

## Testovací scénář 2: Testování funkcionalit

**Projekt:** Library DB System (D1)

**Verze:** 1.0

**Datum:** 2025-01-11

**Předpoklady:** Aplikace je nainstalována a běží (viz Scénář 1)

---

### Test 1: Vložení do více tabulek najednou (BOD 4)

#### Krok 1.1: Přidání knihy s kategoriemi

1. Otevřete `http://localhost:5000`
2. Klikněte na “Add Book with Categories”
3. Vyplňte formulář:
  - Title: `Foundation`
  - Author: `Isaac Asimov`
  - Price: `299.90`
  - Published Date: `1951-05-01`
  - Categories: Zaškrtněte `Science Fiction`
4. Klikněte “Add Book”

**Očekávaný výsledek:** - Zpráva: “Book with categories added successfully!”  
- Kniha se zobrazí v seznamu “Books” - V databázi jsou data v tabulkách: `book`, `author` (pokud neexistoval), `book_category`

#### Krok 1.2: Ověření v databázi

Spustíte SQL dotaz:

```
SELECT b.title, a.name AS author, c.name AS category
FROM book b
JOIN author a ON b.author_id = a.id
JOIN book_category bc ON b.id = bc.book_id
JOIN category c ON bc.category_id = c.id
WHERE b.title = 'Foundation';
```

**Očekávaný výsledek:** - Vrátí řádek s knihou `Foundation`, autorem `Isaac Asimov` a kategorií `Science Fiction`

---

### Test 2: Zobrazení a úprava dat

#### Krok 2.1: Zobrazení knihy

1. Klikněte na “View Books”
2. Najděte knihu “Foundation”

**Očekávaný výsledek:** - Kniha se zobrazuje s těmito údaji: - Title: Foundation - Author: Isaac Asimov - Price: 299.90 - Available: Yes - Categories: Science Fiction

### **Krok 2.2: Úprava knihy**

1. Klikněte na “Edit” u knihy “Foundation”
2. Změňte:
  - Price: 350.00
  - Available: No
3. Klikněte “Update Book”

**Očekávaný výsledek:** - Zpráva: “Book updated successfully!” - Cena je změněna na 350.00 - Available je “No”

---

## **Test 3: Smazání dat**

### **Krok 3.1: Smazání autora (test cizích klíčů)**

1. Klikněte na “View Authors”
2. Zkuste smazat autora “Isaac Asimov” (který má knihy)
3. Klikněte “Delete”

**Očekávaný výsledek:** - Chybová hláška: “Database error: ...” (cizí klíč brání smazání) - Autor NENÍ smazán

### **Krok 3.2: Smazání knihy**

1. Klikněte na “View Books”
2. Najděte knihu “Foundation”
3. Klikněte “Delete” a potvrďte

**Očekávaný výsledek:** - Zpráva: “Book deleted successfully!” - Kniha zmizela ze seznamu - V databázi je smazána i vazba v `book_category`

---

## **Test 4: Transakce - Půjčení knihy (BOD 6)**

### **Příprava: Přidejte novou knihu**

1. Přidejte knihu:
  - Title: I, Robot
  - Author: Isaac Asimov
  - Price: 250.00
  - Published Date: 1950-12-02
  - Categories: Science Fiction

#### Krok 4.1: Půjčení knihy

1. Klikněte na “Loan Book”
2. Vyplňte:
  - Book ID: (ID knihy “I, Robot”, např. 2)
  - User ID: (ID uživatele “John Doe”, např. 1)
3. Klikněte “Loan Book”

**Očekávaný výsledek:** - Zpráva: “Book loaned successfully!” - V seznamu “Loans” se zobrazí nová půjčka - V seznamu “Books” je kniha označena jako Available: No

#### Krok 4.2: Ověření v databázi

```
SELECT * FROM loan WHERE book_id = 2;  
SELECT available FROM book WHERE id = 2;
```

**Očekávaný výsledek:** - V loan je záznam s returned = 0 - V book je available = 0

#### Krok 4.3: Vracení knihy

1. Klikněte na “View Loans”
2. Najděte půjčku knihy “I, Robot”
3. Klikněte “Return”

**Očekávaný výsledek:** - Zpráva: “Book returned successfully!” - V seznamu “Loans” je Returned: Yes - V seznamu “Books” je kniha opět Available: Yes

---

### Test 5: Report ze 3+ tabulek (BOD 7)

#### Krok 5.1: Zobrazení reportu

1. Klikněte na “Books by Author & Category”

**Očekávaný výsledek:** - Zobrazí se tabulka s daty ze 3 tabulek (book, author, category) - Obsahuje agregované hodnoty: - Total Books (COUNT) - Avg Price (AVG) - Min Price (MIN) - Max Price (MAX) - Available Books (SUM) - Data jsou seskupena podle autora a typu kategorie

#### Krok 5.2: Ověření SQL dotazu

```
SELECT  
    a.name AS author_name,  
    c.category_type,  
    COUNT(DISTINCT b.id) AS total_books,  
    AVG(b.price) AS avg_price,  
    MIN(b.price) AS min_price,
```

```

MAX(b.price) AS max_price,
SUM(CASE WHEN b.available = 1 THEN 1 ELSE 0 END) AS available_books
FROM book b
JOIN author a ON b.author_id = a.id
LEFT JOIN book_category bc ON b.id = bc.book_id
LEFT JOIN category c ON bc.category_id = c.id
GROUP BY a.name, c.category_type
ORDER BY total_books DESC;

```

**Očekávaný výsledek:** - Dotaz vrací stejná data jako report v aplikaci

---

## Test 6: Import JSON (BOD 8)

**Příprava:** Vytvořte JSON soubor

Vytvořte soubor `test_import.json`:

```

[
  {
    "title": "Dune",
    "author": "Frank Herbert",
    "price": 399.99,
    "published_date": "1965-08-01",
    "categories": ["fiction", "study"]
  },
  {
    "title": "Neuromancer",
    "author": "William Gibson",
    "price": 299.50,
    "published_date": "1984-07-01",
    "categories": ["fiction"]
  }
]

```

### Krok 6.1: Import JSON

1. Klikněte na “Import JSON”
2. Vyberte soubor `test_import.json`
3. Klikněte “Import”

**Očekávaný výsledek:** - Zpráva: “JSON imported successfully!” - V seznamu “Books” se zobrazují nové knihy: Dune, Neuromancer - V seznamu “Authors” jsou noví autoři: Frank Herbert, William Gibson - Data jsou v tabulkách: `author`, `book`, `category`, `book_category`

## Krok 6.2: Ověření v databázi

```
SELECT b.title, a.name AS author, STRING_AGG(c.name, ', ') AS categories
FROM book b
JOIN author a ON b.author_id = a.id
LEFT JOIN book_category bc ON b.id = bc.book_id
LEFT JOIN category c ON bc.category_id = c.id
WHERE b.title IN ('Dune', 'Neuromancer')
GROUP BY b.title, a.name;
```

**Očekávaný výsledek:** - Dune má 2 kategorie - Neuromancer má 1 kategorii

---

## Test 7: Views (BOD 2)

### Krok 7.1: Test view vw\_books\_overview

```
SELECT * FROM vw_books_overview;
```

**Očekávaný výsledek:** - Vrací seznam knih s autory - Obsahuje sloupce: id, title, author, price, available, published\_date

### Krok 7.2: Test view vw\_loan\_report

1. V aplikaci klikněte na “Loan Statistics”

**Očekávaný výsledek:** - Zobrazí statistiky půjček pro každého uživatele - Obsahuje: Total Loans, Active Loans, Last Loan Date

---

## Test 8: Konfigurace (BOD 9)

### Krok 8.1: Špatné heslo

1. Zastavte aplikaci (Ctrl+C)
2. V config/config.json změňte password na špatné
3. Spusťte aplikaci

**Očekávaný výsledek:** - Aplikace zobrazí chybu: “FATAL ERROR: Database connection failed: ...” - Aplikace se neukončí s Python chybou, ale s user-friendly hláškou

### Krok 8.2: Chybějící config soubor

1. Přejmenujte config/config.json na config/config.json.bak
2. Spusťte aplikaci

**Očekávaný výsledek:** - Chyba: “FATAL ERROR: Failed to load config file: ...” - Aplikace se nezhroutí

### Krok 8.3: Oprava konfigurace

1. Vratte `config/config.json`
2. Opravte heslo
3. Spustte aplikaci

**Očekávaný výsledek:** - Aplikace běží normálně

---

## Test 9: Validace vstupů

### Krok 9.1: Prázdné pole

1. Klikněte na “Add User”
2. Nechte pole prázdná
3. Klikněte “Add User”

**Očekávaný výsledek:** - HTML validace: “Please fill out this field” - Data NEJSOU vložena

### Krok 9.2: Neplatný email

1. Klikněte na “Add User”
2. Vyplňte:
  - Full Name: Test User
  - Email: invalid-email
3. Klikněte “Add User”

**Očekávaný výsledek:** - Chybová hláška: “Invalid email format” - Uživatel NENÍ vytvořen

### Krok 9.3: Neplatná cena

1. Klikněte na “Add Book with Categories”
2. Vyplňte vše, ale Price: abc
3. Klikněte “Add Book”

**Očekávaný výsledek:** - Chybová hláška: “Price must be a valid number” - Kniha NENÍ vytvořena

---

## Závěr testování

### Shrnutí testů:

Test	Funkce	Výsledek
1	Vložení do více tabulek	PASS / FAIL
2	Zobrazení a úprava dat	PASS / FAIL

Test	Funkce	Výsledek
3	Smazání dat	PASS / FAIL
4	Transakce (půjčení/vrácení)	PASS / FAIL
5	Report ze 3+ tabulek	PASS / FAIL
6	Import JSON	PASS / FAIL
7	Views	PASS / FAIL
8	Konfigurace	PASS / FAIL
9	Validace vstupů	PASS / FAIL

**Tester:** \_\_\_\_\_

**Datum:** \_\_\_\_\_

**Celkový výsledek:** PASS / FAIL

**Poznámky:**