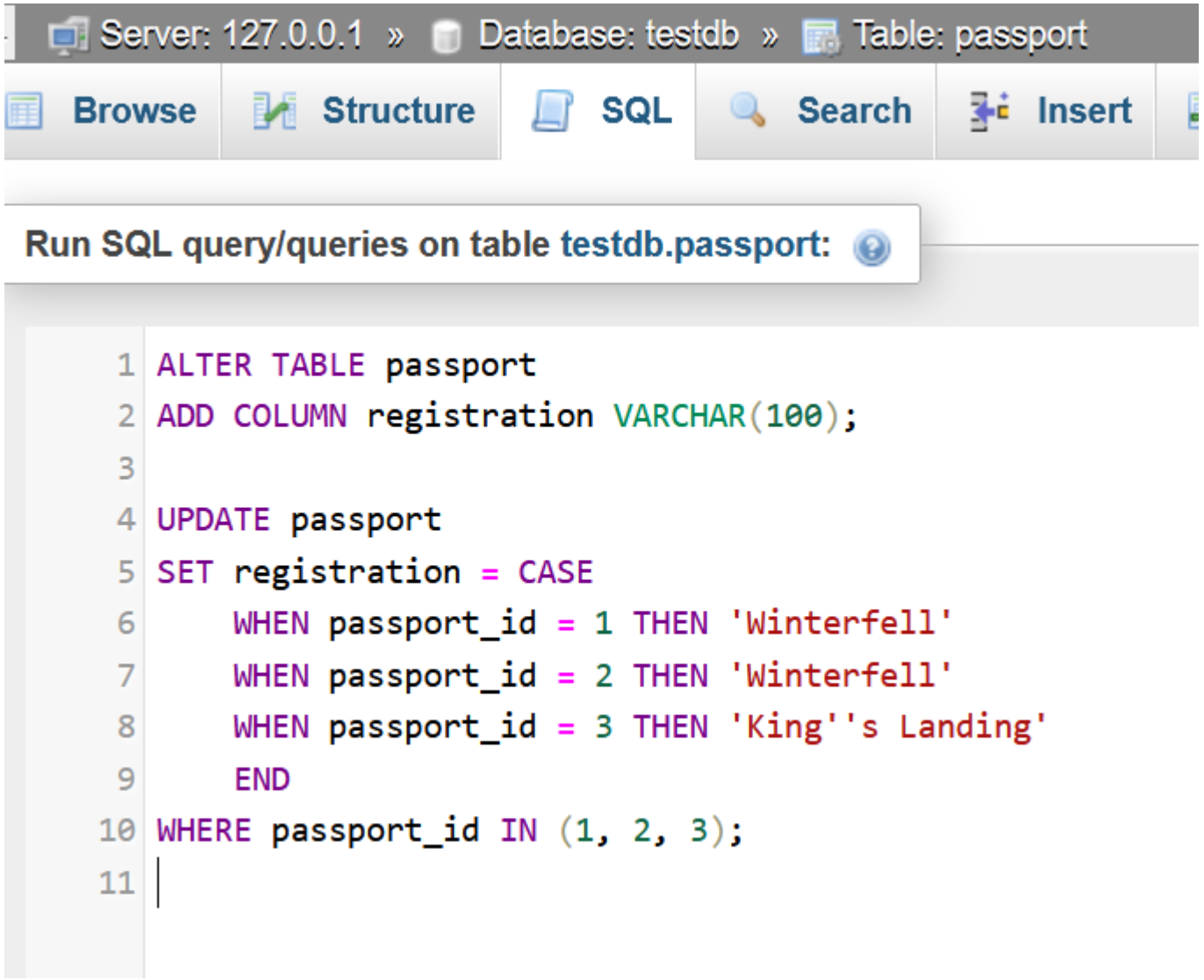
Server: 127.0.0.1 » Database: testdb

Structure | SQL | Search | Query

Run SQL query/queries on database testdb:

```sql
INSERT INTO passport VALUES (1, 123456, 1);
INSERT INTO passport VALUES (2, 789012, 2);
INSERT INTO passport VALUES (3, 345678, 3);
```

Person & Passport



- W tym zapytaniu **SET registration = CASE** pozwala określić różne wartości dla rejestracji w zależności od passport_id.
- Warunek **WHERE passport_id IN (1, 2, 3)** gwarantuje, że aktualizacja zostanie zastosowana tylko do określonych rekordów.

Relacja wiele do wielu

Book

book_id	title	isbn
1	Book for Dummies	123456
2	Book for Smart Guys	7890123
3	Book for Happy People	4567890
4	Book for Unhappy People	1234567

Author

author_id	full_name	rating
1	Bob	4,50
2	Alice	4,00
3	John	4,70

book_author

book_id	author_id
1	1
2	1
3	1
3	2
4	1
4	2
4	3