Task Management Assistant – Complete Python Code

```
Task Management Assistant
Python 3.8+

Features:
- Natural language task creation
- Priority detection
- Due date parsing
- Category assignment
- Progress tracking
- SQLite database
- CLI interface

--------------------------------------------------

import sqlite3
import re
from dateutil.parser import parse

DB_NAME = "tasks.db"

def init_db():
    conn = sqlite3.connect(DB_NAME)
    cursor = conn.cursor()
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS tasks (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            title TEXT,
            priority TEXT,
            category TEXT,
            due_date TEXT,
            status TEXT
        )
    """)
    conn.commit()
    conn.close()

def add_task(title, priority, category, due_date):
    conn = sqlite3.connect(DB_NAME)
    cursor = conn.cursor()
    cursor.execute(
        "INSERT INTO tasks (title, priority, category, due_date, status) VALUES (?, ?, ?, ?, ?)",
        (title, priority, category, due_date, "Pending")
    )
    conn.commit()
    conn.close()

def get_tasks(status=None):
    conn = sqlite3.connect(DB_NAME)
    cursor = conn.cursor()
    if status:
        cursor.execute("SELECT * FROM tasks WHERE status=?", (status,))
    else:
        cursor.execute("SELECT * FROM tasks")
    tasks = cursor.fetchall()
    conn.close()
    return tasks

def update_task_status(task_id, status):
    conn = sqlite3.connect(DB_NAME)
    cursor = conn.cursor()
    cursor.execute("UPDATE tasks SET status=? WHERE id=?", (status, task_id))
    conn.commit()
    conn.close()

def detect_priority(text):
    if "urgent" in text or "high" in text:
        return "High"
    elif "low" in text:
        return "Low"
    return "Medium"
```

```
def detect_category(text):
    if "assignment" in text or "study" in text:
        return "Study"
    elif "office" in text or "work" in text:
        return "Work"
    elif "health" in text:
        return "Health"
    return "Personal"

def extract_due_date(text):
    try:
        return str(parse(text, fuzzy=True).date())
    except:
        return "Not specified"

def extract_title(text):
    text = re.sub(r"(add|create|remind me to)", "", text, flags=re.I)
    return text.strip().capitalize()

def main():
    init_db()
    print("Task Manager Assistant")
    while True:
        user_input = input("You: ").lower()
        if user_input == "exit":
            break
        elif "show" in user_input:
            tasks = get_tasks()
            for t in tasks:
                print(t)
        elif "complete" in user_input:
            task_id = int(input("Enter task ID: "))
            update_task_status(task_id, "Completed")
        else:
            add_task(
                extract_title(user_input),
                detect_priority(user_input),
                detect_category(user_input),
                extract_due_date(user_input)
            )

if __name__ == "__main__":
    main()
```