1) **Print a list of all doctors based at a particular hospital.**

```
BEGIN
      // Step 1: Input hospital details
      INPUT hospital_id OR hospital_name

      // Step 2: Validate input – hospital exists
      IF hospital_name IS PROVIDED THEN
            QUERY HOSPITALS
            WHERE hospital_name = input hospital_name
            SET hospital_id = input_hospital_id
      END IF

      IF hospitals_id IS NOT FOUND THEN
      OUTPUT "Hospital not found."
      END
      END IF

      // Step 3: Retrieve doctors
      QUERY DOCTORS
      SELECT first_name, last_name, address
      WHERE hospital_id = input_hospital_id

      // Step 4: Output list of doctors
      IF doctors exist THEN
      OUTPUT "List of doctors: "
      DISPLAY first_name, last_name, address
      ELSE
      OUTPUT "No doctors found for specified hospital."
```

```
END
```

2) **Print a list of all prescriptions for a particular patient, ordered by the prescription date.**

```
BEGIN
```

```
// Step 1: Input patient details
INPUT patient_id

// Step 2: Validate input – patient exists
IF patient_id IS NOT PROVIDED THEN
QUERY PATIENTS
WHERE first_name = input_first_name AND last_name =
input_last_name
IF NO RECORDS FOUND THEN
            OUTPUT “Patient not found.”
            END
      ELSE
            SET patient_id = input_patient_id
      END_IF
END_IF

// Step 3: Retrieve prescriptions for patient (and order
ascending)
QUERY PRESCRIPTIONS
SELECT med_id, prescription_date
WHERE patient_id = input_patient_id
ORDER BY prescription_date ASC

// Step 4: Join medications for prescriptions
QUERY MEDICATIONS
SELECT med_name
JOIN WITH PRESCRIPTIONS on MEDICATIONS.med_id =
PRESCRIPTIONS.med_id
WHERE patient_id = input_patient_id
ORDER by prescription_date ASC
```

```
      // Step 5: Output the prescriptions
      IF prescriptions exist THEN
      OUTPUT "List of prescriptions: "
      DISPLAY med_name, prescription_date
      ELSE
      OUTPUT "No prescriptions found for specified patient."
END
```

3) **Print a list of all prescriptions that a particular doctor has prescribed.**

```
BEGIN
      // Step 1: Input doctor details
      INPUT doctor_id
```

```
      // Step 2: Validate input – does the doctor exist?
      IF doctor_id IS NOT PROVIDED THEN
      QUERY DOCTORS
      WHERE first_name = input_first_name AND last_name =
input_last_name
      IF NO RECORDS FOUND THEN
                  OUTPUT "Doctor not found."
                  END
            ELSE
                  SET doctor_id = input_doctor_id
            END_IF
      END_IF

      // Step 3: Retrieve prescriptions for patient (and order
ascending)
      QUERY PRESCRIPTIONS
      SELECT med_id
      WHERE doctor_id = input_doctor_id
      ORDER BY med_id ASC

      // Step 4: Join medications for prescriptions
      QUERY MEDICATIONS
      SELECT med_name
      JOIN WITH PRESCRIPTIONS on MEDICATIONS.med_id =
PRESCRIPTIONS.med_id
      WHERE doctor_id = input_doctor_id
      ORDER by prescription_date ASC

      // Step 5: Output the prescriptions
```

```
      IF prescriptions exist THEN
      OUTPUT "List of prescriptions: "
      DISPLAY med_name
      ELSE
      OUTPUT "No prescriptions found for specified doctor."
END
```

4) **Print a table showing all prescriptions ordered by the patient name alphabetically.**

```
BEGIN
      // Step 1: Query patient details
      QUERY PATIENTS
            SELECT first_name, last_name
```

```
      // Step 2: Join prescriptions for patients
      QUERY PRESCRIPTIONS
      SELECT patient_name
      JOIN WITH PRESCRIPTION on patient_name
            ORDER BY patient last_name ASC,
      ORDER BY patient first_name ASC

      // Step 3: Output the prescriptions, ordered by patient name
      OUTPUT "List of prescriptions: "
      DISPLAY med_name
END
```

QUERY NAMES NEEDED FOR FOLLOWING QUESTIONS, IF NOT ALREADY INCLUDED

5) **Add a new customer to the database, including being registered with one of the doctors.**

```
BEGIN
      //Step 1: Validate that the patient is not on the PATIENTS
database
      SELECT first_name, last_name, date_of_birth, address
      IF patient_exists(first_name, last_name, date_of_birth, address)
      THEN
            OUTPUT "Patient is already in database"
            END
```

```
ELSE
      INPUT first_name, last_name, date_of_birth, address,
doctor_id

// Step 2: Validate doctor_id from within DOCTORS database
QUERY DOCTORS
WHERE doctor_id = input_doctor_id
IF NO RECORDS FOUND THEN
            OUTPUT "Doctor not found."
            END
      ELSE
            SET last_name = input_patient_name
            SET first_name = input_first_name
            SET address = input_address
            GENERATE unique patient_id =
      generate_unique_patient_id
      END_IF
END
```

6) **Modify address details of an existing customer.**

```
BEGIN
      //Step 1: Select the patient to update address and confirm they
are on the PATIENTS database
      SELECT patient_id
      IF patient_not_in_database(patient_id)
      THEN
            OUTPUT "Patient is not in the database"
```

```
      ELSE
            SELECT patient_id, address from PATIENTS

      // Step 2: Modify address of selected patient on PATIENTS
database
      SET address = new_address
      WHERE patient_id = patient_id

      // Step 3: Confirm the address has been changed
      IF update_successful
      SELECT address FROM PATIENTS WHERE patient_id = patient id
      OUTPUT "address has been changed.  Address: + address"
      ELSE
      OUTPUT "Error changing the address"
      END IF
END
```

7) **Print a list of all patient names and addresses for patients registered to doctors based at one particular hospital - that could be used for posting information mail to all of one hospital's registered patients.**

```
BEGIN
Input: "Hospital Name" (e.g., 'Royal London Hospital')

Load Data:
   a. Load `cleaned_patients` table containing:
      - patient_id, first_name, last_name, address, doctor_id
   b. Load `doctors` table containing:
```

- doctor_id, hospital_id

c. Load `hospitals` table containing:

- hospital_id, Hospital Name

Filter Data:

a. Find the `hospital_id` in the `hospitals` table where "Hospital Name" matches the input.

b. Identify all `doctor_id` values in the `doctors` table where `hospital_id` matches the result from Step 3a.

Join Data:

a. Join the `cleaned_patients` table with the filtered `doctor_id` values from Step 3b.

5. Output:

a. Select and print the `first_name`, `last_name`, and `address` columns for all matched patients.

END

8) **Print a list of all doctors based at Teaching hospitals which were accredited between 2015-2024.**

```
BEGIN

    // Step 1: Input hospital type + accrediation range

    INPUT start_year, end_year, hospital_type

        SET start_year = 2015
        SET end_year = 2024
        SET hospital_type = "Teaching"


    // Step 2: Query HOSPITALS table - retrieve teaching hospitals
    accredited in date range
```

```
QUERY HOSPITALS

SELECT hospital_id

WHERE type = hospital_type AND accreditation_year BETWEEN
start_year AND end_year

STORE RESULT IN hospital_list

// Step 3: Validate if any hospitals match criteria

IF hospital_list is EMPTY THEN

      OUTPUT “No teaching hospitals accredited 2015-2024 found.”

      EXIT

END IF

// Step 4: Query DOCTORS table - retrieve doctors linked to the
hospitals

QUERY DOCTORS

SELECT doctor_id, first_name, second_name

WHERE hospital_id IN hospital_list

STORE RESULT IN doctor_list



// Step 5: Output list of doctors

IF doctor_list IS NOT EMPTY THEN

      OUTPUT “List of doctors at teaching hospitals accredited
      2015-2024:”

      DISPLAY doctor_id, first_name, second_name

ELSE
```

```
            OUTPUT "No doctors found in teaching hospitals accredited
            2015-2024."

      END IF

END
```

9) **List all patients who may have a particular disease based on which medication they have been prescribed.**

```
BEGIN

      // Step 1: Input disease details

      INPUT disease_id OR disease_name


      // Step 2: Query DISEASE table -retrieve medications associated
      with the disease

      IF disease_id IS PROVIDED THEN

            QUERY DISEASE

            SELECT med_id

            WHERE disease_id = input_disease_id
```

```
ELSE

      QUERY DISEASE

      SELECT med_id

      WHERE disease_name = input_disease_name

      SET disease_id = retrieved_disease_id

END IF


// Step 3: Validate if any medications are associated with the
disease

IF NO med_id RECORDS FOUND THEN

      OUTPUT “No medications found for specified disease.”

      EXIT

END IF



// Step 4: Query PRESCRIPTIONS and PATIENTS tables to retrieve
patients prescribed the medication

QUERY PRESCRIPTIONS

SELECT P.patient_id, PT.first_name, PT.second_name, PT.address

      FROM PRESCRIPTIONS P

      JOIN PATIENTS PT ON P.patient_id = PT.patient_id

      WHERE P.med_id IN (retrieved med_ids)


// Step 5: Output the list of patients

      IF records EXIST THEN
```

```
            OUTPUT "List of patient names and addresses:"

            DISPLAY first_name, second_name, address

      ELSE

            OUTPUT "No patients found for specified disease and
            prescribed medications."

      END IF

END
```

10) **List all doctors based at who specialize in a particular disease.**

```
BEGIN
      // Step 1: Input disease details


      INPUT disease_id OR disease_name


      // Step 2: Query DISEASE table to retrieve doctor(s)
      associated with the disease


      IF disease_id IS PROVIDED THEN


QUERY DISEASE


SELECT doctor_id


WHERE disease_id = input_disease_id
```

```
     ELSE

QUERY DISEASE

SELECT doctor_id

WHERE disease_name = input_disease_name

SET disease_id = retrieved_disease_id

     END IF




     // Step 3: Validate if any doctors are associated with the
     disease

     IF NO doctor_id RECORDS FOUND THEN

OUTPUT "No doctors found who specialise in specified
     disease."

EXIT

     END IF

     // Step 4: Query DOCTORS table to get doctor details
```

```
     QUERY DOCTORS

     SELECT doctor_id, first_name, second_name, address,
     hospital_id

     WHERE doctor_id IN (retrieved doctor_ids)

     // Step 5: Output the list of doctors

     IF doctor records EXIST THEN

OUTPUT "List of doctors specialising in the disease:"

DISPLAY doctor_id, first_name, second_name, address,   h
     ospital_id

     ELSE

OUTPUT "No doctors found who specialise in specified   disease."

     END IF

END
```

11) **List all lab results for all patients over the age of 60.**

```
BEGIN

      // Step 1: Calculate cutoff birth date for patients over 60

      SET current_date = TODAY()

      SET cutoff_birth_date = DATE_SUBTRACT(current_date, INTERVAL 60
      YEARS)


      // Step 2: Query PATIENTS table to find patients over 60 years
      old

      QUERY PATIENTS

            SELECT patient_id, first_name, second_name, date_of_birth

            WHERE date_of_birth <= cutoff_birth_date

            STORE RESULT IN patient_list


      // Step 3: Check if any patients over 60 were found

      IF patient_list IS EMPTY THEN

            OUTPUT “No patients found over the age of 60.”

            EXIT
```

```
    END IF


    // Step 4: Query LAB_RESULTS table to retrieve lab results for
    identified patients

    QUERY LAB_RESULTS

        SELECT lab_results_id, lab_test, lab_date, lab_result,
        patient_id

        WHERE patient_id IN (patient_list)



    // Step 5: Output lab results

    IF records EXIST THEN

        OUTPUT “List of lab results for all patients over 60:”

        DISPLAY lab_results_id, lab_test, lab_date, lab_result,
        patient_id

    ELSE

        OUTPUT “No lab results found for patients over 60.”

    END IF

END
```

12) **Print a list of all appointments for a given patient.**

```
BEGIN
      // Step 1: Input patient details
      INPUT patient_id OR patient_name (first_name and second_name)

      // Step 2: Validate input – check if patient exists
      IF patiend_id IS PROVIDED THEN
      SET patient_id = input_patient_id
      ELSE
QUERY "ValidatePatientByName”
SELECT patient_id
FROM PATIENTS
WHERE LOWER(first_name) = LOWER(input_first_name) AND
LOWER(second_name) = LOWER(input_second_name)
           SET patient_id = retrieved_patient_id

           IF patient_id IS NULL THEN
OUTPUT “Patient not found.”
RAISE ERROR (e.g., SIGNAL SQLSTATE in SQL)
EXIT
END IF
END IF
```

```
// Step 3: Retrieve appointments for the patient
QUERY "GetAppointmentsForPatient”
SELECT appt_date, appt_purpose, doctor_id
FROM APPOINTMENTS
WHERE patient_id = patient_id
ORDER BY appt_date ASC

// Step 4: Join with doctors table to get doctor details
QUERY "GetDoctorDetailsForAppointments”
SELECT CONCAT(d.first_name, ‘ ’, d.second_name) AS DoctorName
FROM DOCTORS d
JOIN APPOINMENTS a ON d.doctor_id = a.doctor_id
WHERE a.patient_id = patient_id
ORDER BY a.appt_date ASC

// Step 5: Output the appointments
IF records exist THEN
OUTPUT “List of appointmnets:”
DISPLAY appt_date, appt_purpose, doctor_name
ELSE
OUTPUT “No appointments found for the specified patient.”   END IF
END
```

13) **Print a list of all appointments for a given doctor.**

```
BEGIN
// Step 1: Input doctor details
INPUT doctor_id OR doctor_name (first_name and second_name)

// Step 2: Validate input - check if doctor exists
IF doctor_id IS PROVIDED THEN
SER doctor_id = input_doctor_id
ELSE
QUERY "ValidateDoctorByName”
SELECT doctor_id
FROM DOCTORS
WHERE LOWER(first_name) = LOWER(input_first_name)
AND LOWER(second_name) = LOWER(input_second_name)
          SET doctor_id = retrieved_doctor_id

IF doctor_id IS NULL THEN
OUTPUT “Doctor not found.”
RAISE ERROR (e.g., SIGNAL SQLSTATE in SQL)
EXIT
END IF
END IF
```

```
// Step 3: Retrieve appointments and patient details for the
doctor

QUERY "GetAppointmentsAndPatientsForDoctor”

SELECT

               appt_date,

               appt_purpose,

               CONCAT(patient_first_name, ‘ ’,
          patient_second_name) AS PatientName

FROM APPOINTMENTS

JOIN PATIENTS ON APPOINTMENTS.patient_id = PATIENTS.patient_id

     WHERE APPOINTMENTS.doctor_id = doctor_id

ORDER BY appt_date ASC


// Step 4: Output the appointments

IF records exist THEN

OUTPUT “List of appointments for the doctor:”

DISPLAY appt_date, appt_purpose, patient_name

ELSE

OUTPUT “No appointments found for the specified doctor.”

END IF

END
```

14) **Print all prescriptions made from a particular hospital ordered alphabetically by the name of the medication being prescribed - The output of this SQL query should include only these 4 columns: the medication name, the name of doctor who prescribed it, the name of the patient, and the name of hospital.**

```
BEGIN
// Step 1: Input hospital details
INPUT hospital_name

// Step 2: Validate input – check if hospital exists
QUERY "ValidateHospitalByName”
SELECT hospital_id
FROM HOSPITALS
WHERE LOWER(hospital_name) = LOWER(input_hospital_name)

           SET hospital_id = retrieved_hospital_id

IF hospital_id IS NULL THEN
OUTPUT “Hospital not found.”
RAISE ERROR (e.g., SIGNAL SQLSTATE in SQL)
EXIT
END IF

// Step 3: Retrieve prescriptions details
QUERY "GetPrescriptionsByHospital”
SELECT
                MEDICATIONS.med_name AS medication_name,
CONCAT(DOCTORS.first_name, ‘ ’, DOCTORS.second_name) AS doctor_name,
```

```
CONCAT(PATIENTS.first_name, ' ', PATIENTS.second_name) AS
patient_name,

HOSPITALS.hospital_name AS hospital_name

FROM PRESCRIPTIONS

JOIN MEDICATIONS ON PRESCRIPTIONS.med_id = MEDICATIONS.med_id

JOIN DOCTORS ON PRESCRIPTIONS.doctor_id = DOCTORS.doctor_id

JOIN PATIENTS ON PRESCRIPTIONS.patient_id = PATIENTS.patient_id

JOIN HOSPITALS ON DOCTORS.hospital_id = HOSPITALS.hospital_id

WHERE HOSPITALS.hospital_id = hospital_id

ORDER BY MEDICATIONS.med_name ASC


// Step 4: Output the prescriptions

IF records exist THEN

OUTPUT "List of prescriptions:"

DISPLAY medication_name, doctor_name, patient_name, hospital_name

ELSE

OUTPUT "No prescriptions found for the specified hospital."

END IF

END
```

15) **Print a list of all lab results from all hospitals that were accredited between 2013- 2020.**

```
BEGIN

// Step 1: Retrieve lab results for hospitals accredited between 2013
and 2020

QUERY "ListLabResultsByAccredited"

SELECT

LAB_RESULTS.lab_test AS lab_test_name,

LAB_RESULTS.lab_date AS lab_test_date,

LAB_RESULTS.lab_result AS lab_test_result,

CONCAT(PATIENTS.first_name, ' ', PATIENTS.second_name) AS
patient_name,

CONCAT(DOCTORS.first_name, ' ', DOCTORS.second_name) AS doctor_name,

FROM LAB_RESULTS

JOIN PATIENTS ON LAB_RESULTS.patient_id = PATIENTS.patient_id

JOIN DOCTORS ON LAB_RESULTS.doctor_id = DOCTORS.doctor_id

JOIN HOSPITALS ON DOCTORS.hospital_id = HOSPITALS.hospital_id

WHERE HOSPITALS.accreditation_year BETWEEN 2013 AND 2020;


// Step 2: Output the lab results

IF records exist THEN

OUTPUT "List of lab results from hospitals accredited between 2013 and
2020:"

DISPLAY lab_test_name, lab_test_date, lab_test_result, patient_name,
doctor_name, hospital_name

ELSE

OUTPUT "No lab results found for hospitals accredited between 2013 and
2020."

END IF

END
```

16) **Identify which doctor has made the most prescriptions.**

```
BEGIN
// Step 1: Count prescriptions for each doctor
QUERY "CountPrescriptionsPerDr”
          SELECT doctor_id, COUNT(prescription_id)AS
     prescription_count
          FROM PRESCRIPTIONS
GROUP BY doctor_id
          ORDER by prescription_count DESC
          LIMIT 1
          SET doctor_w_max_presc = retrieved_doctor_id

// Step 2: Validate doctor exists
IF doctor_w_max_presc IS NULL THEN
OUTPUT “No doctor found.”
END
END IF

// Step 3: Retrieve doctor details
QUERY "GetDrDetailsWithMostPrescriptions”
SELECT first_name, second_name, address
FROM DOCTORS
WHERE doctor_id = doctor_w_max_presc

// Step 4: Output
IF doctor details exist THEN
OUTPUT “Doctor with the most prescriptions:”
DISPLAY first_name, second_name, address
ELSE
OUTPUT “No doctor details found.”
```

```
END IF
END
```

17) **Print a list of all doctors at the hospital with biggest size (number of beds).**

```
BEGIN
```

```
// Step 1: Retrieve hospital with largest size by number of beds
QUERY "GetHospitalWithLargestSize
SELECT hosptial_id, hospital_name, size
FROM HOSPITALS
WHERE size = MAX(size)
SET largest_hospital_id = retrieved hospital_id
SET largest_size = retrieved size

// Step 2: Validate hospital exists
IF largest_hospital_id IS NULL THEN
OUPUT “No hospitals meet the criteria.”
END
END IF

// Step 3: Retrieve doctors working at the hospital with the largest
size
QUERY "GetDoctorsAtLargestHospital”
SELECT doctor_id, first_name, second_name, hospital_id
FROM DOCTORS
WHERE hospital_id = largest_hospital_id

// Step 4: Output doctors
IF doctors exist THEN
           OUTPUT “List of doctors at the hospital with the largest
bed size:”
DISPLAY doctor first_name, second_name, hospital_id
ELSE
OUTPUT “No doctors found at the hospital with the largest bed size.”
END IF
END
```

18) **A list of all hospital names which were accredited prior to 2015 and do have Emergency Service facilities.**

```
BEGIN

// Step 1: Input criteria

INPUT accreditation_year (< 2015)

INPUT emergency_service (TRUE)
```

```
// Step 2: Query HOSPITALS table for accreditation before 2015
QUERY “GetHospitalsAccreditedBefore2015”
SELECT hosptial_id, hosptial_name, accreditation_year,
emergency_service
FROM HOSPITALS
WHERE accreditation_year < 2015
SET selected_hospitals = retrieved results

// Step 3: Check hospitals meet this criteria
IF selected_hospitals IS NULL THEN
OUTPUT “No hospitals meet the accreditation criteria.”
END
END IF

// Step 4: Query for hospitals with emergency services
     QUERY "GetHospitalsWithEmergencyService”
          SELECT hospital_name
          FROM selected_hospitals
          WHERE emergency_service = True
          SET emergency_hospitals = retrieved results

// Step 5: Check presence of emergency hospitals
IF emergency_hospitals IS NULL THEN
OUTPUT “No hospitals with emergency services meet accreditation
criteria.”
END
END IF
```

```
// Step 6: Output
OUTPUT "Hospitals accredited before 2015 with emergency services:"
DISPLAY hospital_name
END
```

19) **A list of patients registered with doctors who are based at hospital with <400 beds**

```
BEGIN
// Step 1: Query HOSPITALS with less than 400 beds
QUERY "GetHospitalsWithLessThan400Beds"
SELECT hospital_id, hospital_name, size
           FROM HOSPITALS
```

```
WHERE size < 400
SET selected_hospitals = retrieved results

// Step 2: Check hospitals meet criteria
IF selected_hospitals IS NULL THEN
OUTPUT “No hospitals meet the size criteria.”
END
END IF

// Step 3: Retrieve doctors working at hospital
QUERY “GetDrsAtSelectedHospitals”
SELECT doctor_id, first_name, second_name, hospital_id
FROM DOCTORS
WHERE hospital_id IN selected_hospitals
SET selected_doctors = retrieved results

// Step 4: Check doctors are associated with hospitals
IF selected_doctors IS NULL THEN
OUTPUT “No doctors found at hospitals with less than 400 beds.”
END
END IF

// Step 5: Retrieve patients registered with doctors
QUERY “GetPatientsOfSelectedDrs”
SELECT patient_id, first_name, second_name, doctor_id
FROM PATIENTS
WHERE doctor_id IN selected_doctors
SET selected_patients = retrieved results
```

```
// Step 6: Check patients registered with doctors

IF selected_patients IS NULL THEN

OUTPUT “No patients registered with doctors found at hospitals with less than 400 beds.”

END
      END IF


      // Step 7: Output

      OUTPUT “List of patients registered with doctors at hospitals with less than 400 beds:”

      DISPLAY patient first_name, second_name, doctor_id

END
```