import sqlite3

# Connect to database and create tables

def connect\_db():

conn = sqlite3.connect("database.db")

cursor = conn.cursor()

# Students table

cursor.execute('''CREATE TABLE IF NOT EXISTS students (

id INTEGER PRIMARY KEY AUTOINCREMENT,

name TEXT NOT NULL,

age INTEGER NOT NULL,

room\_no INTEGER)''')

# Rooms table

cursor.execute('''CREATE TABLE IF NOT EXISTS rooms (

id INTEGER PRIMARY KEY AUTOINCREMENT,

room\_no INTEGER UNIQUE NOT NULL,

capacity INTEGER NOT NULL,

occupied INTEGER DEFAULT 0)''')

# Payments table

cursor.execute('''CREATE TABLE IF NOT EXISTS payments (

id INTEGER PRIMARY KEY AUTOINCREMENT,

student\_id INTEGER,

amount REAL,

date TEXT,

FOREIGN KEY(student\_id) REFERENCES students(id))''')

conn.commit()

conn.close()

# Add Room

def add\_room(room\_no, capacity):

conn = sqlite3.connect("database.db")

cursor = conn.cursor()

cursor.execute("INSERT INTO rooms (room\_no, capacity) VALUES (?, ?)", (room\_no, capacity))

conn.commit()

conn.close()

# Fetch Rooms

def fetch\_rooms():

conn = sqlite3.connect("database.db")

cursor = conn.cursor()

cursor.execute("SELECT \* FROM rooms")

rooms = cursor.fetchall()

conn.close()

return rooms

# Add Payment

def add\_payment(student\_id, amount, date):

conn = sqlite3.connect("database.db")

cursor = conn.cursor()

cursor.execute("INSERT INTO payments (student\_id, amount, date) VALUES (?, ?, ?)", (student\_id, amount, date))

conn.commit()

conn.close()

# Fetch Payments

def fetch\_payments():

conn = sqlite3.connect("database.db")

cursor = conn.cursor()

cursor.execute("SELECT payments.id, students.name, payments.amount, payments.date FROM payments JOIN students ON payments.student\_id = students.id")

payments = cursor.fetchall()

conn.close()

return payments