```
#!/usr/bin/env python3
import psycopg2

def openConnection():
    try:
        # Connect to the database
        conn = psycopg2.connect(
            host="awsprddbs4836.shared.sydney.edu.au",
            database="y24s2c9120_xliu0943",
            user="y24s2c9120_xliu0943",
            password="whBcbt6y"
        )
        print("Database connection successful.")
        return conn
    except psycopg2.Error as sqle:
        print("psycopg2.Error : " + sqle.pgerror)
        return None


def checkLogin(login, password):
    conn = openConnection()
    cur = conn.cursor()
    try:
        # Validate username and password
        query = """
            SELECT UserName, FirstName, LastName, Email
            FROM Administrator
            WHERE UserName = %s AND Password = %s
        """
        cur.execute(query, (login, password))
        user = cur.fetchall()
        return list(user[0]) if user else None
    except Exception as e:
        return None
    finally:
        cur.close()  # Ensure the cursor is closed
        conn.close()  # Ensure the connection is closed

def findAdmissionsByAdmin(login):
    conn = openConnection()
    cur = conn.cursor()
    try:
        # Query admissions for the administrator
        query = """
            SELECT A.AdmissionID AS admission_id,
                   T.AdmissionTypeName AS admission_type,
                   D.DeptName AS admission_department,
                   COALESCE(TO_CHAR(A.DischargeDate, 'DD-MM-YYYY'), '') AS
discharge_date,
                   COALESCE(A.Fee::text, '') AS fee,
                   A.Patient AS patient,
                   COALESCE(A.Condition, '') AS condition
            FROM Admission AS A
            JOIN AdmissionType AS T ON A.AdmissionType = T.AdmissionTypeID
            JOIN Department AS D ON A.Department = D.DeptId
            WHERE A.Administrator = %s
            ORDER BY A.DischargeDate DESC NULLS LAST, A.Patient ASC,
T.AdmissionTypeName DESC
        """
```

```
        cur.execute(query, (login,))
        admissions = cur.fetchall()
        if not admissions:
            return []  # Return an empty list if there are no records
        details = [desc[0] for desc in cur.description]
        dictionary = [dict(zip(details, row)) for row in admissions]  # Convert to
a list of dictionaries
        return dictionary
    except Exception as e:
        return []
    finally:
        cur.close()
        conn.close()

def findAdmissionsByCriteria(searchString):
    conn = openConnection()
    cur = conn.cursor()
    try:
        # Search for admissions based on criteria
        query = """
            SELECT A.AdmissionID AS admission_id,
                   T.AdmissionTypeName AS admission_type,
                   D.DeptName AS admission_department,
                   COALESCE(TO_CHAR(A.DischargeDate, 'DD-MM-YYYY'), '') AS
discharge_date,
                   COALESCE(A.Fee::text, '') AS fee,
                   A.Patient AS patient,
                   COALESCE(A.Condition, '') AS condition
            FROM Admission AS A
            JOIN AdmissionType AS T ON A.AdmissionType = T.AdmissionTypeID
            JOIN Department AS D ON A.Department = D.DeptId
            WHERE (T.AdmissionTypeName ILIKE %s OR D.DeptName ILIKE %s
                  OR A.Patient ILIKE %s OR A.Condition ILIKE %s)
            AND (A.DischargeDate IS NULL OR A.DischargeDate >= CURRENT_DATE -
INTERVAL '2 years')
            ORDER BY A.DischargeDate DESC NULLS LAST, A.Patient ASC
        """
        keyword = f"%{searchString}%"  # Fuzzy search keyword
        cur.execute(query, (keyword, keyword, keyword, keyword))
        results = cur.fetchall()
        return [dict(zip([desc[0] for desc in cur.description], row)) for row in
results] if results else []
    except (psycopg2.Error, ValueError) as e:
        return []
    finally:
        cur.close()
        conn.close()

def addAdmission(type_name, department_name, patient_name, condition, admin):
    conn = openConnection()
    cur = conn.cursor()
    try:
        # Add a new admission record
        cur.execute("CALL add_admission(%s, %s, %s, %s, %s)",
                    (type_name, department_name, patient_name, admin, condition))
        conn.commit()
        return True
    except (psycopg2.Error, ValueError) as e:
        return False
```

```
    finally:
        cur.close()
        conn.close()

def updateAdmission(id, type_name, department_name, dischargeDate, fee, patient,
condition):
    conn = openConnection()
    cur = conn.cursor()
    try:
        # Update a new admission record
        cur.execute("CALL update_admission(%s, %s, %s, %s, %s, %s, %s)",
                    (id, type_name, department_name, dischargeDate, fee, patient,
condition))
        conn.commit()
        return True
    except (psycopg2.Error, ValueError) as e:
        return False
    finally:
        cur.close()
        conn.close()
```