Python Solutions for Given Tasks

SQLite Database Operations

```
import sqlite3

def create_and_insert():
    try:
        conn = sqlite3.connect("students.db")
        cursor = conn.cursor()
        cursor.execute("CREATE TABLE IF NOT EXISTS students (id INTEGER PRIMARY KEY,
name TEXT, grade TEXT)")

        data = [(1, "Alice", "A"), (2, "Bob", "B"), (3, "Charlie", "A"), (4, "David",
"C"), (5, "Eve", "B")]
        cursor.executemany("INSERT INTO students VALUES (?, ?, ?)", data)
        conn.commit()

        cursor.execute("SELECT * FROM students")
        rows = cursor.fetchall()
        for row in rows:
            print(row)

    except sqlite3.Error as e:
        print("SQLite error:", e)
    finally:
        conn.close()

create_and_insert()
```

Database and Exception Handling

```
def safe_create_and_insert():
    try:
        conn = sqlite3.connect("students.db")
        cursor = conn.cursor()
        cursor.execute("CREATE TABLE IF NOT EXISTS students (id INTEGER PRIMARY KEY,
name TEXT, grade TEXT)")

        data = [(6, "Frank", "A"), (7, "Grace", "B")]
        cursor.executemany("INSERT INTO students VALUES (?, ?, ?)", data)
        conn.commit()

        cursor.execute("SELECT * FROM students")
        rows = cursor.fetchall()
        for row in rows:
            print(row)

    except sqlite3.OperationalError as e:
        print("Operational Error:", e)
    except sqlite3.IntegrityError as e:
        print("Integrity Error:", e)
    except sqlite3.Error as e:
        print("SQLite Error:", e)
    finally:
        if conn:
            conn.close()

safe_create_and_insert()
```

Modules for Data Manipulation (data_handler.py)

```
def read_data(file_path):
    try:
        with open(file_path, "r") as file:
            return file.read()
    except FileNotFoundError:
        return "File not found."

def write_data(data, file_path):
    try:
        with open(file_path, "w") as file:
            file.write(data)
        return "Data written successfully."
    except Exception as e:
        return f"Error writing file: {e}"
```

Package Creation

```
from my_package.data_handler import read_data, write_data

file_content = read_data("input.txt")
write_data(file_content, "output.txt")
```

Advanced Lambda and Exception Handling

```
inverse = lambda x: 1 / x if isinstance(x, (int, float)) and x != 0 else "Invalid input"

test_values = [2, 5, 0, "hello", 3.5]
results = [inverse(x) for x in test_values]
```

```
print(results)
```

Created by Sayak