```
DROP TABLE IF EXISTS Administrator CASCADE;
DROP TABLE IF EXISTS Patient CASCADE;
DROP TABLE IF EXISTS AdmissionType CASCADE;
DROP TABLE IF EXISTS Department CASCADE;
DROP TABLE IF EXISTS Admission CASCADE;

CREATE TABLE Administrator (
    UserName VARCHAR(10) PRIMARY KEY,
    Password VARCHAR(20) NOT NULL,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Email VARCHAR(20) NOT NULL
);

INSERT INTO Administrator VALUES
('jdoe', 'Pass1234', 'John', 'Doe', 'jdoe@csh.com'),
('jsmith', 'Pass5678', 'Jane', 'Smith', 'jsmith@csh.com'),
('ajohnson', 'Passabcd', 'Alice', 'Johnson', 'ajohnson@csh.com'),
('bbrown', 'Passwxyz', 'Bob', 'Brown', 'bbrown@csh.com'),
('cdavis', 'Pass9876', 'Charlie', 'Davis', 'cdavis@csh.com'),
('ksmith', 'Pass5566', 'Karen', 'Smith', 'ksmith@csh.com');

CREATE TABLE Patient (
    PatientID VARCHAR(10) PRIMARY KEY,
    Password VARCHAR(20) NOT NULL,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Mobile VARCHAR(20) NOT NULL
);

INSERT INTO Patient VALUES
('dwilson', 'Pass5432', 'David', 'Wilson', '4455667788'),
('etylor', 'Passlmno', 'Eva', 'Taylor', '5566778899'),
('faderson', 'Passrstu', 'Frank', 'Anderson', '6677889900'),
('gthomas', 'Pass1357', 'Grace', 'Thomas', '7788990011'),
('smartinez', 'Pass2468', 'Stan', 'Martinez', '8899001122'),
('lroberts', 'Pass1122', 'Laura', 'Roberts', '9900112233');


CREATE TABLE AdmissionType (
    AdmissionTypeID SERIAL PRIMARY KEY,
    AdmissionTypeName VARCHAR(20) UNIQUE NOT NULL
);

INSERT INTO AdmissionType VALUES (1, 'Emergency');
INSERT INTO AdmissionType VALUES (2, 'Transfer');
INSERT INTO AdmissionType VALUES (3, 'Inpatient');
INSERT INTO AdmissionType VALUES (4, 'Outpatient');

CREATE TABLE Department (
    DeptId SERIAL PRIMARY Key,
    DeptName VARCHAR(20) UNIQUE not NULL
);

INSERT INTO Department VALUES (1, 'General');
INSERT INTO Department VALUES (2, 'Emergency');
INSERT INTO Department VALUES (3, 'Surgery');
INSERT INTO Department VALUES (4, 'Obstetrics');
INSERT INTO Department VALUES (5, 'Rehabilitation');
INSERT INTO Department VALUES (6, 'Paediatrics');

CREATE table Admission (
    AdmissionID SERIAL PRIMARY KEY,
    AdmissionType INTEGER NOT NULL,
    Department INTEGER NOT NULL,
      Patient VARCHAR(10) NOT NULL,
      Administrator VARCHAR(10) NOT NULL,
    Fee Decimal(7,2),
    DischargeDate Date,
    Condition VARCHAR(500),
      FOREIGN KEY(AdmissionType) REFERENCES AdmissionType,
      FOREIGN KEY(Department) REFERENCES Department,
      FOREIGN KEY(Patient) REFERENCES Patient,
      FOREIGN KEY(Administrator) REFERENCES Administrator
);

INSERT INTO Admission (AdmissionType, Department, Fee, Patient, Administrator,
DischargeDate, Condition) VALUES
    (4, 1, 666.00, 'lroberts', 'jdoe', '28/02/2024', 'a red patch on my skin that
looks irritated. It started small but has been spreading and feels warm to the
touch'),
      (2, 1, 100.00, 'gthomas', 'jdoe', '11/09/2021', NULL),
      (1, 2, NULL, 'lroberts','jsmith', '02/09/2019', 'Admitted to the emergency
department after suffering head trauma from a fall, requiring a CT scan and
observation for potential concussion.'),
      (2, 3, 7688.00, 'dwilson','ajohnson', '01/12/2022', NULL),
      (2, 6, 1600.00, 'faderson', 'ajohnson', '03/09/2014', 'Child admitted to the
hospital with a severe asthma attack, requiring oxygen therapy and nebulizer
treatment.'),
      (4, 1, 90.00, 'gthomas', 'ksmith', '04/07/2021', 'Routine follow-up
consultation to review progress after recent knee surgery, with positive recovery
observed.'),
      (1, 2, 1450.00, 'smartinez', 'jsmith', NULL, 'Admitted to the emergency
department with severe food poisoning, requiring IV fluids and anti-nausea
medication for recovery.'),
      (4, 5, 180.95, 'dwilson', 'cdavis', '06/11/2021', 'Attended a physiotherapy
session as part of an ongoing rehabilitation program following shoulder surgery.'),
      (3, 1, 2000.00, 'etylor', 'ajohnson', '10/09/2021', NULL),
      (2, 4, 8290.00, 'gthomas', 'jsmith', '01/09/2024', 'Postpartum care following
a natural childbirth, including monitoring of both the mother and the newborn for
potential complications.'),
      (2, 6, 1800.00, 'faderson', 'bbrown',  NULL, 'Child admitted to the
paediatrics department for severe pneumonia, requiring intravenous antibiotics and
respiratory therapy.'),
      (4, 1, 75.00, 'gthomas', 'bbrown', '19/11/2023', 'Routine general
practitioner consultation for a follow-up after a recent bout of seasonal
allergies.'),
      (3, 3, 7000.50, 'smartinez', 'jdoe', '15/10/2024', NULL),
      (1, 2, NULL, 'etylor', 'jdoe', NULL, 'I am having intense, crushing pain in
my chest that feels like an elephant is sitting on it. It is spreading to my left
arm and neck.');

SET datestyle = 'DMY';

DROP PROCEDURE IF EXISTS add_admission;
DROP PROCEDURE IF EXISTS update_admission;

-- Create or replace the add_admission procedure
```

```
CREATE OR REPLACE PROCEDURE add_admission(
    p_type_name VARCHAR,        -- Admission type name
    p_department_name VARCHAR,  -- Department name
    p_patient_name VARCHAR,     -- Patient name
    p_admin VARCHAR,            -- Administrator name
    p_condition TEXT            -- Condition description
)
LANGUAGE plpgsql AS $$
DECLARE
    type_id INT;               -- Admission type ID
    department_id INT;         -- Department ID
BEGIN
    -- Find the corresponding ID based on the admission type name
    SELECT AdmissionTypeID INTO type_id
    FROM AdmissionType
    WHERE AdmissionTypeName ILIKE p_type_name;

    -- Raise an exception if the admission type is not found
    IF type_id IS NULL THEN
        RAISE EXCEPTION 'Admission type % not found', p_type_name;
    END IF;

    -- Find the corresponding ID based on the department name
    SELECT DeptId INTO department_id
    FROM Department
    WHERE DeptName ILIKE p_department_name;

    -- Raise an exception if the department is not found
    IF department_id IS NULL THEN
        RAISE EXCEPTION 'Department % not found', p_department_name;
    END IF;

    -- Check if the patient exists; if not, insert a new patient
    IF NOT EXISTS (SELECT 1 FROM Patient WHERE PatientID = p_patient_name) THEN
        INSERT INTO Patient (PatientID, Password, FirstName, LastName, Mobile)
        VALUES (p_patient_name, 'default123', p_patient_name, p_patient_name, 
'0000000000');
    END IF;

    -- Insert a new admission record
    INSERT INTO Admission (AdmissionType, Department, Patient, Administrator, 
Condition)
    VALUES (type_id, department_id, p_patient_name, p_admin, p_condition);
END $$;


-- Create or replace the update_admission procedure
CREATE OR REPLACE PROCEDURE update_admission(
    p_admission_id INT,        -- Admission record ID
    p_type_name VARCHAR,       -- Admission type name
    p_department_name VARCHAR,  -- Department name
    p_discharge_date VARCHAR,  -- Discharge date
    p_fee VARCHAR,             -- Fee
    p_patient_name VARCHAR,    -- Patient name
    p_condition TEXT           -- Condition description
)
LANGUAGE plpgsql AS $$
DECLARE
    type_id INT;               -- Admission type ID
    department_id INT;         -- Department ID
    fee_value NUMERIC;         -- Fee value
BEGIN
    -- Find the corresponding ID based on the admission type name
    SELECT AdmissionTypeID INTO type_id
    FROM AdmissionType
    WHERE AdmissionTypeName ILIKE p_type_name;

    -- Raise an exception if the admission type is not found
    IF type_id IS NULL THEN
        RAISE EXCEPTION 'Admission type % not found', p_type_name;
    END IF;

    -- Find the corresponding ID based on the department name
    SELECT DeptId INTO department_id
    FROM Department
    WHERE DeptName ILIKE p_department_name;

    -- Raise an exception if the department is not found
    IF department_id IS NULL THEN
        RAISE EXCEPTION 'Department % not found', p_department_name;
    END IF;

    -- Convert fee to numeric, empty string to NULL
    fee_value := NULLIF(p_fee, '')::numeric;

    -- Update the admission record
    UPDATE Admission
    SET AdmissionType = type_id,
        Department = department_id,
        DischargeDate = NULLIF(p_discharge_date, '')::date, -- Convert empty string
to NULL
        Fee = fee_value,
        Patient = p_patient_name,
        Condition = p_condition
    WHERE AdmissionID = p_admission_id;
END $$;
```