Lab Test:

```
CREATE TABLE Lab_Tests (
    TestID INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY, -- Auto-incrementing Test ID
    PatientID INT NOT NULL,                              -- Reference to Patients table
    DoctorID INT NOT NULL,                               -- Reference to Doctors table
    TestType VARCHAR(255) NOT NULL,                     -- Type of test (e.g., Blood Test, X-Ray)
    TestDate DATE DEFAULT SYSDATE,                      -- Date of the test
    Results CLOB,                                        -- Test results
    Status VARCHAR(50) DEFAULT 'Pending',               -- Test status (Pending/Completed)
    FOREIGN KEY (PatientID) REFERENCES Patients(PATIENT_ID),
    FOREIGN KEY (DoctorID) REFERENCES Doctors(DOCTOR_ID)
);
```

Emergency Service:

```
SELECT
    es.Service_ID,
    p.Name AS Patient_Name,
    p.Contact AS Patient_Contact,
    es.Type AS Emergency_Type,
    d.Name AS Doctor_Name,
    r.Type AS Resource_Type,
    es.Response_Time,
    es.Date_of_Entry
FROM
    YOUR_SCHEMA.EMERGENCY_SERVICE es
JOIN
    YOUR_SCHEMA.PATIENT p ON es.Patient_ID = p.Patient_ID
JOIN
    YOUR_SCHEMA.DOCTOR d ON es.Doctor_ID = d.Doctor_ID
JOIN
    YOUR_SCHEMA.RESOURCE_ALLOCATION r ON es.Resource_ID = r.Resource_ID
WHERE
    es.Date_of_Entry >= SYSDATE - 30
ORDER BY
    es.Date_of_Entry DESC;
```

Medical Record:

```
CREATE TABLE Medical_Records (
    PatientID INT NOT NULL,             -- Reference to the patient
    DoctorID INT NOT NULL,              -- Reference to the doctor
    Diagnosis VARCHAR(255),             -- Diagnosis information
    Treatment CLOB,                     -- Details of the treatment
    Date_of_Record DATE DEFAULT SYSDATE, -- Automatically use the current date
    Notes CLOB,                         -- Additional notes (optional)
    Blood_Pressure VARCHAR(10),         -- Example: 120/80
    Heart_Rate INT,                     -- Heartbeats per minute
    Temperature DECIMAL(5,2),           -- Body temperature (e.g., 98.6)
    Oxygen_Saturation DECIMAL(5,2),     -- Oxygen level in percentage
    PRIMARY KEY (PatientID, Date_of_Record) -- Ensures one record per patient per day
);
```

Log Donation:

```
CREATE TABLE DONATIONS (
    DONATION_ID NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY, -- Unique ID for each
    DONOR_NAME VARCHAR2(100) NOT NULL,                               -- Name of the donor
    DONATION_TYPE VARCHAR2(50) NOT NULL,                             -- Type of donation (e.g., Blood, Money, Equipment)
    DONATION_DATE DATE DEFAULT SYSDATE,                              -- Date of the donation
    PATIENT_ID NUMBER,                                               -- Optional: Patient ID for specific patient-related donations
    AMOUNT NUMBER                                                   -- Amount donated (for monetary donations)
);
```

Billing:

```
CREATE TABLE Billing (
    Billing_ID NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY, -- Unique Billing ID
    Patient_ID NUMBER NOT NULL, -- Foreign key to Patient table
    Service_Type VARCHAR(50) NOT NULL, -- Foreign key to Charges table
    Insurance_ID NUMBER, -- Foreign key to Insurance_Company table
    Billing_Date DATE DEFAULT SYSDATE, -- Date of the billing record
    Amount NUMBER NOT NULL, -- Amount associated with the billing
    Payment_Status VARCHAR2(20) DEFAULT 'Pending', -- Payment status (e.g., Pending, Paid)
    CONSTRAINT FK_Patient_ID FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID), -- Links to Patient table
    CONSTRAINT FK_Service_Type FOREIGN KEY (Service_Type) REFERENCES Charges (Service_Type), -- Links to Charges table
    CONSTRAINT FK_Insurance_ID FOREIGN KEY (Insurance_ID) REFERENCES Insurance_Company (Insurance_ID) -- Links to Insurance_Company table
);
```

Insurance Company:

```
CREATE TABLE Insurance_Company (
    Insurance_ID NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY, -- Unique identifier for insurance company
    Company_Name VARCHAR2(100) NOT NULL, -- Name of the insurance company
    Coverage_Amount NUMBER(10, 2) NOT NULL, -- Coverage amount provided by the insurance
    Contact_Info VARCHAR2(150) -- Contact information (phone/email)
);
```

Surgery:

```
CREATE TABLE Surgery (
     Patient_ID INT NOT NULL, -- Foreign key referencing Patient table
    Doctor_ID INT NOT NULL, -- Foreign key referencing Doctor table
    Surgery_ID INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY, -- Unique identifier for the surgery
    Surgery_Description VARCHAR(255) NOT NULL, -- Description of the surgery

    Surgery_Date DATE NOT NULL, -- Date of the surgery
    Surgery_Time TIMESTAMP NOT NULL, -- Date and time of the surgery
    Operating_Room VARCHAR(50) NOT NULL, -- Operating room number or name

    CONSTRAINT FK_Patient FOREIGN KEY (Patient_ID) REFERENCES Patient(Patient_ID),
    CONSTRAINT FK_Doctor FOREIGN KEY (Doctor_ID) REFERENCES DOCTORS(DOCTOR_ID)
);
```

Appointments:

```
CREATE TABLE APPOINTMENTS (
    APPOINTMENT_ID NUMBER PRIMARY KEY,
    PATIENT_ID NUMBER,
    DOCTOR_ID NUMBER,
    APPOINTMENT_DATE DATE,
    APPOINTMENT_TIME VARCHAR2(50),
    STATUS VARCHAR2(50),
    FOREIGN KEY (PATIENT_ID) REFERENCES PATIENTS (PATIENT_ID),
    FOREIGN KEY (DOCTOR_ID) REFERENCES DOCTORS (DOCTOR_ID)
);
```

Patient:

```
CREATE TABLE Patient (
    Patient_ID NUMBER PRIMARY KEY,          -- Primary Key for identifying patients
    Patient_Name VARCHAR2(255) NOT NULL,     -- Patient's name, not null
    Date_of_Birth DATE NOT NULL,             -- Date of birth, not null
    Contact_Number VARCHAR2(15),             -- Contact number, can be null
    Email VARCHAR2(255),                     -- Email address, can be null
    Address VARCHAR2(500)                    -- Address, can be null
);
```

Doctor:

```
CREATE TABLE DOCTORS (
    DOCTOR_ID NUMBER PRIMARY KEY,
    NAME VARCHAR2(100) NOT NULL,
    SPECIALTY VARCHAR2(100),
    AVAILABILITY_SCHEDULE VARCHAR2(255)
);
INSERT INTO DOCTORS (DOCTOR_ID, NAME, SPECIALTY, AVAILABILITY_SCHEDULE)
VALUES (1, 'Dr. John Smith', 'Cardiologist', 'Mon-Fri 9:00 AM - 5:00 PM');

INSERT INTO DOCTORS (DOCTOR_ID, NAME, SPECIALTY, AVAILABILITY_SCHEDULE)
VALUES (2, 'Dr. Emily Davis', 'Dermatologist', 'Mon, Wed, Fri 10:00 AM - 2:00 PM');

INSERT INTO DOCTORS (DOCTOR_ID, NAME, SPECIALTY, AVAILABILITY_SCHEDULE)
VALUES (3, 'Dr. Robert Lee', 'Orthopedic Surgeon', 'Tue-Thu 8:00 AM - 12:00 PM');
```