

---

**Green Valleys Hospital**

---

### **Team Members**

- Bhuwan Bahadur Neupane (20224)
- Jubaida Tasnim (20175)
- Rajat Kamble (20230)
- Raman Deep Kaur (20215)

---

## **MODULE 1**

---

### **ABSTRACT**

Green Valleys Hospital in Edinburgh focuses on providing care for elderly patients. The hospital has 17 wards, each with unique details, and manages various supplies needed for patient care. Currently, the hospital uses a paper-based system to keep track of all this information, which can be slow and prone to mistakes. This project aims to design a database to make data management faster, more accurate, and easier. The database will cover important areas such as ward details, patient medication records, supply inventories, requisition forms, supplier information, and staff details. We will identify the necessary data fields, create input forms and routine reports, and design input and output screens for daily transactions, all based on user needs. Implementing this database will improve data handling, reduce the staff's administrative burden, and enhance patient care. The project will be developed in stages, with feedback from hospital staff to ensure it meets their needs. The goal is to provide a reliable and easy-to-use database system that supports the hospital's mission of delivering high-quality care.

## **Scenario**

Green Valleys Hospital has 17 wards with a total of 240 beds available for short- and long-term patients, and an outpatient clinic. The hospital records and maintains various data to support management and day-to-day operations. This includes details of wards, patient medication, surgical and nonsurgical supplies, pharmaceutical supplies, ward requisitions, and supplier information. The hospital also manages staff information, including qualifications and previous work experience.

## **Problem Statement**

The current system at Green Valleys Hospital is paper-based, which is inefficient and prone to errors. The hospital needs a computerized database system to streamline data entry, update, retrieval, and reporting processes. The new system should ensure accurate and timely information management, reduce the workload on hospital staff, and improve overall operational efficiency.

## **Approach**

The project will follow these steps to develop the database system:

1. Design the data dictionary, detailing all the fields and data types required for the database.
2. List initial assumptions to clarify the project scope and ensure a clear understanding of the requirements.
3. Create a cross-reference table to link different terms and data fields used in the database.
4. Design input documents and routine reports, ensuring they meet user needs without making assumptions about the database structure.

### **B.3.1 Data Requirements**

#### **Wards**

The *Green Valleys Hospital* has 17 wards with a total of 240 beds available for short- and long-term patients, and an outpatient clinic. Each ward is uniquely identified by a number (for example, ward 11) and also a ward name (for example, Orthopedic), location (for example, E Block), total number of beds, and telephone extension number (for example, Extn. 7711).

#### **Patient medication**

When a patient is prescribed medication, the details are recorded. This includes the patient's name and number, drug number and name, units per day, method of administration (for example, oral, intravenous (IV)), start and finish date. The medication (pharmaceutical supplies) given to each patient is monitored. An example of a *Green Valleys Hospital* report would be a form used to record the details of medication given to a patient .

#### **Surgical and nonsurgical supplies**

The *Green Valleys Hospital* maintains a central stock of surgical (for example, syringes and sterile dressings) and nonsurgical (for example, plastic bags and aprons) supplies. The details of surgical and nonsurgical supplies include the item number and name, item description, quantity in stock, reorder level, and cost per unit. The item number uniquely identifies each type of surgical or nonsurgical supply.

The supplies used by each ward are monitored.

#### **Pharmaceutical supplies**

The hospital also maintains a stock of pharmaceutical supplies (for example, antibiotics and painkillers). The details of pharmaceutical supplies include drug number and name, description, dosage, method of administration, quantity in stock, reorder level, and cost per unit. The drug number uniquely identifies each type of pharmaceutical supply. The pharmaceutical supplies used by each ward are monitored.

## **Ward requisitions**

When required, the Charge Nurse may obtain surgical, nonsurgical, and pharmaceutical supplies from the central stock of supplies held by the hospital. This is achieved by ordering supplies for the ward using a requisition form. The information detailed on a requisition form includes a unique requisition number, the name of the member of staff placing the requisition, and the number and name of the ward. Also included is the item or drug number, name, description, dosage and method of administration (for drugs only), cost per unit, quantity required, and date ordered. When the requisitioned supplies are delivered to the ward, the form must be signed and dated by the Charge Nurse who initiated the order. An example of a *Green Valleys Hospital* requisition form would be a form used to order supplies of morphine for ward 11 .

## **Suppliers**

The details of the suppliers of the surgical, nonsurgical, and pharmaceutical items are stored. This information includes the supplier's name and number, address, email, and telephone and fax numbers. The supplier number is unique to each supplier.

Research and identify the common operations eg. Data Entry ,Data update or deletion operations ,List the problems that needs to be addressed etc.

Examples could be :

Create and maintain records recording the details of members of staff (Personnel Officer).

Search for staff who have particular qualifications or previous work experience

(Personnel Officer).

Produce a report listing the details of staff allocated to each ward (Personnel

Officer and Charge Nurse).

## **Step 1.1: Format of Every Input Document**

### **1. Wards Information Form**

Ward Number: \_\_\_\_\_

Ward Name: \_\_\_\_\_

Location: \_\_\_\_\_

Total Number of Beds: \_\_\_\_\_

Telephone Extension: \_\_\_\_\_

Charge Nurse: \_\_\_\_\_

### **2. Patient Medication Form**

Patient Name: \_\_\_\_\_

Patient Number: \_\_\_\_\_

Drug Number: \_\_\_\_\_

Drug Name: \_\_\_\_\_

Units Per Day: \_\_\_\_\_

Method of Administration: \_\_\_\_\_

Start Date: \_\_\_\_\_

Finish Date: \_\_\_\_\_

Date Admitted: \_\_\_\_\_

Ward Assigned: \_\_\_\_\_

### **3. Surgical and Nonsurgical Supplies Form**

Item Number: \_\_\_\_\_

Item Name: \_\_\_\_\_

Types of Item[surgical/nonsurgical]:\_\_\_\_\_

Item Description: \_\_\_\_\_

Quantity in Stock: \_\_\_\_\_

Reorder Level: \_\_\_\_\_

Cost Per Unit: \_\_\_\_\_

Supplier Number\_\_\_\_\_

### **4. Pharmaceutical Supplies Form**

Drug Number: \_\_\_\_\_

Drug Name: \_\_\_\_\_

Description: \_\_\_\_\_

Dosage: \_\_\_\_\_

Method of Administration: \_\_\_\_\_

Quantity in Stock: \_\_\_\_\_

Reorder Level: \_\_\_\_\_

Cost Per Unit: \_\_\_\_\_

Supplier Number \_\_\_\_\_

## **5. Ward Requisition Form**

Requisition Number: \_\_\_\_\_

Staff Name: \_\_\_\_\_

Ward Number: \_\_\_\_\_

Ward Name: \_\_\_\_\_

Item or Drug Number: \_\_\_\_\_

Item or Drug Name: \_\_\_\_\_

Item Description: \_\_\_\_\_

Dosage: \_\_\_\_\_

Method of Administration: \_\_\_\_\_

Cost Per Unit: \_\_\_\_\_

Quantity Required: \_\_\_\_\_

Order Date: \_\_\_\_\_

Charge Nurse Signature: \_\_\_\_\_

Delivery Date: \_\_\_\_\_

## **6. Suppliers Information**

Supplier Name: \_\_\_\_\_

Supplier Number: \_\_\_\_\_

Address: \_\_\_\_\_

Email: \_\_\_\_\_

Telephone Number: \_\_\_\_\_

Fax Number: \_\_\_\_\_

## **7. Staff Information Form**

Staff ID: \_\_\_\_\_

First Name: \_\_\_\_\_

Last Name: \_\_\_\_\_

Position: \_\_\_\_\_

Ward Number Assigned: \_\_\_\_\_

Date of Hire: \_\_\_\_\_

Phone Number: \_\_\_\_\_

Email Address: \_\_\_\_\_

Address: \_\_\_\_\_

## **8. Staff Qualifications**

Staff ID: \_\_\_\_\_

Qualification Name: \_\_\_\_\_

Institution: \_\_\_\_\_

Year Obtained: \_\_\_\_\_

#### **9. Staff Experience**

Staff ID: \_\_\_\_\_

Job Title: \_\_\_\_\_

Organization: \_\_\_\_\_

Start Date: \_\_\_\_\_

End Date: \_\_\_\_\_

#### **Step 1.2: Format of Every Routine Report**

## **1. Ward Details Report**

Report Title: Ward Details Report

Report Date: yyyy/mm/dd

Ward Number	Ward Name	Location	Total Number of Beds	Telephone Extension
xxx	xxx	xxx	xxx	xxx

## **2. Patient Medication Report**

Report Title: Patient Medication Report

Report Date: yyyy/mm/dd

Patient Number	Patient Name	Drug Number	Drug Name	Units Per Day	Method of Administration	Start Date	Finish Date
xxx	xxx	xxx	xxx	[Amount]	xxx	yyyy/mm/dd	yyyy/mm/dd

## **3. Surgical and Nonsurgical Supplies Inventory Report**

Report Title: Surgical and Nonsurgical Supplies Inventory Report

Report Date: yyyy/mm/dd

Item Number	Item Name	Item Description	Quantity in Stock	Reorder Level	Cost Per Unit
xxx	xxx	xxx	[Number]	[Number]	[Amount]

## **4. Pharmaceutical Supplies Inventory Report**

Report Title: Pharmaceutical Supplies Inventory Report

Report Date: yyyy/mm/dd

Drug Number	Drug Name	Description	Quantity in Stock	Reorder Level	Cost Per Unit
xxx	xxx	xxx	[Number]	[Number]	[Amount]

## 5. Suppliers Detail Report

Report Title: Suppliers Detail Report

Report Date: yyyy/mm/dd

SupplierNumber	SupplierName	Address	Email	Telephone Number	Fax Number
xxx	xxx	xxx	xxx	xxx	xxx

## 6. Staff Allocated to Ward Report

Report Title: Ward Allocation Report

Report Date: yyyy/mm/dd

StaffID	First Name	Last Name	Position	Ward Number	Phone Number	Email
xxx	xxx	xxx	xxx	xxx	xxx	xxx

## 7. Staff Qualifications Report

Report Title: Staff Qualifications Report

Report Date: yyyy/mm/dd

StaffID	First Name	Last Name	Qualification Name	Institution	Year Obtained
xxx	xxx	xxx	xxx	xxx	[yyyy/mm/dd]

## **8. Staff Experience Report -**

Report Title: Staff Experience Report

Report Date: yyyy/mm/dd

StaffID	First Name	Last Name	Job Title	Organization	Start Date	End Date
xxx	xxx	xxx	xxx	xxx	[yyyy/mm/dd]	[yyyy/mm/dd]

## **9. Active Patient Summary Report**

Report Title: Active Patient Report

Report Date: yyyy/mm.dd

Patient Name	Patient Number	Ward Number	Date Admitted
xxx	xxx	xxx	[yyyy/mm/dd]

## **10. Ward Supply Usage Report**

WardNumber	ItemNumber or DrugNumber	OrderDate	Quantity Ordered	DeliveryDate
xxx	xxx	xxx	[Amount]	[yyyy/mm/dd]

### Step 1.3 Describe or sketch the input and output screens for routine transactions.

#### 1. Patient Medication Entry

##### Input Screen:

- Title: Patient Medication Entry
- Fields:
  - Patient Number: [Text Input]
  - Patient Name: [Text Input]
  - Drug Number: [Dropdown]
  - Drug Name: [Text Input]
  - Units per Day: [Number Input]
  - Method of Administration: [Dropdown (Oral, IV, etc.)]
  - Start Date: [Date Picker]
  - Finish Date: [Date Picker]
- **Button:** Submit

##### Output Screen:

- Title: Patient Medication Form with the below details is submitted
  - Patient Number
  - Patient Name
  - Drug Number
  - Drug Name
  - Units per Day
  - Method of Administration

- Start Date
- Finish Date

## 2. Ward Requisition Entry

### Input Screen:

- Title: Ward Requisition Form
- Fields:
  - Requisition Number: [Text Input]
  - Staff Name: [Text Input]
  - Ward Number: [Dropdown]
  - Ward Name: [Dropdown]
  - Item/Drug Number: [Dropdown]
  - Item/Drug Name: [Dropdown]
  - Item Description: [Text Input]
  - Dosage: [Text Input]
  - Method of Administration: [Dropdown (if applicable)]
  - Cost per Unit: [Number Input]
  - Quantity Required: [Number Input]
  - Date Ordered: [Date Picker]
- **Button:** Submit

### Output Screen:

- Title: Requisition Form is submitted with below details
  - Requisition Number
  - Staff Name
  - Ward Number
  - Ward Name
  - Item/Drug Number
  - Item/Drug Name

- Item Description
- Dosage
- Method of Administration
- Cost per Unit
- Quantity Required
- Date Ordered

### **3. Supplier Details Entry**

#### Input Screen:

- Title: Supplier Details Entry
- Fields:
  - Supplier Number: [Text Input]
  - Supplier Name: [Text Input]
  - Address: [Text Input]
  - Email: [email Input]
  - Telephone Number: [Text Input]
  - Fax Number: [Text Input]
- **Button:** Submit

#### Output Screen:

- Title: Supplier Record is submitted with below details:
  - Supplier Number
  - Supplier Name
  - Address
  - Email
  - Telephone Number
  - Fax Number

#### **4. Ward Details Inquiry**

##### Input Screen:

- Title: Ward Details Inquiry
- Fields:
  - Ward Number: [Dropdown]
- **Button:** Search

##### Output Screen:

- Title: Ward Details
- Fields Displayed:
  - Ward Number
  - Ward Name
  - Location
  - Total Number of Beds
  - Telephone Extension Number

#### **5. Medication Report for a Patient**

##### Input Screen:

- Title: Medication Report Inquiry
- Fields:
  - Patient Number: [Dropdown]
- **Button:** Search

##### Output Screen:

- Title: Medication Report for Patient [Patient Name]
- Fields Displayed:
  - Patient Number
  - Patient Name
  - Drug Number

- Drug Name
- Units per Day
- Method of Administration
- Start Date
- Finish Date

## 6. Search staff with particular qualifications

### Input Screen:

- Title: Qualification Report for Staff [Qualification Name]
- Fields:
  - Qualification Name: [Text Input]
- **Button:** Search

### Output Screen:

- Title: Qualification Report for Staff [Qualification Name]
- Fields Displayed:
  - StaffID
  - First Name
  - Last Name
  - Institution
  - Year Obtained

## 7. Search supply details by item number

### Input Screen:

- Title: Supply details for item [Item Number]
- Fields:
  - Item Number: [Text Input]
- **Button:** Search

### Output Screen:

- Title: Supply details for item [Item Number]
- Fields Displayed:
  - Item Name
  - Reorder Level
  - Quantity in Stock
  - Cost Per Unit

### **Step 1.4 Initial list of assumptions for the project**

1. Each patient, staff member, and supplier has a unique identifier.
2. Each Ward has a unique ward number
3. Each Supply item (Surgical/NonSurgical/Pharmaceutical) has a unique item/drug number
4. Each requisition form has a unique requisition number
5. All input forms will be digital and accessible through a user-friendly interface.
6. The database will handle current and future data requirements.
7. Reports will be generated automatically based on the data entered through the input forms.

### **Common Operations**

#### **1. Data Entry:**

- Entering details of new wards.
- Recording patient medication information.
- Adding new surgical and nonsurgical supplies.
- Logging pharmaceutical supplies.
- Filling out and submitting requisition forms.
- Entering supplier information.
- Entering Staff Information

#### **2. Data Update:**

- Updating ward details (e.g., change in number of beds, location).
- Modifying patient medication records (e.g., changes in dosage or administration method).
- Adjusting stock levels for surgical, nonsurgical, and pharmaceutical supplies.

- Editing supplier information.
- Editing Staff Information (change or contact, ward reassignment)

**3. Data Deletion:**

- Removing outdated or erroneous records of wards.
- Deleting patient medication records after discharge.
- Discarding obsolete or expired supply items from the inventory.
- Archiving old requisition forms.
- Removing supplier records when no longer in use.

**4. Search and Retrieval:**

- Searching for specific wards by ID, name, or location.
- Looking up patient medication history by patient name or number.
- Retrieving supply details by item number or name.
- Finding specific requisitions by requisition number or staff name.
- Searching for supplier details by supplier number or name.
- Searching staff details allocated to a particular ward
- Retrieving supply details that need to be reordered
- Searching for staff by qualification or previous experience.
- 

---

**MODULE 2**

---

## Step 2.1 Designing the Data Dictionary

Term	Data Type	Description
WardNumber	int	An integer value that uniquely identifies each ward in the hospital
WardName	varchar	A string representing the name of the ward, indicating the type of care provided
Location	varchar	A string that specifies the physical location of the ward within hospital
NumberOfBeds	int	An integer that indicates the total number of beds available in the ward
telephone extension	varchar	A string that provides the telephone extension number for the ward
ChargeNurse	int	StaffID of ChargeNurse of a specific ward
PatientName	varchar	A string representing the name of the patient visited/admitted to the hospital
PatientNumber	int	An integer value that uniquely identifies each patient
DrugName	varchar	A string indicating the name of the drug prescribed to a patient
DrugNumber	int	An integer value that uniquely identifies each drug
UnitsPerDay	int	An integer that specifies the number of units of medication administered to the patient per day
AdministrationMethod	varchar	A string describing the method of administering the medication
StartDate	date	A date value indicating the start date for the medication administration
FinishDate	date	A date value indicating the end date for the medication administration
DateAdmitted	date	Date patient is admitted to hospital (if applicable)
WardAssignedToPatient	int	WardNumber assigned to admitted patient (if applicable)
ItemNumber	int	An integer that uniquely identifies each type of supply, whether surgical or nonsurgical
ItemName	varchar	A string representing the name of the supply item

ItemDescription	varchar	A string describing the supply item
ItemType	varchar	A string representing the type of item - surgical/nonsurgical
QuantityInStock	int	An integer that indicates the current quantity(units) of the supply item in stock
ReorderLevel	int	An integer specifying the quantity level at which new stock should be ordered
CostPerUnit	decimal	A decimal value that indicates the cost per unit of the supply item
Dosage	varchar	Amount of a drug that is to be administered at one time or over a specified period, according to medical guidelines.
requisition number	int	An integer that uniquely identifies each requisition form
StaffName	varchar	A string representing the name of the staff member placing the requisition
OrderDate	date	The date of the order
ChargeNurseSignature	varchar	A string indicating the Nurse's Signature
QuantityRequired	int	Count of units of a supply item being requested
DeliveryDate	date	The date of the order delivery
SupplierName	varchar	A string indicating the name of the supplier providing the hospital with supplies
SupplierNumber	int	An integer value that uniquely identifies each supplier
SupplierAddress	varchar	A string providing the address of the supplier
SupplierEmail	varchar	A string representing the email address of the supplier
SupplierPhone	varchar	A string indicating the phone number of the supplier
SupplierFax	varchar	A string representing the fax number of the supplier
StaffID	int	Unique identifier for each staff member
FirstName	varchar	First name of the staff member
LastName	varchar	Last name of the staff member
Position	varchar	Job position of the staff member

WardNumberAssigned	int	The ward number assigned to the staff member
DateOfHire	date	The date when the staff member was hired
StaffPhoneNumber	varchar	The contact number of the staff member
StaffEmailAddress	varchar	Email address of the staff member
Address	varchar	Home address of staff member
QualificationName	varchar	Name of the qualification
Institution	varchar	Name of the institution where the qualification was obtained
YearObtained	int	The year when the qualification was obtained
JobTitle	varchar	Staff previous experience - Title of the job
Organization	varchar	Staff previous experience - Name of the organization where the job was held
StartDateJob	date	Start date of the job
EndDateJob	date	End date of the job

## Step 2.2: List of Assumptions

1. Each ward is assigned a unique WardNumber.
  2. Each patient is assigned a unique PatientNumber upon visit/admission.
  3. Each drug has a unique DrugNumber.
  4. Each type of supply (surgical or nonsurgical) has a unique ItemNumber.
  5. Requisition forms have unique RequisitionNumbers.
  6. Suppliers have unique SupplierNumbers.
  7. Staff has unique StaffID

### **Step 2.3: Cross-Reference Table**

Term	Ward Inform	Patient Medicati	Surgical and	Pharmac	utical	Ward Requisitio	Suppliers Informatio	Staff Informati	Staff Qualificat	Staff Experi
------	-------------	------------------	--------------	---------	--------	-----------------	----------------------	-----------------	------------------	--------------

	ation	on	Nonsurgic al Supplies	Supplies	n	n	on	ion	ence
Ward Number	✓				✓				
WardName	✓				✓				
Location	✓								
NumberOfBe ds	✓								
TelephoneEx tension	✓								
ChargeNurse	✓								
PatientName		✓							
PatientNum ber		✓							
DrugNumber		✓		✓	✓				
DrugName		✓		✓	✓				
UnitsPerDay		✓							
MethodOfAd ministration		✓		✓	✓				
StartDate		✓							
FinishDate		✓							
DateAdmitte d		✓							
WardAssigne dToPatient		✓							
TypeOfItem			✓						
ItemNumber			✓		✓				
ItemName			✓		✓				
ItemDescript ion			✓		✓				



Qualification Name								✓	
Institution								✓	
YearObtained								✓	
JobTitle									✓
Organization									✓
StartDateJob									✓
EndDateJob									✓

---

## MODULE 3

---

## **Step 3.1: Make a list of all entities and their associated attributes**

### **Entities and Attributes**

1. **Ward**
  - a. WardNumber (int)
  - b. WardName (varchar)
  - c. Location (varchar)
  - d. NumberOfBeds (int)
  - e. TelephoneExtension (varchar)
  - f. ChargeNurse (int)
2. **Patient**
  - a. PatientName (varchar)
  - b. PatientNumber (int)
  - c. DateAdmitted (date)
  - d. WardAssignedToPatient (int)
3. **Medication**
  - a. DrugName (varchar)
  - b. DrugNumber (int)
  - c. UnitsPerDay (int)
  - d. AdministrationMethod (varchar)
  - e. StartDate (date)
  - f. FinishDate (date)
  - g. PatientNumber
4. **Supply**
  - a. ItemNumber (int)
  - b. ItemName (varchar)
  - c. ItemDescription (varchar)
  - d. ItemType (varchar)
  - e. QuantityInStock (int)
  - f. ReorderLevel (int)
  - g. CostPerUnit (decimal)
  - h. Dosage (varchar) (applicable fro pharmaceutical supplies)
  - i. SupplierNumber (new addition)
5. **Requisition**
  - a. RequisitionNumber (int)
  - b. OrderDate (date)
  - c. QuantityRequired (int)
  - d. DeliveryDate (date)
  - e. StaffID
  - f. StaffName
  - g. ItemNumber
  - h. WardNumber
6. **Supplier**

- a. SupplierName (varchar)
- b. SupplierNumber (int)
- c. SupplierAddress (varchar)
- d. SupplierEmail (varchar)
- e. SupplierPhone (varchar)
- f. SupplierFax (varchar)

7. **StaffInformation**

- a. StaffID (int)
- b. FirstName (varchar)
- c. LastName (varchar)
- d. Position (varchar)
- e. WardNumberAssigned (int)
- f. DateOfHire (date)
- g. StaffPhoneNumber (varchar)
- h. StaffEmailAddress (varchar)
- i. Address (varchar)

8. **StaffQualification**

- a. QualificationName (varchar)
- b. Institution (varchar)
- c. YearObtained (int)
- d. StaffID

9. **StaffExperience**

- a. JobTitle (varchar)
- b. Organization (varchar)
- c. StartDateJob (date)
- d. EndDateJob (date)
- e. StaffID

**Step 3.2: List of relationships to be represented:**

**1. Ward and Patient**

- Relationship: A patient is admitted to a ward
- Cardinality: One to Many: Ward to Patient
  - A single ward can have multiple patients assigned to it

- Each patient is assigned to one ward at a time.
- Descriptive Attributes:
  - DateAdmitted (date) : Records the date the patient was admitted to the ward.

## 2. Patient and Medication

- Relationship: Patient is prescribed a medication
- Cardinality: One to Many
  - A patient can have multiple medications prescribed,
  - Each medication is prescribed to one patient
- Descriptive Attributes:
  - UnitsPerDay (int): Number of units of the medication the patient is prescribed daily.
  - AdministrationMethod (varchar): The method of administering the medication (e.g., oral, IV).
  - StartDate (date): The start date for the medication.
  - FinishDate (date): The end date for the medication.

## 3. Ward and Staff

- Relationship: Ward is assigned to staff
- Cardinality: Many to One (Staff to Ward)
  - Each staff member will be assigned to only one ward
  - Each ward can have many staff members assigned
- Descriptive Attributes: N/A

## 4. Supply and Supplier

- Relationship: Supply is provided by supplier
- Cardinality: One to Many: Supplier to Supply
  - Each supplier can supply many items
  - Each supply item is supplied by only one supplier
- Descriptive Attributes: N/A

## 5. Supply and Requisition

- Relationship: Supplies are requested through requisition form
- Cardinality: Many to Many
  - Each requisition form can request many supply items.
  - Each supply item can be part of many requisition form

- Descriptive Attributes:
  - QuantityRequired (int): Describes how much of the supply item is being requested in a particular requisition.
  - OrderDate (date): Indicates the date when the requisition was placed.
  - DeliveryDate (date): Indicates the date of delivery for the requisitioned supplies.

## **6. Requisition and Ward**

- Relationship: Requisitions are made for specific wards.
- Cardinality: Many to One - Requisition to Ward
  - Each requisition is made for one ward
  - Each ward can have many requisition forms
- Descriptive Attributes: N/A

## **7. Staff and Requisition**

- Relationship: StaffMember (charge nurse) makes requisition
- Cardinality: One to Many (Staff to Requisition)
  - One Staff (charge nurse) can make many Requisitions.
  - Each Requisition are associated with one Staff (charge nurse).
- Descriptive Attributes: N/A

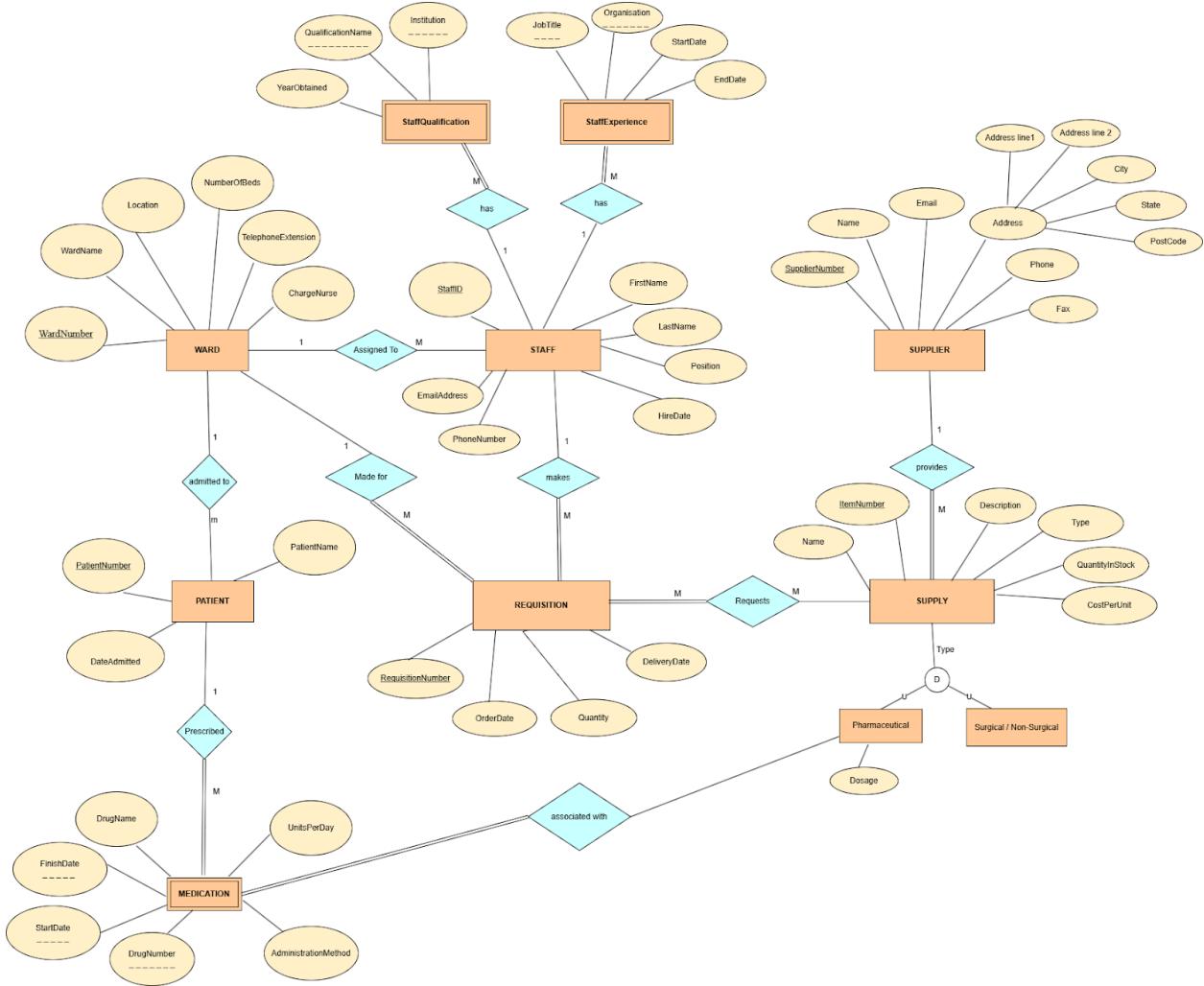
## **8. Staff and StaffQualification**

- Relationship: Staff has qualifications
- Cardinality: One to Many (Staff to Qualification)
  - A staff member can have multiple qualifications
  - Each qualification belongs to one staff member.
- Descriptive Attributes:
  - QualificationName (varchar): Name of the qualification.
  - Institution (varchar): Name of the institution where the qualification was obtained.
  - YearObtained (int): Year when the qualification was obtained.

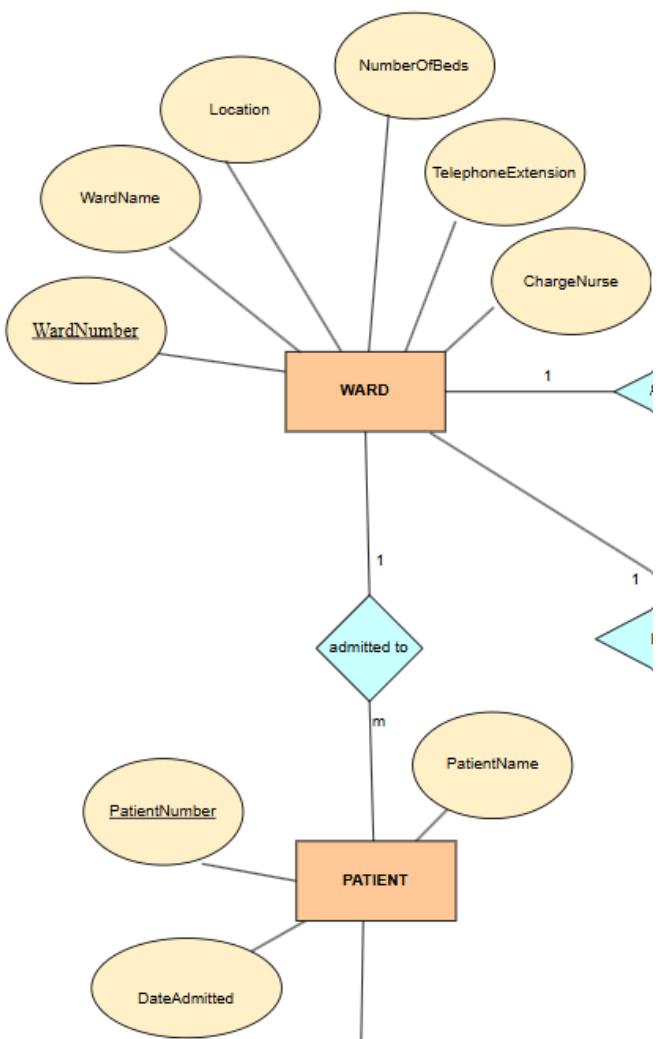
## **9. Staff and StaffExperience**

- Relationship: Staff has experience
- Cardinality: One to Many: Staff to StaffExperience
  - A staff member can have multiple job experiences
  - Each experience is related to one staff member.
- Descriptive Attributes:
  - JobTitle (varchar): The title of the previous job.
  - Organization (varchar): The organization where the job was held.
  - StartDateJob (date): Start date of the job.
  - EndDateJob (date): End date of the job.

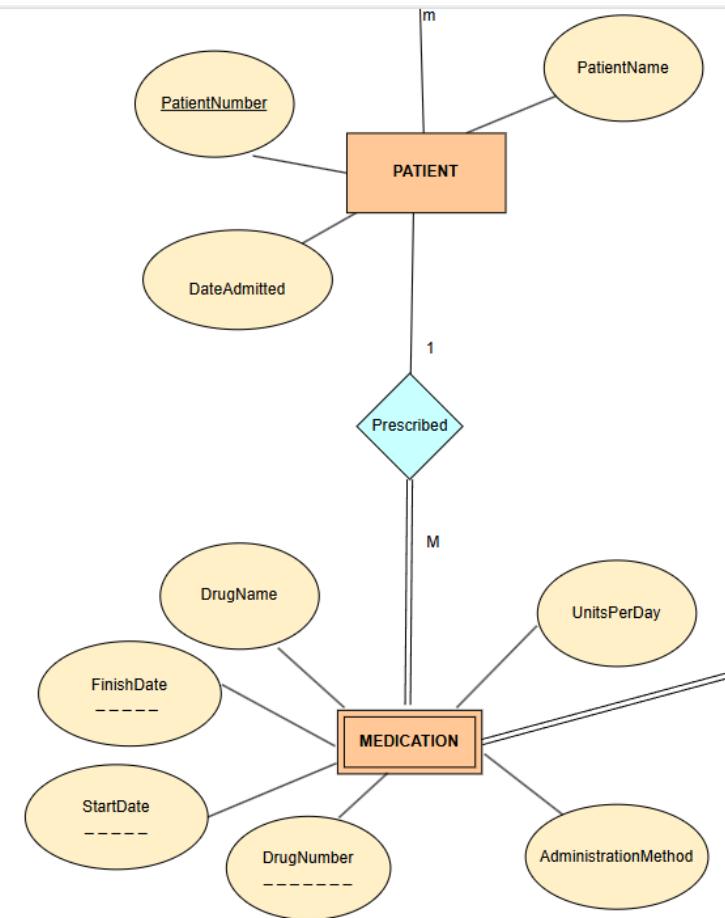
### **3.3 E-r Diagram:**



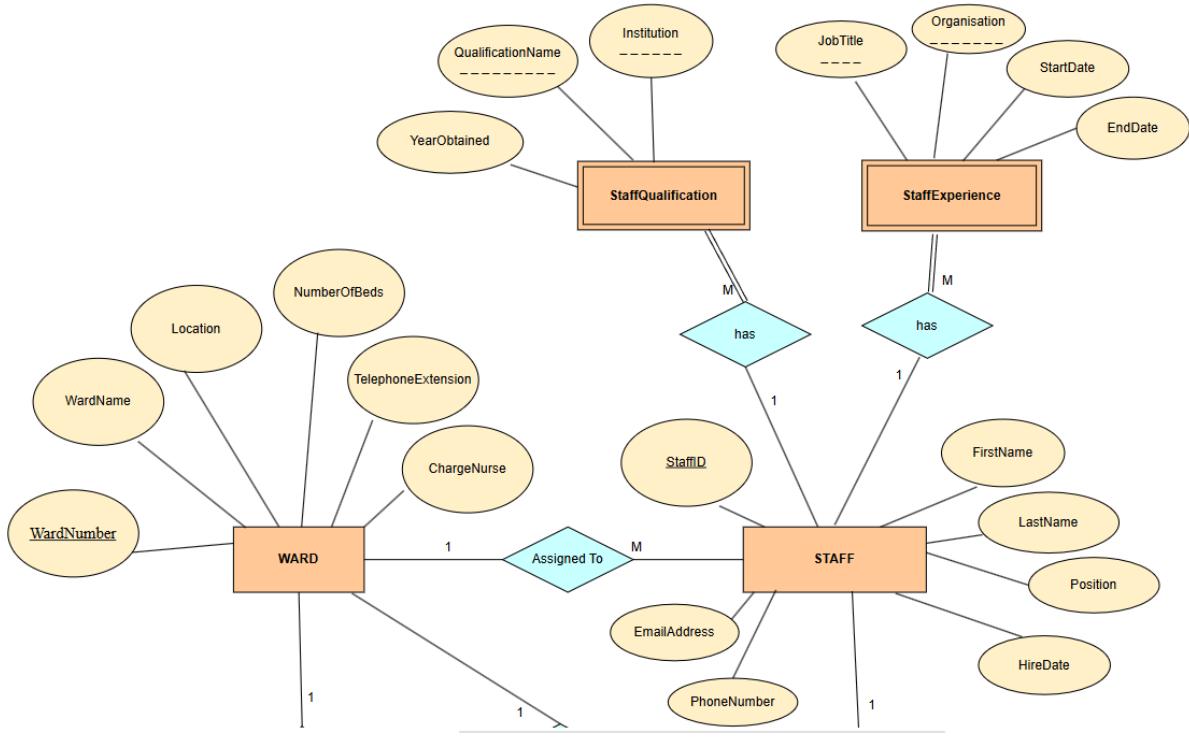
## Ward and Patient



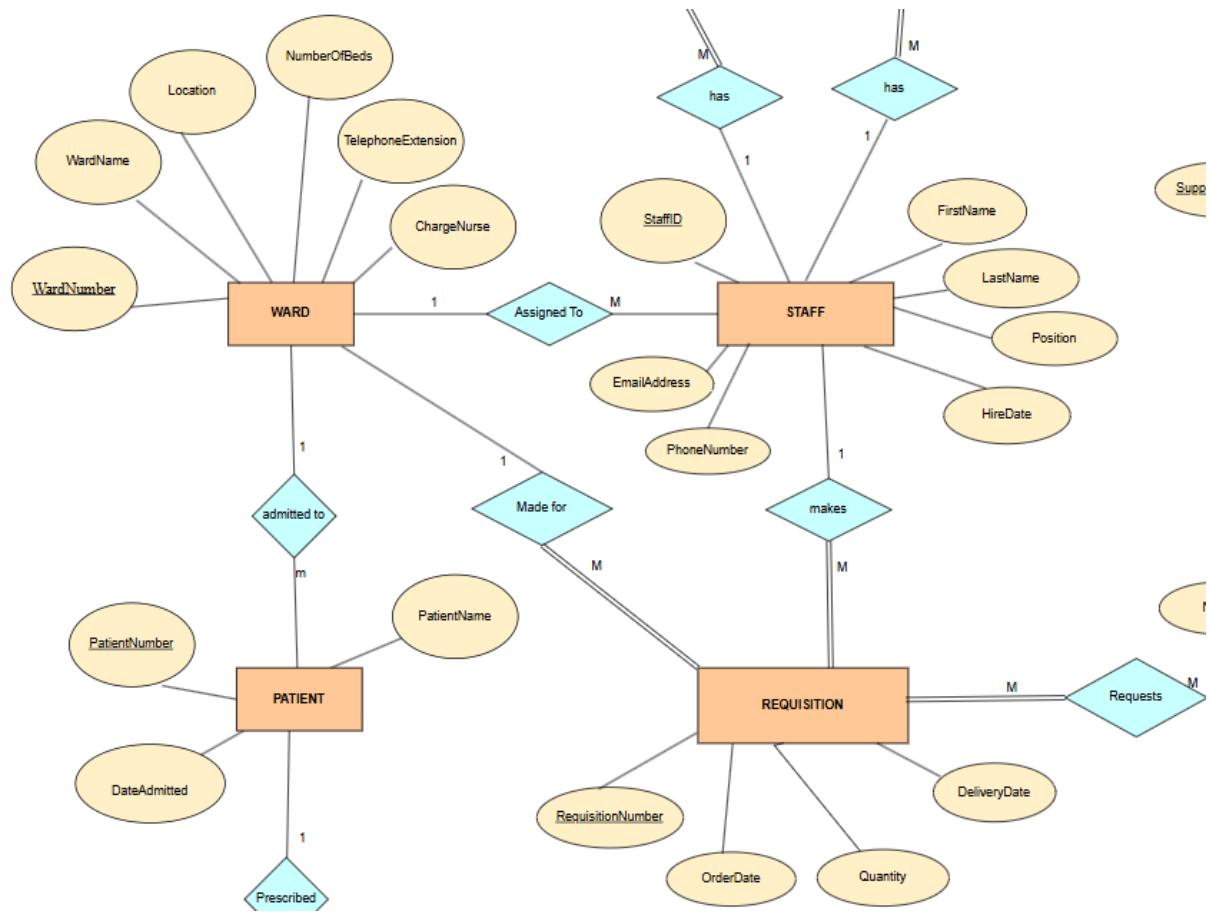
## Patient and Medication



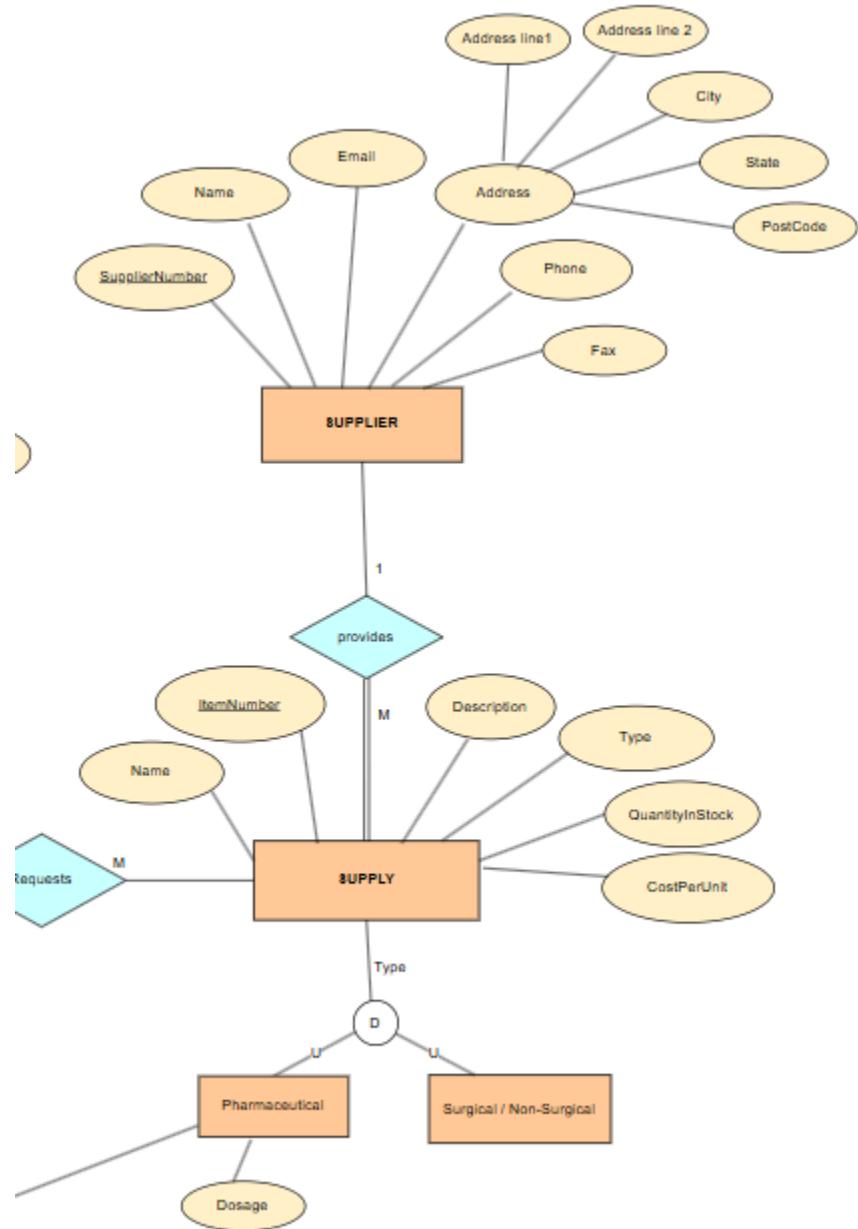
## Staff And Ward



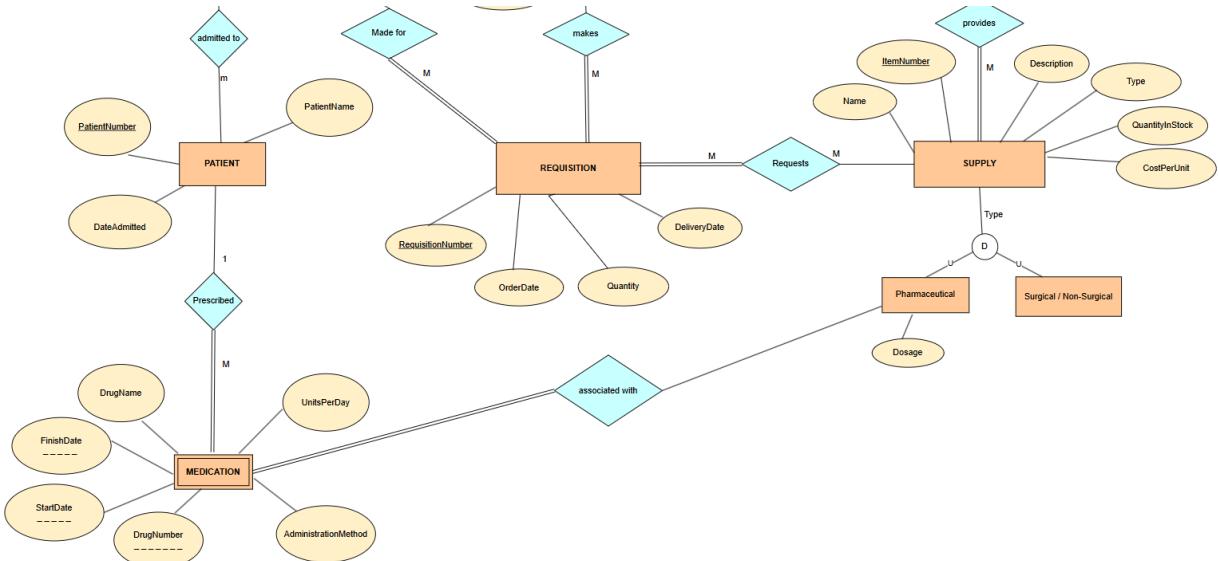
**Patient, Ward, Staff and Requisition**



## Supplier , Supply



Supply, medication



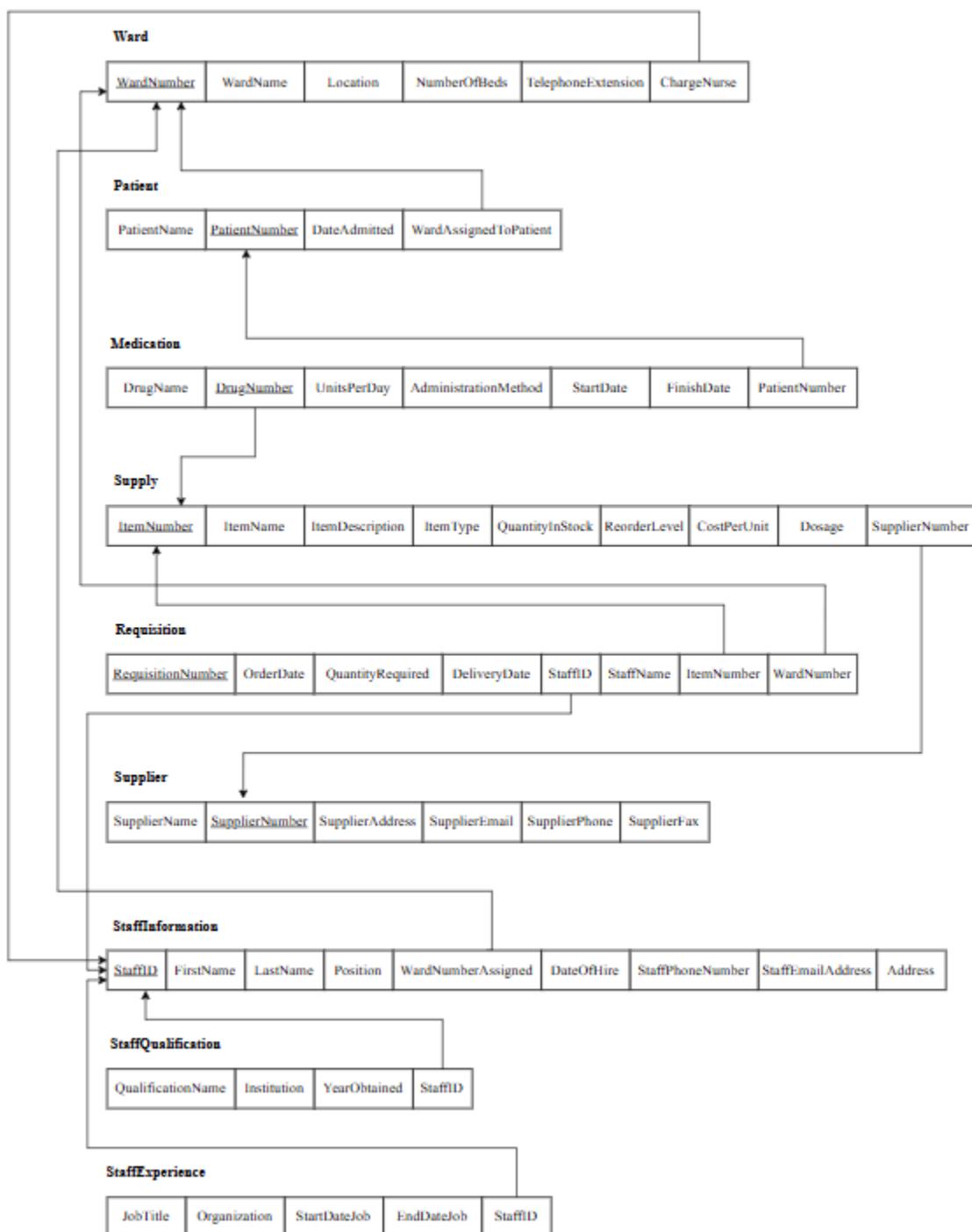
### 3.4 Updation of data dictionary:

Term	Data Type	Description
WardNumber	int	An integer value that uniquely identifies each ward in the hospital
WardName	varchar	A string representing the name of the ward, indicating the type of care provided
Location	varchar	A string that specifies the physical location of the ward within the hospital
NumberOfBeds	int	An integer that indicates the total number of beds available in the ward
TelephoneExtension	varchar	A string that provides the telephone extension number for the ward
ChargeNurse	int	StaffID of ChargeNurse of a specific ward
PatientName	varchar	A string representing the name of the patient vistied/admitted to the hospital
PatientNumber	int	An integer value that uniquely identifies each patient
DateAdmitted	date	Date patient is admitted to hospital (if applicable)
WardAssignedToPatient	int	WardNumber assigned to admitted patient (if applicable)
DrugName	varchar	A string indicating the name of the drug prescribed to a patient
DrugNumber	int	An integer value that uniquely identifies each drug
UnitsPerDay	int	An integer that specifies the number of units of medication administered to the patient per day
AdministrationMethod	varchar	A string describing the method of administering the medication
StartDate	date	A date value indicating the start date for the medication administration
FinishDate	date	A date value indicating the end date for the medication administration
ItemNumber	int	An integer that uniquely identifies each type of supply, whether surgical or nonsurgical
ItemName	varchar	A string representing the name of the supply item
ItemDescription	varchar	A string providing a description of the supply item
ItemType	varchar	A string representing type of item - surgical/non surgical

QuantityInStock	int	An integer that indicates the current quantity(units) of the supply item in stock
ReorderLevel	int	An integer specifying the quantity level at which new stock should be ordered
CostPerUnit	decimal	A decimal value that indicates the cost per unit of the supply item
Dosage	varchar	Amount of a drug that is to be administered at one time or over a specified period, according to medical guidelines.
RequisitionNumber	int	An integer that uniquely identifies each requisition form
OrderDate	date	The date of the order
QuantityRequired	int	Count of units of a supply item being requested
DeliveryDate	date	The date of the order delivery
StaffID	int	Unique identifier for each staff member
StaffName	varchar	A string representing the name of the staff member placing the requisition
SupplierName	varchar	A string indicating the name of the supplier providing the hospital with supplies
SupplierNumber	int	An integer value that uniquely identifies each supplier
SupplierAddress	varchar	A string providing the address of the supplier
SupplierEmail	varchar	A string representing the email address of the supplier
SupplierPhone	varchar	A string indicating the phone number of the supplier
SupplierFax	varchar	A string representing the fax number of the supplier
StaffID	int	Unique identifier for each staff member
FirstName	varchar	First name of the staff member
LastName	varchar	Last name of the staff member
Position	varchar	Job position of the staff member
WardNumberAssigned	int	Ward number assigned to the staff member
DateOfHire	date	The date when the staff member was hired
StaffPhoneNumber	varchar	Contact number of the staff member
StaffEmailAddress	varchar	Email address of the staff member

Address	varchar	Home address of staff member
QualificationName	varchar	Name of the qualification
Institution	varchar	Name of the institution where the qualification was obtained
YearObtained	int	Year when the qualification was obtained
JobTitle	varchar	Staff previous experience - Title of the job
Organization	varchar	Staff previous experience - Name of the organization where the job was held
StartDateJob	date	Start date of the job
EndDateJob	date	End date of the job

### 3.5 Relational Schema for Database:



## MODULE 4

---

**Step 4.1** Update the data dictionary and list of assumptions as needed. For each table, write the table name and write out the names, data types, and sizes of all the data items, and identify any constraints, using the conventions of the DBMS you will use for implementation

Term	Data Type	Size	Constraints
<b>Ward</b>			
WardNumber	INT		Primary Key, Not-null, Unique
WardName	VARCHAR	200	
Location	VARCHAR	200	
NumberOfBeds	INT		
TelephoneExtension	VARCHAR	20	
ChargeNurse	INT		Foreign Key- References Staff(StaffID), Nullable (can be NULL if not assigned)
<b>Supplier</b>			
SupplierNumber	INT		Primary Key, Not-null, Unique
Name	VARCHAR	50	
Address	VARCHAR	100	
City	VARCHAR	50	
State	VARCHAR	50	
PostCode	VARCHAR	20	
Email	VARCHAR	50	
Phone	VARCHAR	20	
Fax	VARCHAR	20	
<b>Staff</b>			
StaffID	INT		Primary Key, Not-null, Unique
FirstName	VARCHAR	50	

LastName	VARCHAR	50	
Position	VARCHAR	50	
WardNumber	INT		Foreign Key- References Ward(WardNumber), Nullable (can be NULL if not assigned to a ward).
HireDate	DATE		
PhoneNumber	VARCHAR	20	
Email	VARCHAR	50	
Address	VARCHAR	100	
City	VARCHAR	50	
State	VARCHAR	50	
PostCode	VARCHAR	20	
<b>StaffQualification</b>			
QualificationName	VARCHAR	50	Primary Key (part of the composite key), Not-null
Institution	VARCHAR	100	
YearObtained	INT		Primary Key (part of composite key), Not-null
StaffID	INT		Foreign Key - References Staff(StaffID), Not-null Primary Key (part of the composite key), Not-null
<b>StaffExperience</b>			
JobTitle	VARCHAR	50	Primary Key (part of composite key), Not-null
Organization	VARCHAR	100	Primary Key (part of composite key), Not-null
StartDate	DATE		
EndDate	DATE		
StaffID	INT		Primary Key (part of composite key) Foreign Key- References Staff(StaffID), Not-null
<b>Patient</b>			
PatientNumber	INT		Primary Key, Not-null, Unique
Name	VARCHAR	50	Not-null

Phone	VARCHAR	20	
Email	VARCHAR	100	
<b>Supply</b>			
ItemNumber	INT		Primary Key, Not-null, Unique
Name	VARCHAR	100	
Description	VARCHAR	150	
Type	VARCHAR	50	
QuantityInStock	INT		
ReorderLevel	INT		
CostPerUnit	DECIMAL(10, 2)		
Dosage	VARCHAR	50	Nullable
SupplierNumber	INT		Foreign Key- References Supplier(SupplierNumber), Not-null
<b>Requisition</b>			
RequisitionNumber	INT		Primary Key, Not-null, Unique
StaffID	INT		Foreign Key- References Staff(StaffID), Not-null
WardNumber	INT		Foreign Key- References Ward(WardNumber), Not-null
<b>RequisitionDetails</b>			
RequisitionNumber	INT		Foreign Key- References Requisition(RequisitionNumber), Non-null. Primary Key (part of composite key)
ItemNumber	INT		Foreign Key- References Supply(ItemNumber), Non-null. Primary Key (part of composite key)
Quantity	INT		
DeliveryDate	Date		
OrderDate	DATE		

<b>Admission</b>			
AdmissionID	INT		Primary Key, Not-null, Unique
PatientNumber	INT		Foreign Key references Patient(PatientNumber)
WardNumber	INT		Foreign Key references Ward(WardNumber)
DateAdmitted	DATE		Not-null
DateDischarged	DATE		nullable
<b>Medication</b>			
PatientNumber	INT		Foreign Key- References Patient(PatientNumber), Not-null.  Part of the composite primary key
DrugNumber	INT		Foreign Key- References Supply(LineNumber), Not-null  Part of the composite primary key
UnitsPerDay	INT		Not-null
AdministrationMethod	VARCHAR	50	Not-null
StartDate	DATE		Not-null  Part of the composite primary key
FinishDate	DATE		Not-null  Part of the composite primary key

### Assumptions

1. Each ward is assigned a unique WardNumber.
2. Each patient is assigned a unique PatientNumber upon visit/admission.
3. Each drug has a unique DrugNumber.
4. Each type of supply (surgical or nonsurgical) has a unique ItemNumber.
5. Requisition forms have unique RequisitionNumbers.
6. Suppliers have unique SupplierNumbers.
7. Staff has unique StaffID

8. StaffQualification , there is possibility of the same qualification being obtained multiple times
9. Staff Experience - A staff can have same job title over different organisation.
10. Each staff is only assigned to one ward at a time
11. A specific supply item is only provided by a particular supplier.
12. Patient can be prescribed same drug but over different time period.
13. Supply item requested through requisition can be delivered on different dates.

**Step 4.2 Write and execute SQL statements to create all tables needed to implement the design**

**– Creating database**

```
CREATE DATABASE HospitalManagement;
USE HospitalManagement;
```

**– Creating tables**

**-- Ward table**

```
CREATE TABLE Ward (
    WardNumber INT PRIMARY KEY,
    WardName VARCHAR(200),
    Location VARCHAR(200),
    NumberOfBeds INT,
    TelephoneExtension VARCHAR(20),
    ChargeNurse INT
);
```

**-- Supplier table**

```
CREATE TABLE Supplier (
    SupplierNumber INT PRIMARY KEY,
    Name VARCHAR(50),
    Address VARCHAR(100),
    City VARCHAR(50),
    State VARCHAR(50),
    PostCode VARCHAR(20),
    Email VARCHAR(50),
    Phone VARCHAR(20),
    Fax VARCHAR(20)
);
```

**-- Staff table**

```
CREATE TABLE Staff (
    StaffID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Position VARCHAR(50),
    WardNumber INT,
    HireDate DATE,
    PhoneNumber VARCHAR(20),
    Email VARCHAR(50),
    Address VARCHAR(100),
    City VARCHAR(50),
    State VARCHAR(50),
    PostCode VARCHAR(20),
    FOREIGN KEY (WardNumber) REFERENCES Ward(WardNumber)
);
```

**-- StaffQualification table**

```
CREATE TABLE StaffQualification (
    Staff_id int,
    QualificationName VARCHAR(50),
    Institution VARCHAR(100),
    YearObtained INT,
    PRIMARY KEY (StaffID, QualificationName, YearObtained),
    FOREIGN KEY (StaffID) REFERENCES Staff(StaffID)
);
```

**-- StaffExperience table**

```
CREATE TABLE StaffExperience (
    staff_id INT,
    JobTitle VARCHAR(50),
    Organization VARCHAR(100),
    StartDate DATE,
    EndDate DATE,
    PRIMARY KEY (StaffID, JobTitle, Organization),
    FOREIGN KEY (StaffID) REFERENCES Staff(StaffID)
);
```

**-- Patient table**

```
CREATE TABLE Patient (
    PatientNumber INT PRIMARY KEY,
    Name VARCHAR(50) NOT NULL,
    Phone VARCHAR(20),
```

```
Email varchar(100)
);

-- Supply table
CREATE TABLE Supply (
    ItemNumber INT PRIMARY KEY,
    Name VARCHAR(100),
    Description VARCHAR(150),
    Type VARCHAR(50),
    QuantityInStock INT,
    ReorderLevel INT,
    CostPerUnit DECIMAL(10,2),
    Dosage VARCHAR(50),
    SupplierNumber INT,
    FOREIGN KEY (SupplierNumber) REFERENCES Supplier(SupplierNumber)
);
```

```
-- Requisition table
CREATE TABLE Requisition (
    RequisitionNumber INT PRIMARY KEY,
    StaffID INT,
    WardNumber INT,
    FOREIGN KEY (WardNumber) REFERENCES Ward(WardNumber)
);
```

```
-- Adding foreign key constraint for StaffID in requisition table (missed before)
ALTER TABLE Requisition
ADD CONSTRAINT fk_staffid
FOREIGN KEY (StaffID) REFERENCES Staff(StaffID);
```

```
-- RequisitionDetails table
CREATE TABLE RequisitionDetails (
    RequisitionNumber INT,
    ItemNumber INT,
    Quantity INT,
    DeliveryDate DATE,
    OrderDate DATE,
    PRIMARY KEY (RequisitionNumber, ItemNumber),
    FOREIGN KEY (ItemNumber) REFERENCES Supply(ItemNumber),
    FOREIGN KEY (RequisitionNumber) REFERENCES Requisition(RequisitionNumber)
);
```

- Admission table

```
CREATE TABLE Admission (
    AdmissionID INT PRIMARY KEY,
    PatientNumber INT,
    WardNumber INT,
    DateAdmitted DATE NOT NULL,
    DateDischarged DATE,
    FOREIGN KEY (PatientNumber) REFERENCES Patient(PatientNumber),
    FOREIGN KEY (WardNumber) REFERENCES Ward(WardNumber)
);
```

**-- Medication table**

```
CREATE TABLE Medication (
    PatientNumber INT,
    DrugNumber INT,
    UnitsPerDay INT NOT NULL,
    AdministrationMethod VARCHAR(50) NOT NULL,
    StartDate DATE NOT NULL,
    FinishDate DATE NOT NULL,
    FOREIGN KEY (PatientNumber) REFERENCES Patient(PatientNumber),
    FOREIGN KEY (DrugNumber) REFERENCES Supply(ItemNumber),
    PRIMARY KEY (PatientNumber,DrugNumber,StartDate,FinishDate)
);
```

– TO BE ADDED LATER

-- <adding the foreign key constraint for ChargeNurse <to be added later after inserting data in staff and ward table because of circular dependency cannot add constraint now>

– ALTER TABLE Ward

– ADD CONSTRAINT FK\_ChargeNurse FOREIGN KEY (ChargeNurse) REFERENCES Staff(StaffID);

**Paste the screen shots of the table created**

### 1. Ward table

```
1 •   SELECT * FROM hospitalmanagement.ward;
```

The screenshot shows a database result grid titled "Result Grid". At the top, there are buttons for "Edit" (with icons for edit, insert, delete, and refresh) and "Export/Import". Below the title, there are six columns with headers: "WardNumber", "WardName", "Location", "NumberOfBeds", "TelephoneExtension", and "ChargeNurse". A single row is present, with all cells containing the value "NULL".

	WardNumber	WardName	Location	NumberOfBeds	TelephoneExtension	ChargeNurse
*	NULL	NULL	NULL	NULL	NULL	NULL

### 2. Supplier table

```
1 •   SELECT * FROM hospitalmanagement.supplier;
```

The screenshot shows a database result grid titled "Result Grid". At the top, there are buttons for "Edit" (with icons for edit, insert, delete, and refresh) and "Export/Import". Below the title, there are ten columns with headers: "SupplierNumber", "Name", "Address", "City", "State", "PostCode", "Email", "Phone", and "Fax". A single row is present, with all cells containing the value "NULL".

	SupplierNumber	Name	Address	City	State	PostCode	Email	Phone	Fax
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### 3. Staff table

```
1 •  SELECT * FROM hospitalmanagement.staff;
```

A screenshot of a database result grid titled "Result Grid". The grid has a header row with columns labeled: StaffID, FirstName, LastName, Position, WardNumber, HireDate, PhoneNumber, Email, Address, City, State, and PostCode. Below the header, there is one data row starting with an asterisk (\*) followed by twelve "NULL" entries. The grid includes standard toolbar icons for filtering, editing, and exporting.

	StaffID	FirstName	LastName	Position	WardNumber	HireDate	PhoneNumber	Email	Address	City	State	PostCode
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#### 4. Staffqualification table

```
1 •  SELECT * FROM hospitalmanagement.staffqualification;
```

A screenshot of a database result grid titled "Result Grid". The grid has a header row with columns labeled: StaffID, QualificationName, Institution, and YearObtained. Below the header, there is one data row starting with an asterisk (\*) followed by three "NULL" entries. The grid includes standard toolbar icons for filtering, editing, and exporting.

	StaffID	QualificationName	Institution	YearObtained
*	NULL	NULL	NULL	NULL

#### 5. Staffexperience table

```
1 •  SELECT * FROM hospitalmanagement.staffexperience;
```

A screenshot of a database result grid titled "Result Grid". The grid has a header row with columns labeled: StaffID, JobTitle, Organization, StartDate, and EndDate. Below the header, there is one data row starting with an asterisk (\*) followed by five "NULL" entries. The grid includes standard toolbar icons for filtering, editing, and exporting.

	StaffID	JobTitle	Organization	StartDate	EndDate
*	NULL	NULL	NULL	NULL	NULL

#### 6. Patient table

```
1 •   SELECT * FROM hospitalmanagement.patient;
```

A screenshot of a database result grid titled "Result Grid". The grid has four columns: PatientNumber, Name, Phone, and Email. All four columns contain the value "NULL". The grid includes standard database navigation and editing tools at the top.

PatientNumber	Name	Phone	Email
NULL	NULL	NULL	NULL

## 7. Supply table

```
1 •   SELECT * FROM hospitalmanagement.supply;
```

A screenshot of a database result grid titled "Result Grid". The grid has ten columns: ItemNumber, Name, Description, Type, QuantityInStock, ReorderLevel, CostPerUnit, Dosage, and SupplierNumber. All ten columns contain the value "NULL". The grid includes standard database navigation and editing tools at the top.

ItemNumber	Name	Description	Type	QuantityInStock	ReorderLevel	CostPerUnit	Dosage	SupplierNumber
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## 8. Requisition table

```
1 •   SELECT * FROM hospitalmanagement.requisition;
```

	RequisitionNumber	StaffID	WardNumber
*	NULL	NULL	NULL

#### 9. Requisitiondetails table

```
1 •   SELECT * FROM hospitalmanagement.requisitiondetails;
```

	RequisitionNumber	ItemNumber	Quantity	DeliveryDate	OrderDate
1	NULL	NULL	NULL	NULL	NULL

#### 10. Admission table

```
1 •   SELECT * FROM hospitalmanagement.admission;
```

The screenshot shows a database grid interface with the following details:

- Toolbar:** Includes "Result Grid" (selected), "Filter Rows:" (with a search bar), "Edit" (with icons for edit, insert, delete, and refresh), and "Export/Import".
- Table Headers:** AdmissionID, PatientNumber, WardNumber, DateAdmitted, DateDischarged.
- Table Data:** A single row with all columns set to NULL.

	AdmissionID	PatientNumber	WardNumber	DateAdmitted	DateDischarged
*	NULL	NULL	NULL	NULL	NULL

#### 11. Medication table

```
1 •   SELECT * FROM hospitalmanagement.medication;
```

The screenshot shows a database grid interface with the following details:

- Toolbar:** Includes "Result Grid" (selected), "Filter Rows:" (with a search bar), "Edit" (with icons for edit, insert, delete, and refresh), and "Export/Import".
- Table Headers:** PatientNumber, DrugNumber, UnitsPerDay, AdministrationMethod, StartDate, FinishDate.
- Table Data:** A single row with all columns set to NULL.

	PatientNumber	DrugNumber	UnitsPerDay	AdministrationMethod	StartDate	FinishDate
*	NULL	NULL	NULL	NULL	NULL	NULL

**Step 5.3 Insert at least ten records in each table, preserving all constraints. Put in enough data to demonstrate how the database will function.**

### 1. WardTable

```
INSERT INTO Ward (WardNumber, WardName, Location, NumberOfBeds, TelephoneExtension, ChargeNurse) VALUES  
(1, 'General', '1st Floor', 20, '1234', 201),  
(2, 'Pediatrics', '2nd Floor', 15, '5678', 202),  
(3, 'ICU', '3rd Floor', 10, '9101', 203),  
(4, 'Maternity', '4th Floor', 25, '1122', 204),  
(5, 'Oncology', '5th Floor', 30, '3344', 205),  
(6, 'Cardiology', '6th Floor', 20, '5566', 206),  
(7, 'Neurology', '7th Floor', 15, '7788', 207),  
(8, 'Orthopedics', '8th Floor', 18, '9900', 208),  
(9, 'Dermatology', '9th Floor', 12, '2233', 209),  
(10, 'Psychiatry', '10th Floor', 20, '4455', 210);
```

	WardNumber	WardName	Location	NumberOfBeds	TelephoneExtension	ChargeNurse
▶	1	General	1st Floor	20	1234	201
	2	Pediatrics	2nd Floor	15	5678	202
	3	ICU	3rd Floor	10	9101	203
	4	Maternity	4th Floor	25	1122	204
	5	Oncology	5th Floor	30	3344	205
	6	Cardiology	6th Floor	20	5566	206
	7	Neurology	7th Floor	15	7788	207
	8	Orthopedics	8th Floor	18	9900	208
	9	Dermatology	9th Floor	12	2233	209
	10	Psychiatry	10th Floor	20	4455	210
*	NULL	NULL	NULL	NULL	NULL	NULL

### 2. Staff Table

```
INSERT INTO Staff (StaffID, FirstName, LastName, Position, WardNumber, HireDate, PhoneNumber, Email, Address, City, State, PostCode)  
VALUES  
(201, 'John', 'Doe', 'Nurse', 1, '2020-01-15', '217-555-1101', 'johndoe@example.com', '101  
Maple St', 'Springfield', 'IL', '62701'),  
(202, 'Jane', 'Smith', 'Nurse', 2, '2018-06-01', '402-555-1201', 'janessmith@example.com', '202  
Elm St', 'Lincoln', 'NE', '68508'),
```

(203, 'Alice', 'Johnson', 'Nurse', 3, '2019-03-22', '608-555-1301', 'alicejohnson@example.com',  
'303 Oak St', 'Madison', 'WI', '53703'),  
(204, 'Michael', 'Brown', 'Nurse', 4, '2021-05-10', '312-555-1401', 'michaelbrown@example.com',  
'404 Pine St', 'Chicago', 'IL', '60601'),  
(205, 'Emma', 'Wilson', 'Nurse', 5, '2017-09-14', '313-555-1501', 'emmwilson@example.com',  
'505 Oak St', 'Detroit', 'MI', '48201'),  
(206, 'Oliver', 'Martinez', 'Nurse', 6, '2019-12-30', '314-555-1601',  
'olivermartinez@example.com', '606 Maple St', 'St. Louis', 'MO', '63101'),  
(207, 'Sophia', 'Garcia', 'Nurse', 7, '2022-02-20', '612-555-1701', 'sophiagarcia@example.com',  
'707 Elm St', 'Minneapolis', 'MN', '55401'),  
(208, 'James', 'Lee', 'Nurse', 8, '2016-11-03', '317-555-1801', 'jameslee@example.com', '808  
Pine St', 'Indianapolis', 'IN', '46201'),  
(209, 'Isabella', 'Hernandez', 'Nurse', 9, '2020-04-25', '614-555-1901',  
'isabellahernandez@example.com', '909 Oak St', 'Columbus', 'OH', '43215'),  
(210, 'David', 'Clark', 'Nurse', 1, '2021-03-12', '217-555-2101', 'davidclark@example.com', '102  
Birch St', 'Springfield', 'IL', '62701'),  
(211, 'Olivia', 'White', 'Doctor', 2, '2019-07-08', '402-555-2201', 'oliviawhite@example.com', '203  
Cedar St', 'Lincoln', 'NE', '68508'),  
(212, 'Liam', 'Davis', 'Surgeon', 3, '2020-10-17', '608-555-2301', 'liamdavis@example.com', '304  
Pine St', 'Madison', 'WI', '53703'),  
(213, 'Mia', 'Rodriguez', 'Nurse', 4, '2017-05-05', '312-555-2401', 'miarodriguez@example.com',  
'405 Oak St', 'Chicago', 'IL', '60601'),  
(214, 'Noah', 'Walker', 'Doctor', 5, '2021-01-22', '313-555-2501', 'noahwalker@example.com',  
'506 Maple St', 'Detroit', 'MI', '48201'),  
(215, 'Sophia', 'Hall', 'Surgeon', 6, '2018-09-30', '314-555-2601', 'sophiahall@example.com',  
'607 Birch St', 'St. Louis', 'MO', '63101'),  
(216, 'Lucas', 'Allen', 'Nurse', 7, '2020-04-14', '612-555-2701', 'lucasallen@example.com', '708  
Elm St', 'Minneapolis', 'MN', '55401'),  
(217, 'Amelia', 'Scott', 'Doctor', 8, '2016-06-19', '317-555-2801', 'ameliascott@example.com',  
'809 Cedar St', 'Indianapolis', 'IN', '46201'),  
(218, 'Henry', 'Lewis', 'Surgeon', 9, '2019-12-03', '614-555-2901', 'henrylewis@example.com',  
'910 Oak St', 'Columbus', 'OH', '43215'),  
(219, 'Chloe', 'King', 'Nurse', 10, '2022-07-25', '614-555-3001', 'chloeking@example.com', '911  
Pine St', 'Columbus', 'OH', '43215'),  
(220, 'Daniel', 'Moore', 'Doctor', 1, '2018-11-10', '217-555-3101', 'danielmoore@example.com',  
'111 Maple St', 'Springfield', 'IL', '62701');

	StaffID	FirstName	LastName	Position	WardNumber	HireDate	PhoneNumber	Email	Address	City	State	PostCode
▶	201	John	Doe	Nurse	1	2020-01-15	217-555-1101	johndoe@example.com	101 Maple St	Springfield	IL	62701
	202	Jane	Smith	Nurse	2	2018-06-01	402-555-1201	janesmith@example.com	202 Elm St	Lincoln	NE	68508
	203	Alice	Johnson	Nurse	3	2019-03-22	608-555-1301	alicejohnson@example.com	303 Oak St	Madison	WI	53703
	204	Michael	Brown	Nurse	4	2021-05-10	312-555-1401	michaelbrown@example.com	404 Pine St	Chicago	IL	60601
	205	Emma	Wilson	Nurse	5	2017-09-14	313-555-1501	emmawilson@example.com	505 Oak St	Detroit	MI	48201
	206	Oliver	Martinez	Nurse	6	2019-12-30	314-555-1601	olivermartinez@example.com	606 Maple St	St. Louis	MO	63101
	207	Sophia	Garcia	Nurse	7	2022-02-20	612-555-1701	sophiagarcia@example.com	707 Elm St	Minneapolis	MN	55401
	208	James	Lee	Nurse	8	2016-11-03	317-555-1801	jameslee@example.com	808 Pine St	Indianapolis	IN	46201
	209	Isabella	Hernandez	Nurse	9	2020-04-25	614-555-1901	isabellahernandez@example.com	909 Oak St	Columbus	OH	43215
	210	David	Clark	Nurse	1	2021-03-12	217-555-2101	daviddark@example.com	102 Birch St	Springfield	IL	62701
	211	Olivia	White	Doctor	2	2019-07-08	402-555-2201	oliviawhite@example.com	203 Cedar St	Lincoln	NE	68508
	212	Liam	Davis	Surgeon	3	2020-10-17	608-555-2301	liamdavis@example.com	304 Pine St	Madison	WI	53703
	213	Mia	Rodriguez	Nurse	4	2017-05-05	312-555-2401	miarodriguez@example.com	405 Oak St	Chicago	IL	60601
	214	Noah	Walker	Doctor	5	2021-01-22	313-555-2501	noahwalker@example.com	506 Maple St	Detroit	MI	48201
	215	Sophia	Hall	Surgeon	6	2018-09-30	314-555-2601	sophiahall@example.com	607 Birch St	St. Louis	MO	63101
	216	Lucas	Allen	Nurse	7	2020-04-14	612-555-2701	lucasallen@example.com	708 Elm St	Minneapolis	MN	55401
	217	Amelia	Scott	Doctor	8	2016-06-19	317-555-2801	ameliascott@example.com	809 Cedar St	Indianapolis	IN	46201
	218	Henry	Lewis	Surgeon	9	2019-12-03	614-555-2901	henrylewis@example.com	910 Oak St	Columbus	OH	43215
	219	Chloe	King	Nurse	10	2022-07-25	614-555-3001	chloeking@example.com	911 Pine St	Columbus	OH	43215
	220	Daniel	Moore	Doctor	1	2018-11-10	217-555-3101	danielmoore@example.com	111 Maple St	Springfield	IL	62701

### – Adding foreign key constraint for ChargeNurse in WardTable

```
-- adding the foreign key constraint for ChargeNurse in Ward table
-- This is done after inserting data as staff table is not created yet and ChargeNurse is foreign key referencing
-- StaffID in Staff table. So circular dependency is there
ALTER TABLE Ward
ADD CONSTRAINT FK_ChargeNurse FOREIGN KEY (ChargeNurse) REFERENCES Staff(StaffID);
```

### 3. StaffQualification table

```
INSERT INTO StaffQualification (StaffID, QualificationName, Institution, YearObtained)
VALUES (201, 'Bachelor of Nursing', 'Springfield University', 2015),
(201, 'Advanced Cardiac Life Support', 'American Heart Association', 2017),
(202, 'Bachelor of Nursing', 'University of Nebraska', 2014),
(202, 'Pediatric Nursing Certification', 'Pediatric Nursing Certification Board', 2016),
(203, 'Bachelor of Nursing', 'University of Wisconsin', 2016),
(203, 'Trauma Nursing Core Course', 'Emergency Nurses Association', 2018),
(204, 'Bachelor of Nursing', 'University of Chicago', 2018),
(204, 'Certified Emergency Nurse', 'Board of Certification for Emergency Nursing', 2019),
(205, 'Bachelor of Nursing', 'University of Michigan', 2013),
(205, 'Oncology Nursing Certification', 'Oncology Nursing Certification Corporation', 2015),
(206, 'Bachelor of Nursing', 'St. Louis University', 2016),
(206, 'Critical Care Nursing Certification', 'American Association of Critical-Care Nurses', 2018),
(207, 'Bachelor of Nursing', 'University of Minnesota', 2019),
(207, 'Psychiatric Nursing Certification', 'American Nurses Credentialing Center', 2021),
(208, 'Bachelor of Nursing', 'Indiana University', 2012),
(208, 'Orthopedic Nursing Certification', 'Orthopaedic Nurses Certification Board', 2014),
(209, 'Bachelor of Nursing', 'Ohio State University', 2015),
(209, 'Dermatology Nursing Certification', 'Dermatology Nurses' Association', 2017),
```

(210, 'Bachelor of Nursing', 'Springfield University', 2018),  
 (210, 'Neonatal Resuscitation Program', 'American Academy of Pediatrics', 2020),  
 (211, 'Doctor of Medicine', 'University of Nebraska Medical Center', 2014),  
 (211, 'Pediatric Advanced Life Support', 'American Heart Association', 2016),  
 (212, 'Doctor of Medicine', 'University of Wisconsin School of Medicine', 2016),  
 (212, 'Advanced Trauma Life Support', 'American College of Surgeons', 2018),  
 (213, 'Bachelor of Nursing', 'University of Chicago', 2013),  
 (213, 'Critical Care Registered Nurse', 'American Association of Critical-Care Nurses', 2015),  
 (214, 'Doctor of Medicine', 'University of Michigan Medical School', 2017),  
 (214, 'Advanced Cardiovascular Life Support', 'American Heart Association', 2019),  
 (215, 'Doctor of Medicine', 'St. Louis University School of Medicine', 2015),  
 (215, 'Board Certified in Surgery', 'American Board of Surgery', 2017),  
 (216, 'Bachelor of Nursing', 'University of Minnesota', 2016),  
 (216, 'Certified Pediatric Nurse', 'Pediatric Nursing Certification Board', 2018),  
 (217, 'Doctor of Medicine', 'Indiana University School of Medicine', 2012),  
 (217, 'Certified Anesthesiologist', 'American Board of Anesthesiology', 2014),  
 (218, 'Doctor of Medicine', 'Ohio State University College of Medicine', 2014),  
 (218, 'Board Certified in Dermatology', 'American Board of Dermatology', 2016),  
 (219, 'Bachelor of Nursing', 'Ohio State University', 2020),  
 (219, 'Certified Medical-Surgical Registered Nurse', 'Medical-Surgical Nursing Certification Board', 2022),  
 (220, 'Doctor of Medicine', 'University of Illinois College of Medicine', 2014),  
 (220, 'Board Certified in Family Medicine', 'American Board of Family Medicine', 2016);

	StaffID	QualificationName	Institution	YearObtained
►	201	Advanced Cardiac Life Support	American Heart Association	2017
	201	Bachelor of Nursing	Springfield University	2015
	202	Bachelor of Nursing	University of Nebraska	2014
	202	Pediatric Nursing Certification	Pediatric Nursing Certification Board	2016
	203	Bachelor of Nursing	University of Wisconsin	2016
	203	Trauma Nursing Core Course	Emergency Nurses Association	2018
	204	Bachelor of Nursing	University of Chicago	2018
	204	Certified Emergency Nurse	Board of Certification for Emergency Nursing	2019
	205	Bachelor of Nursing	University of Michigan	2013
	205	Oncology Nursing Certification	Oncology Nursing Certification Corporation	2015
	206	Bachelor of Nursing	St. Louis University	2016
	206	Critical Care Nursing Certificat...	American Association of Critical-Care Nurses	2018
	207	Bachelor of Nursing	University of Minnesota	2019
	207	Psychiatric Nursing Certification	American Nurses Credentialing Center	2021
	208	Bachelor of Nursing	Indiana University	2012
	208	Orthopedic Nursing Certification	Orthopaedic Nurses Certification Board	2014
	209	Bachelor of Nursing	Ohio State University	2015
	209	Dermatology Nursing Certifica...	Dermatology Nurses' Association	2017

	StaffID	QualificationName	Institution	YearObtained
	210	Bachelor of Nursing	Springfield University	2018
	210	Neonatal Resuscitation Program	American Academy of Pediatrics	2020
	211	Doctor of Medicine	University of Nebraska Medical Center	2014
	211	Pediatric Advanced Life Support	American Heart Association	2016
	212	Advanced Trauma Life Support	American College of Surgeons	2018
	212	Doctor of Medicine	University of Wisconsin School of Medicine	2016
	213	Bachelor of Nursing	University of Chicago	2013
	213	Critical Care Registered Nurse	American Association of Critical-Care Nurses	2015
	214	Advanced Cardiovascular Life ...	American Heart Association	2019
	214	Doctor of Medicine	University of Michigan Medical School	2017
	215	Board Certified in Surgery	American Board of Surgery	2017
	215	Doctor of Medicine	St. Louis University School of Medicine	2015
	216	Bachelor of Nursing	University of Minnesota	2016
	216	Certified Pediatric Nurse	Pediatric Nursing Certification Board	2018
	217	Certified Anesthesiologist	American Board of Anesthesiology	2014
	217	Doctor of Medicine	Indiana University School of Medicine	2012
	218	Board Certified in Dermatology	American Board of Dermatology	2016
	218	Doctor of Medicine	Ohio State University College of Medicine	2014
	219	Bachelor of Nursing	Ohio State University	2020
	219	Certified Medical-Surgical Regi...	Medical-Surgical Nursing Certification Board	2022
	220	Board Certified in Family Medi...	American Board of Family Medicine	2016
	220	Doctor of Medicine	University of Illinois College of Medicine	2014

#### 4. Supplier Table

```
INSERT INTO Supplier (SupplierNumber, Name, Address, City, State, PostCode, Email, Phone, Fax) VALUES
(1, 'MedSupply Co.', '123 Elm St', 'Springfield', 'IL', '62701', 'medsupply@example.com', '217-555-0101', '217-555-0102'),
(2, 'Health Equip', '456 Oak St', 'Lincoln', 'NE', '68508', 'healthequip@example.com', '402-555-0103', '402-555-0104'),
(3, 'PharmaGoods', '789 Pine St', 'Madison', 'WI', '53703', 'pharmagoods@example.com', '608-555-0105', '608-555-0106'),
(4, 'MedTech Supplies', '321 Cedar St', 'Omaha', 'NE', '68102', 'medtech@example.com', '402-555-0201', '402-555-0202'),
(5, 'HealthLine', '654 Birch St', 'Chicago', 'IL', '60601', 'healthline@example.com', '312-555-0301', '312-555-0302'),
(6, 'CareEssentials', '987 Maple St', 'Detroit', 'MI', '48201', 'careessentials@example.com', '313-555-0401', '313-555-0402'),
```

(7, 'MediFast', '159 Pine St', 'St. Louis', 'MO', '63101', 'medifast@example.com', '314-555-0501', '314-555-0502'),  
 (8, 'QuickMed Supplies', '753 Elm St', 'Minneapolis', 'MN', '55401', 'quickmed@example.com', '612-555-0601', '612-555-0602'),  
 (9, 'PrimeHealth', '246 Oak St', 'Indianapolis', 'IN', '46201', 'primehealth@example.com', '317-555-0701', '317-555-0702'),  
 (10, 'MediSupply Central', '864 Cedar St', 'Columbus', 'OH', '43215', 'medisupplycentral@example.com', '614-555-0801', '614-555-0802');

	SupplierNumber	Name	Address	City	State	PostCode	Email	Phone	Fax
▶	1	MedSupply Co.	123 Elm St	Springfield	IL	62701	medsupply@example.com	217-555-0101	217-555-0102
	2	Health Equip	456 Oak St	Lincoln	NE	68508	healthequip@example.com	402-555-0103	402-555-0104
	3	PharmaGoods	789 Pine St	Madison	WI	53703	pharmagoods@example.com	608-555-0105	608-555-0106
	4	MedTech Supplies	321 Cedar St	Omaha	NE	68102	medtech@example.com	402-555-0201	402-555-0202
	5	HealthLine	654 Birch St	Chicago	IL	60601	healthline@example.com	312-555-0301	312-555-0302
	6	CareEssentials	987 Maple St	Detroit	MI	48201	careessentials@example.com	313-555-0401	313-555-0402
	7	MediFast	159 Pine St	St. Louis	MO	63101	medifast@example.com	314-555-0501	314-555-0502
	8	QuickMed Supplies	753 Elm St	Minneapolis	MN	55401	quickmed@example.com	612-555-0601	612-555-0602
	9	PrimeHealth	246 Oak St	Indianapolis	IN	46201	primehealth@example.com	317-555-0701	317-555-0702
	10	MediSupply Central	864 Cedar St	Columbus	OH	43215	medisupplycentral@example.com	614-555-0801	614-555-0802

## 5. StaffExperience table

-- Staff member 201

```
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES
(201, 'Nurse', 'Springfield Hospital', '2020-01-15', '2022-01-14'),
(201, 'Nurse', 'Springfield Clinic', '2022-01-15', '2023-01-14');
```

-- Staff member 202

```
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES
(202, 'Nurse', 'Lincoln Medical Center', '2017-06-01', '2021-06-01');
```

-- Staff member 203

```
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES
(203, 'Nurse', 'Madison Health Clinic', '2019-03-22', '2022-03-22'),
(203, 'Nurse', 'Madison Community Hospital', '2022-03-23', '2024-03-23');
```

-- Staff member 204

```
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES
(204, 'Nurse', 'Chicago Medical Group', '2021-05-10', '2023-05-10'),
(204, 'Nurse', 'Chicago General Hospital', '2023-05-11', '2024-05-10');
```

-- Staff member 205

```
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES
(205, 'Nurse', 'Detroit Health Services', '2014-09-14', '2019-09-14'),
(205, 'Nurse', 'Detroit Specialty Care', '2019-09-15', '2022-09-14');
```

-- Staff member 206  
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(206, 'Nurse', 'St. Louis Hospital', '2019-12-30', '2022-12-30'),  
(206, 'Nurse', 'St. Louis Health Clinic', '2022-12-31', '2024-12-30');

-- Staff member 207  
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(207, 'Nurse', 'Minneapolis Medical Center', '2022-02-20', '2024-02-20');

-- Staff member 208  
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(208, 'Nurse', 'Indianapolis Clinic', '2016-11-03', '2019-11-03'),  
(208, 'Nurse', 'Indianapolis General Hospital', '2019-11-04', '2022-11-03');

-- Staff member 209  
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(209, 'Nurse', 'Columbus Health Center', '2017-04-25', '2022-04-25');

-- Staff member 210  
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(210, 'Nurse', 'Springfield Medical Group', '2021-03-12', '2023-03-12'),  
(210, 'Nurse', 'Springfield General', '2023-03-13', '2024-03-12');

-- Staff member 211  
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(211, 'Doctor', 'Lincoln General Hospital', '2018-07-08', '2022-07-08'),  
(211, 'Doctor', 'Lincoln Medical Center', '2022-07-09', '2024-07-08');

-- Staff member 212  
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(212, 'Surgeon', 'Madison Hospital', '2020-10-17', '2023-10-17'),  
(212, 'Surgeon', 'Madison Specialty Clinic', '2023-10-18', '2024-10-17');

-- Staff member 213  
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(213, 'Nurse', 'Chicago Health Clinic', '2017-05-05', '2020-05-05'),  
(213, 'Nurse', 'Chicago General Hospital', '2020-05-06', '2023-05-05');

-- Staff member 214  
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(214, 'Doctor', 'Detroit Medical Center', '2021-01-22', '2023-01-22'),  
(214, 'Doctor', 'Detroit General Hospital', '2023-01-23', '2024-01-22');

-- Staff member 215

```
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(215, 'Surgeon', 'St. Louis Health Clinic', '2018-09-30', '2021-09-30'),  
(215, 'Surgeon', 'St. Louis General Hospital', '2021-10-01', '2024-09-30');
```

-- Staff member 216

```
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(216, 'Nurse', 'Minneapolis Medical Center', '2020-04-14', '2022-04-14'),  
(216, 'Nurse', 'Minneapolis Health Clinic', '2022-04-15', '2024-04-14');
```

-- Staff member 217

```
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(217, 'Doctor', 'Indianapolis General Hospital', '2015-06-19', '2019-06-19'),  
(217, 'Doctor', 'Indianapolis Specialty Clinic', '2019-06-20', '2022-06-19');
```

-- Staff member 218

```
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(218, 'Surgeon', 'Columbus Health Center', '2019-12-03', '2022-12-03'),  
(218, 'Surgeon', 'Columbus Medical Clinic', '2022-12-04', '2024-12-03');
```

-- Staff member 219

```
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(219, 'Nurse', 'Columbus Health Center', '2022-07-25', '2024-07-25');
```

-- Staff member 220

```
INSERT INTO StaffExperience (StaffID, JobTitle, Organization, StartDate, EndDate) VALUES  
(220, 'Nurse', 'Springfield Medical Center', '2018-11-10', '2021-11-10'),  
(220, 'Doctor', 'Springfield General Hospital', '2021-11-11', '2024-11-10');
```

	StaffID	JobTitle	Organization	StartDate	EndDate
▶	201	Nurse	Springfield Clinic	2022-01-15	2023-01-14
	201	Nurse	Springfield Hospital	2020-01-15	2022-01-14
	202	Nurse	Lincoln Medical Center	2017-06-01	2021-06-01
	203	Nurse	Madison Community Hospital	2022-03-23	2024-03-23
	203	Nurse	Madison Health Clinic	2019-03-22	2022-03-22
	204	Nurse	Chicago General Hospital	2023-05-11	2024-05-10
	204	Nurse	Chicago Medical Group	2021-05-10	2023-05-10
	205	Nurse	Detroit Health Services	2014-09-14	2019-09-14
	205	Nurse	Detroit Specialty Care	2019-09-15	2022-09-14
	206	Nurse	St. Louis Health Clinic	2022-12-31	2024-12-30
	206	Nurse	St. Louis Hospital	2019-12-30	2022-12-30
	207	Nurse	Minneapolis Medical Center	2022-02-20	2024-02-20
	208	Nurse	Indianapolis Clinic	2016-11-03	2019-11-03
	208	Nurse	Indianapolis General Hospital	2019-11-04	2022-11-03
	209	Nurse	Columbus Health Center	2017-04-25	2022-04-25
	210	Nurse	Springfield General	2023-03-13	2024-03-12
	210	Nurse	Springfield Medical Group	2021-03-12	2023-03-12
▶	211	Doctor	Lincoln General Hospital	2018-07-08	2022-07-08
	211	Doctor	Lincoln Medical Center	2022-07-09	2024-07-08
	212	Surgeon	Madison Hospital	2020-10-17	2023-10-17
	212	Surgeon	Madison Specialty Clinic	2023-10-18	2024-10-17
	213	Nurse	Chicago General Hospital	2020-05-06	2023-05-05
	213	Nurse	Chicago Health Clinic	2017-05-05	2020-05-05
	214	Doctor	Detroit General Hospital	2023-01-23	2024-01-22
	214	Doctor	Detroit Medical Center	2021-01-22	2023-01-22
	215	Surgeon	St. Louis General Hospital	2021-10-01	2024-09-30
	215	Surgeon	St. Louis Health Clinic	2018-09-30	2021-09-30
	216	Nurse	Minneapolis Health Clinic	2022-04-15	2024-04-14
	216	Nurse	Minneapolis Medical Center	2020-04-14	2022-04-14
	217	Doctor	Indianapolis General Hospital	2015-06-19	2019-06-19
	217	Doctor	Indianapolis Specialty Clinic	2019-06-20	2022-06-19
	218	Surgeon	Columbus Health Center	2019-12-03	2022-12-03
	218	Surgeon	Columbus Medical Clinic	2022-12-04	2024-12-03
	219	Nurse	Columbus Health Center	2022-07-25	2024-07-25
	220	Doctor	Springfield General Hospital	2021-11-11	2024-11-10
	220	Nurse	Springfield Medical Center	2018-11-10	2021-11-10

## 6. SupplyTable

#### -- Surgical Supplies

```
INSERT INTO Supply (ItemNumber, Name, Description, Type, QuantityInStock, ReorderLevel, CostPerUnit, SupplierNumber) VALUES
```

(101, 'Syringes', 'Sterile syringes', 'surgical', 500, 100, 0.50, 1),  
(102, 'Sterile Dressings', 'Sterile gauze dressings', 'surgical', 300, 50, 1.25, 2),  
(103, 'Scalpels', 'Surgical scalpels', 'surgical', 150, 30, 5.00, 3),  
(104, 'Surgical Gloves', 'Latex surgical gloves', 'surgical', 1000, 200, 0.15, 4),  
(105, 'Suture Kits', 'Suture kits for wound closure', 'surgical', 200, 50, 7.50, 5);

## -- Nonsurgical Supplies

```
INSERT INTO Supply (ItemNumber, Name, Description, Type, QuantityInStock, ReorderLevel, CostPerUnit, SupplierNumber) VALUES
```

(201, 'Plastic Bags', 'Disposable plastic bags', 'nonsurgical', 800, 150, 0.10, 6),  
(202, 'Aprons', 'Disposable aprons', 'nonsurgical', 400, 100, 0.75, 7),  
(203, 'Face Masks', 'Surgical face masks', 'nonsurgical', 600, 120, 0.20, 8),  
(204, 'Gloves', 'Disposable gloves', 'nonsurgical', 1200, 250, 0.05, 9),  
(205, 'Bandages', 'Elastic bandages', 'nonsurgical', 350, 70, 2.00, 6);

### -- Pharmaceutical Supplies

```
INSERT INTO Supply (ItemNumber, Name, Description, Type, QuantityInStock, ReorderLevel, CostPerUnit, Dosage, SupplierNumber) VALUES
```

(301, 'Antibiotics', 'Broad-spectrum antibiotics', 'pharmaceutical', 100, 20, 10.00, '500mg', 1),  
(302, 'Painkillers', 'Non-opioid pain relief', 'pharmaceutical', 200, 40, 5.00, '250mg', 2),  
(303, 'Antiseptics', 'Topical antiseptics', 'pharmaceutical', 150, 30, 3.50, '10%', 3),  
(304, 'Cough Syrup', 'Expectorant cough syrup', 'pharmaceutical', 180, 35, 6.00, '100ml', 4),  
(305, 'Insulin', 'Rapid-acting insulin', 'pharmaceutical', 120, 25, 20.00, '10ml', 5);

## 7 Requisition Table

```
INSERT INTO Requisition (RequisitionNumber, StaffID, WardNumber) VALUES  
(1, 203, 3),  
(2, 209, 9),  
(3, 206, 6),  
(4, 203, 3),  
(5, 210, 10),  
(6, 208, 8),  
(7, 210, 10),  
(8, 208, 8),  
(9, 205, 5),  
(10, 206, 6),  
(11, 209, 9),  
(12, 206, 6),  
(13, 201, 1),  
(14, 203, 3),  
(15, 210, 10);
```

	RequisitionNumber	StaffID	WardNumber
▶	1	203	3
	2	209	9
	3	206	6
	4	203	3
	5	210	10
	6	208	8
	7	210	10
	8	208	8
	9	205	5
	10	206	6
	11	209	9
	12	206	6
	13	201	1
	14	203	3
	15	210	10
*	NULL	NULL	NULL

## 8. Requisition Details

-- Requisition 1

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,  
OrderDate) VALUES (1, 101, 50, '2024-04-01', '2024-03-25');  
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,  
OrderDate) VALUES (1, 201, 100, '2024-04-01', '2024-03-25');
```

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (1, 301, 20, NULL, '2024-03-25');
```

-- Requisition 2

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (2, 102, 60, '2024-04-05', '2024-03-28');
```

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (2, 202, 150, NULL, '2024-03-28');
```

-- Requisition 3

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (3, 103, 30, '2024-06-05', '2024-05-20');
```

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (3, 303, 25, '2024-06-06', '2024-05-20');
```

-- Requisition 4

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (4, 104, 100, NULL, '2024-06-15');
```

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (4, 203, 60, NULL, '2024-06-15');
```

-- Requisition 5

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (5, 105, 15, '2024-06-18', '2024-06-01');
```

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (5, 304, 40, '2024-06-18', '2024-06-01');
```

-- Requisition 6

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (6, 204, 200, NULL, '2024-06-20');
```

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (6, 305, 10, '2024-07-01', '2024-06-20');
```

-- Requisition 7

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (7, 105, 20, NULL, '2024-07-05');
```

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (7, 304, 30, '2024-07-05', '2024-07-05');
```

-- Requisition 8

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (8, 101, 80, NULL, '2024-08-01');
```

```
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate, OrderDate) VALUES (8, 204, 150, '2024-08-10', '2024-08-01');
```

```
-- Requisition 9
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,
OrderDate) VALUES (9, 302, 25, NULL, '2024-08-20');

INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,
OrderDate) VALUES (9, 103, 30, '2024-09-01', '2024-08-20');

-- Requisition 10
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,
OrderDate) VALUES (10, 102, 45, '2024-10-01', '2024-09-25');

INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,
OrderDate) VALUES (10, 301, 15, '2024-10-01', '2024-09-25');

-- Requisition 11
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,
OrderDate) VALUES (11, 104, 100, '2024-10-12', '2024-10-05');

INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,
OrderDate) VALUES (11, 203, 60, NULL, '2024-10-05');

-- Requisition 12
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,
OrderDate) VALUES (12, 205, 50, NULL, '2024-11-01');

INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,
OrderDate) VALUES (12, 302, 40, NULL, '2024-11-01');

-- Requisition 13
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,
OrderDate) VALUES (13, 105, 20, '2024-12-01', '2024-11-20');

INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,
OrderDate) VALUES (13, 303, 25, '2024-12-05', '2024-11-20');

-- Requisition 14
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,
OrderDate) VALUES (14, 101, 90, NULL, '2024-12-05');

INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,
OrderDate) VALUES (14, 304, 40, '2024-12-10', '2024-12-05');

-- Requisition 15
INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,
OrderDate) VALUES (15, 105, 25, '2024-12-25', '2024-12-01');

INSERT INTO RequisitionDetails (RequisitionNumber, ItemNumber, Quantity, DeliveryDate,
OrderDate) VALUES (15, 201, 60, '2024-12-25', '2024-12-01');
```

	RequisitionNumber	ItemNumber	Quantity	DeliveryDate	OrderDate
▶	1	101	50	2024-04-01	2024-03-25
	1	201	100	2024-04-01	2024-03-25
	1	301	20	NULL	2024-03-25
	2	102	60	2024-04-05	2024-03-28
	2	202	150	NULL	2024-03-28
	3	103	30	2024-06-05	2024-05-20
	3	303	25	2024-06-06	2024-05-20
	4	104	100	NULL	2024-06-15
	5	105	15	2024-06-18	2024-06-01
	6	204	200	NULL	2024-06-20
	6	305	10	2024-07-01	2024-06-20
	7	304	30	2024-07-05	2024-07-05
	8	101	80	NULL	2024-08-01
	8	204	150	2024-08-10	2024-08-01
	9	103	30	2024-09-01	2024-08-20
	9	302	25	NULL	2024-08-20
	10	102	45	2024-10-01	2024-09-25
	11	104	100	2024-10-12	2024-10-05
	11	203	60	NULL	2024-10-05
	12	205	50	NULL	2024-11-01
	12	302	40	NULL	2024-11-01
	13	105	20	2024-12-01	2024-11-20
	14	101	90	NULL	2024-12-05
	14	304	40	2024-12-10	2024-12-05
	15	105	25	2024-12-25	2024-12-01

## 9. Patient table

```
INSERT INTO Patient (PatientNumber, Name, Phone, Email) VALUES  
(1, 'John Doe', '555-1234', 'johndoe@example.com'),  
(2, 'Jane Smith', '555-5678', 'janeshmith@example.com'),  
(3, 'Michael Johnson', '555-9012', 'michaeljohnson@example.com'),  
(4, 'Emily Davis', '555-3456', 'emilydavis@example.com'),  
(5, 'David Brown', '555-7890', 'davidbrown@example.com'),  
(6, 'Linda Miller', '555-2345', 'lindamiller@example.com'),  
(7, 'James Wilson', '555-6789', 'jameswilson@example.com'),  
(8, 'Patricia Taylor', '555-1122', 'patriciataylor@example.com'),  
(9, 'Robert Moore', '555-3344', 'robertmoore@example.com'),  
(10, 'Barbara Anderson', '555-5566', 'barbaraanderson@example.com'),  
(11, 'Thomas Jackson', '555-7788', 'thomasjackson@example.com'),  
(12, 'Nancy White', '555-9900', 'nancywhite@example.com'),  
(13, 'William Harris', '555-2233', 'williamharris@example.com'),  
(14, 'Karen Martin', '555-4455', 'karenmartin@example.com'),  
(15, 'Christopher Lee', '555-6677', 'christopherlee@example.com');
```

	PatientNumber	Name	Phone	Email
▶	1	John Doe	555-1234	johndoe@example.com
	2	Jane Smith	555-5678	janeshmith@example.com
	3	Michael Johnson	555-9012	michaeljohnson@example.com
	4	Emily Davis	555-3456	emilydavis@example.com
	5	David Brown	555-7890	davidbrown@example.com
	6	Linda Miller	555-2345	lindamiller@example.com
	7	James Wilson	555-6789	jameswilson@example.com
	8	Patricia Taylor	555-1122	patriciataylor@example.com
	9	Robert Moore	555-3344	robertmoore@example.com
	10	Barbara Anderson	555-5566	barbaraanderson@example.com
	11	Thomas Jackson	555-7788	thomasjackson@example.com
	12	Nancy White	555-9900	nancywhite@example.com
	13	William Harris	555-2233	williamharris@example.com
	14	Karen Martin	555-4455	karenmartin@example.com
*	15	Christopher Lee	555-6677	<a href="mailto:christopherlee@example.com">christopherlee@example.com</a>
	NULL	NULL	NULL	NULL

## 10. Admission table

```
INSERT INTO Admission (AdmissionID, PatientNumber, WardNumber, DateAdmitted, DateDischarged) VALUES  
(1, 1, 1, '2024-07-01', '2024-07-05'),  
(3, 3, 3, '2024-07-03', '2024-07-10'),  
(4, 4, 4, '2024-07-04', NULL),  
(5, 5, 1, '2024-07-05', '2024-07-12'),  
(6, 6, 6, '2024-07-06', NULL),  
(7, 7, 7, '2024-07-07', '2024-07-11'),  
(8, 8, 8, '2024-07-08', NULL),  
(9, 9, 9, '2024-07-09', '2024-07-15'),  
(10, 10, 10, '2024-07-10', '2024-07-14'),  
(11, 1, 2, '2024-07-15', NULL),  
(12, 3, 1, '2024-07-16', NULL);
```

	AdmissionID	PatientNumber	WardNumber	DateAdmitted	DateDischarged
▶	1	1	1	2024-07-01	2024-07-05
	3	3	3	2024-07-03	2024-07-10
	4	4	4	2024-07-04	NULL
	5	5	1	2024-07-05	2024-07-12
	6	6	6	2024-07-06	NULL
	7	7	7	2024-07-07	2024-07-11
	8	8	8	2024-07-08	NULL
	9	9	9	2024-07-09	2024-07-15
	10	10	10	2024-07-10	2024-07-14
	11	1	2	2024-07-15	NULL
	12	3	1	2024-07-16	NULL
*	NULL	NULL	NULL	NULL	NULL

## 11. Medication table

```
INSERT INTO Medication (PatientNumber, DrugNumber, UnitsPerDay, AdministrationMethod, StartDate, FinishDate) VALUES  
(1, 301, 2, 'Oral', '2024-07-01', '2024-07-07'),  
(1, 303, 1, 'Topical', '2024-07-16', '2024-07-23'),  
(1, 302, 1, 'Oral', '2024-07-08', '2024-07-14'),  
(2, 303, 1, 'Topical', '2024-07-02', '2024-07-05'),  
(3, 304, 3, 'Oral', '2024-07-03', '2024-07-10'),  
(3, 301, 1, 'Oral', '2024-07-03', '2024-07-10'),  
(4, 302, 3, 'Oral', '2024-07-19', '2024-07-26'),  
(5, 301, 3, 'Oral', '2024-07-05', '2024-07-12'),
```

```
(6, 302, 1, 'Oral', '2024-07-06', '2024-07-10'),
(7, 303, 2, 'Topical', '2024-07-07', '2024-07-13'),
(8, 304, 4, 'Oral', '2024-07-08', '2024-07-15'),
(10, 301, 2, 'Oral', '2024-07-10', '2024-07-17'),
(11, 302, 3, 'Oral', '2024-07-11', '2024-07-18'),
(12, 303, 2, 'Topical', '2024-07-12', '2024-07-19'),
(13, 304, 3, 'Oral', '2024-07-13', '2024-07-20'),
(15, 301, 2, 'Oral', '2024-07-15', '2024-07-22');
```

	PatientNumber	DrugNumber	UnitsPerDay	AdministrationMethod	StartDate	FinishDate
▶	1	301	2	Oral	2024-07-01	2024-07-07
	1	302	1	Oral	2024-07-08	2024-07-14
	1	303	1	Topical	2024-07-16	2024-07-23
	2	303	1	Topical	2024-07-02	2024-07-05
	3	301	1	Oral	2024-07-03	2024-07-10
	3	304	3	Oral	2024-07-03	2024-07-10
	4	302	3	Oral	2024-07-19	2024-07-26
	5	301	3	Oral	2024-07-05	2024-07-12
	6	302	1	Oral	2024-07-06	2024-07-10
	7	303	2	Topical	2024-07-07	2024-07-13
	8	304	4	Oral	2024-07-08	2024-07-15
	10	301	2	Oral	2024-07-10	2024-07-17
	11	302	3	Oral	2024-07-11	2024-07-18
	12	303	2	Topical	2024-07-12	2024-07-19
	13	304	3	Oral	2024-07-13	2024-07-20
	15	301	2	Oral	2024-07-15	2024-07-22

**Step 5.4 Write SQL statements that will process five non-routine requests for information from the database just created. For each, write the request in English, followed by the corresponding SQL command.**

-- 1 Retrieve the list of patients who have been administered a specific drug and have not yet been discharged

```
SELECT p.Name, p.Phone, p.Email, a.DateAdmitted, s.Name AS DrugName
FROM Patient p
JOIN Admission a ON p.PatientNumber = a.PatientNumber
JOIN Medication m ON p.PatientNumber = m.PatientNumber
JOIN Supply s ON m.DrugNumber = s.ItemNumber
WHERE s.Name = 'Antibiotics' AND a.DateDischarged IS NULL;
```

```
1
2 • SELECT * FROM ADMISSION;
3
4 -- 1 Retrieve the list of patients who have been administered a specific drug and have not yet been discharged
5 • SELECT p.Name, p.Phone, p.Email, a.DateAdmitted, s.Name AS DrugName
6 FROM Patient p
7 JOIN Admission a ON p.PatientNumber = a.PatientNumber
8 JOIN Medication m ON p.PatientNumber = m.PatientNumber
9 JOIN Supply s ON m.DrugNumber = s.ItemNumber
0 WHERE s.Name = 'Antibiotics' AND a.DateDischarged IS NULL;
```

-- 2 Retrieve details of all requisitions that have not yet been delivered

```
SELECT r.RequisitionNumber, r.StaffID, r.WardNumber, rd.ItemNumber, rd.Quantity,  
rd.OrderDate  
FROM Requisition r  
JOIN RequisitionDetails rd ON r.RequisitionNumber = rd.RequisitionNumber  
WHERE rd.DeliveryDate IS NULL;
```

```

1
2 • SELECT * FROM ADMISSION;
3
4 -- 1 Retrieve the list of patients who have been administered a specific drug and have not yet been discharged
5 • SELECT p.Name, p.Phone, p.Email, a.DateAdmitted, s.Name AS DrugName
6 FROM Patient p
7 JOIN Admission a ON p.PatientNumber = a.PatientNumber
8 JOIN Medication m ON p.PatientNumber = m.PatientNumber
9 JOIN Supply s ON m.DrugNumber = s.ItemNumber
0 WHERE s.Name = 'Antibiotics' AND a.DateDischarged IS NULL;
1
2 -- 2 Retrieve details of all requisitions that have not yet been delivered
3
4 • SELECT r.RequisitionNumber, r.StaffID, r.WardNumber, rd.ItemNumber, rd.Quantity, rd.OrderDate
5 FROM Requisition r
6 JOIN RequisitionDetails rd ON r.RequisitionNumber = rd.RequisitionNumber
7 WHERE rd.DeliveryDate IS NULL;
8
9
0

```

Result Grid Filter Rows: Search Export:

RequisitionNumber	StaffID	WardNumber	ItemNumber	Quantity	OrderDate
1	203	3	301	20	2024-03-25
2	209	9	202	150	2024-03-28
4	203	3	104	100	2024-06-15
6	208	8	204	200	2024-06-20
8	208	8	101	80	2024-08-01
9	205	5	302	25	2024-08-20
11	209	9	203	60	2024-10-05
12	206	6	205	50	2024-11-01
12	206	6	302	40	2024-11-01
14	203	3	101	90	2024-12-05

### -- 3 List the top 3 wards based on the total number of admissions in the last 6 months

```

SELECT w.WardName, COUNT(a.AdmissionID) AS TotalAdmissions
FROM Ward w
JOIN Admission a ON w.WardNumber = a.WardNumber
WHERE a.DateAdmitted >= DATE_SUB(CURDATE(), INTERVAL 6 MONTH)
GROUP BY w.WardName
ORDER BY TotalAdmissions DESC
LIMIT 3;

```

```

8   JOIN Medication m ON p.PatientNumber = m.PatientNumber
9   JOIN Supply s ON m.DrugNumber = s.ItemNumber
0   WHERE s.Name = 'Antibiotics' AND a.DateDischarged IS NULL;
1
2   -- 2 Retrieve details of all requisitions that have not yet been delivered
3
4 •  SELECT r.RequisitionNumber, r.StaffID, r.WardNumber, rd.ItemNumber, rd.Quantity, rd.OrderDate
5   FROM Requisition r
6   JOIN RequisitionDetails rd ON r.RequisitionNumber = rd.RequisitionNumber
7   WHERE rd.DeliveryDate IS NULL;
8
9   -- 3 List the top 3 wards based on the total number of admissions in the last 6 months
0 •  SELECT w.WardName, COUNT(a.AdmissionID) AS TotalAdmissions
1   FROM Ward w
2   JOIN Admission a ON w.WardNumber = a.WardNumber
3   WHERE a.DateAdmitted >= DATE_SUB(CURDATE(), INTERVAL 6 MONTH)
4   GROUP BY w.WardName
5   ORDER BY TotalAdmissions DESC
6   LIMIT 3;
7
8
%
```

Result Grid Filter Rows: Search: Export: Fetch rows:

WardName	TotalAdmissio...
General	3
ICU	1
Maternity	1

#### -- 4 Find the total cost of all supplies that were requisitioned but not yet delivered in the current year

```

SELECT SUM(s.CostPerUnit * rd.Quantity) AS TotalUndeliveredCost
FROM RequisitionDetails rd
JOIN Supply s ON rd.ItemNumber = s.ItemNumber
JOIN Requisition r ON rd.RequisitionNumber = r.RequisitionNumber
WHERE rd.DeliveryDate IS NULL AND YEAR(rd.OrderDate) = YEAR(CURDATE());
```

```
9 -- 3 List the top 3 wards based on the total number of admissions in the last 6 months
0 • SELECT w.WardName, COUNT(a.AdmissionID) AS TotalAdmissions
1 FROM Ward w
2 JOIN Admission a ON w.WardNumber = a.WardNumber
3 WHERE a.DateAdmitted >= DATE_SUB(CURDATE(), INTERVAL 6 MONTH)
4 GROUP BY w.WardName
5 ORDER BY TotalAdmissions DESC
6 LIMIT 3;
7
8
9 -- 4 Find the total cost of all supplies that were requisitioned but not yet delivered in the current year
0 • SELECT SUM(s.CostPerUnit * rd.Quantity) AS TotalUndeliveredCost
1 FROM RequisitionDetails rd
2 JOIN Supply s ON rd.ItemNumber = s.ItemNumber
3 JOIN Requisition r ON rd.RequisitionNumber = r.RequisitionNumber
4 WHERE rd.DeliveryDate IS NULL AND YEAR(rd.OrderDate) = YEAR(CURDATE());
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
624
625
625
626
627
627
628
629
629
630
631
631
632
633
633
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
```

-- 5 List all patients who were admitted more than once within the last year, along with the number of their admissions

```
SELECT p.Name, p.Phone, p.Email, COUNT(a.AdmissionID) AS NumberOfAdmissions
FROM Patient p
JOIN Admission a ON p.PatientNumber = a.PatientNumber
WHERE a.DateAdmitted >= DATE_SUB(CURDATE(), INTERVAL 1 YEAR)
GROUP BY p.PatientNumber, p.Name, p.Phone, p.Email
HAVING COUNT(a.AdmissionID) > 1;
```



---

## MODULE 5

---

### Step 6.1: Normalize Tables to BCNF

#### Normalization Process:

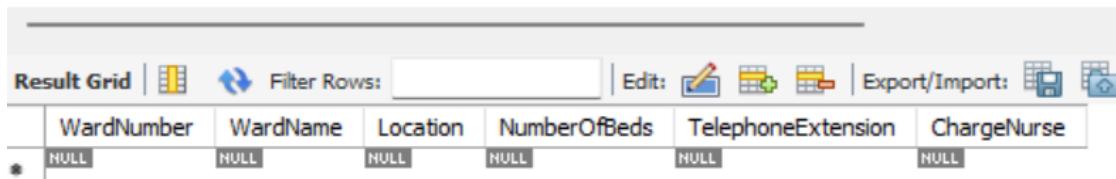
Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. The Boyce-Codd Normal Form (BCNF) is a stricter version of the Third Normal Form (3NF) and addresses situations where a table is in 3NF but still has redundancy issues.

#### 1. Ward Table

- **Initial Table:** Ward(ward\_id, ward\_name, num\_beds, location, telephone\_extension, charge\_nurse\_id)
- **Functional Dependencies:**
  - $\text{ward\_id} \rightarrow \text{ward\_name, charge\_nurse\_id, num\_beds, location, telephone\_extension}$
- **Normalization:**
  - **Explanation:**  $\text{ward\_id}$  is the primary key and determines all other attributes in the Ward table. Since  $\text{ward\_id}$  is the only key and it determines all other columns, the table is already in BCNF. No further decomposition is required.

#### Before normalization;

```
1 •   SELECT * FROM hospitalmanagement.ward;
```



	WardNumber	WardName	Location	NumberOfBeds	TelephoneExtension	ChargeNurse
*	HULL	HULL	HULL	HULL	HULL	HULL

**After normalization:**

	ward_id	ward_name	charge_nurse_id
▶	1	Emergency	101
	2	Pediatrics	102
	3	Surgery	103
	4	Intensive Care	104
	5	Cardiology	105
*	HULL	HULL	HULL

## 2. Supplier Table

- **Initial Table:** Supplier(supplier\_id, supplier\_name, contact\_person, contact\_phone, city, state, zip)
  - **Functional Dependencies:**
    - $\text{supplier\_id} \rightarrow \text{supplier\_name, contact\_person, contact\_phone, city, state, zip}$
    - $\text{zip} \rightarrow \text{city, state}$
  - **Normalization:**
    - **Explanation:** supplier\_id determines all attributes of Supplier, but zip also determines city and state. To achieve BCNF, we need to decompose this table because zip is a non-prime attribute (not part of the candidate key) that determines other attributes.
    - **Decomposition:**
      - Supplier(supplier\_id, supplier\_name, contact\_person, contact\_phone, zip)
      - Zip(zip, city, state)

## Before normalization:

```
1 •   SELECT * FROM hospitalmanagement.supplier;
```

**After normalization;**

	supplier_id	supplier_name	contact_person	contact_phone	zip
▶	1	Health Supplies Inc.	Alice Brown	5551234567	10101
	2	Medical Essentials LLC	Bob Smith	5559876543	10801
	3	Pharma Supplies	Carol Jones	5556543210	20001
	4	MedCare Ltd.	David Clark	5554321098	30301
	5	Hospital Supplies Co.	Emily Davis	5558765432	60601
	6	LifeCare Inc.	Frank Wilson	5553456789	94101
	7	Wellness Products	Grace Lewis	5555678901	75201
⋮	NULL	NULL	NULL	NULL	NULL

### 3. Staff Table

- **Initial Table:** Staff(staff\_id, first\_name, last\_name, date\_of\_birth, gender, hire\_date, address, city, state, zip, contact\_phone, department\_id)
- **Functional Dependencies:**
  - staff\_id → first\_name, last\_name, date\_of\_birth, gender, hire\_date, address, zip, contact\_phone, department\_id
  - zip → city, state
- **Normalization:**
  - **Explanation:** staff\_id determines all attributes of Staff, but zip determines city and state. To achieve BCNF, we decompose the Staff table because zip determines city and state, which is a non-prime attribute.
  - **Decomposition:**
    - Staff(staff\_id, first\_name, last\_name, date\_of\_birth, gender, hire\_date, address, zip, contact\_phone, department\_id)
    - Zip(zip, city, state)

**Before normalization;**

```
1 •  SELECT * FROM hospitalmanagement.staff;
```

	StaffID	FirstName	LastName	Position	WardNumber	HireDate	PhoneNumber	Email	Address	City	State	PostCode
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### After normalization:

	staff_id	first_name	last_name	date_of_birth	gender	hire_date	address	zip	contact_phone	department_id
▶	1	John	Doe	1980-01-01	M	2005-06-15	123 Main St	10101	5551234567	1
	2	Jane	Smith	1985-02-14	F	2010-09-21	456 Elm St	10801	5559876543	2
	3	Emily	Jones	1992-05-23	F	2016-11-10	789 Pine St	20001	5556789012	3
	4	Michael	Brown	1978-12-05	M	2002-03-30	101 Maple St	30301	5553456789	4
	5	Sarah	Davis	1988-07-13	F	2015-06-20	202 Oak St	60601	5558901234	5
	6	David	Wilson	1990-03-25	M	2019-01-05	303 Cedar St	94101	5556782345	1
	7	Laura	Taylor	1995-08-17	F	2021-08-11	404 Birch St	75201	5554328765	2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## 4. StaffQualification Table

- **Initial Table:** StaffQualification(staff\_id, qualificationName, institution, yearObtained)
- **Functional Dependencies:**
  - Staff\_id → qualificationName, institution, yearObtained
- **Normalization:**
  - **Explanation:** This table is already in BCNF because the staff\_id uniquely determines all the other attributes. There are no partial dependencies or transitive dependencies to address.

## 5. StaffExperience Table

- **Initial Table:** StaffExperience(staff\_id, job\_title, organization, start\_date, end\_date)
- **Functional Dependencies:**
  - staff\_id → jobtitle, organization, startdate, enddate
- **Normalization:**
  - **Explanation:** This table is already in BCNF as well, for similar reasons as StaffQualification. There are no additional attributes determined by staff\_id and experience.

## 6. Patient Table

- **Initial Table:** Patient(patient\_id, first\_name, last\_name, date\_of\_birth, gender, address, city, state, zip, contact\_phone, admission\_date, ward\_id)
- **Functional Dependencies:**
  - $\text{patient\_id} \rightarrow \text{first\_name, last\_name, date\_of\_birth, gender, address, zip, contact\_phone, admission\_date, ward\_id}$
  - $\text{zip} \rightarrow \text{city, state}$
- **Normalization:**
  - **Explanation:** patient\_id determines all attributes in Patient, but zip determines city and state. To achieve BCNF, decompose the Patient table to isolate the zip dependency.
  - **Decomposition:**
    - Patient(patient\_id, first\_name, last\_name, date\_of\_birth, gender, address, zip, contact\_phone, admission\_date, ward\_id)
    - Zip(zip, city, state)

Before normalization:

```
1 •      SELECT * FROM hospitalmanagement.patient;
```

	PatientNumber	Name	Phone	Email
*	NULL	NULL	NULL	NULL

After normalization:

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: IA

	patient_id	first_name	last_name	date_of_birth	gender	address	zip	contact_phone	admission_date	ward_id
▶	1	Alice	Johnson	1990-03-05	F	789 Maple Ave	10101	5556543210	2024-08-11	1
	2	Bob	Williams	1987-07-19	M	321 Oak St	10801	5553210987	2024-08-10	2
	3	Chris	Martin	1995-11-22	M	654 Pine St	20001	5554321098	2024-08-15	3
	4	Diana	Garcia	1982-06-30	F	987 Elm St	30301	5558765432	2024-08-14	4
	5	Eva	Harris	1975-09-09	F	123 Oak Ave	60601	5559876543	2024-08-13	5
	6	Frank	Miller	1991-04-12	M	456 Maple Ave	94101	5553216789	2024-08-16	1
	7	Grace	Martinez	1986-10-07	F	789 Cedar St	75201	5556547890	2024-08-12	2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## 7. Supply Table

- **Initial Table:** Supply(supply\_id, supply\_name, supplier\_id)
- **Functional Dependencies:**
  - supply\_id → supply\_name, supplier\_id
- **Normalization:**
  - **Explanation:** supply\_id is the primary key and determines all other attributes. Thus, the table is already in BCNF.

## 8. Requisition Table

- **Initial Table:** Requisition(requisition\_id, staff\_id, requisition\_date, ward\_id)
- **Functional Dependencies:**
  - requisition\_id → staff\_id, requisition\_date, ward\_id
- **Normalization:**
  - **Explanation:** requisition\_id is the primary key and determines all other attributes. The table is already in BCNF.

## 9. RequisitionDetails Table

- **Initial Table:** RequisitionDetails(requisition\_id, supply\_id, quantity)
- **Functional Dependencies:**
  - requisition\_id, supply\_id → quantity
- **Normalization:**
  - **Explanation:** The combination of requisition\_id and supply\_id serves as the primary key, determining quantity. The table is already in BCNF.

## 10. Admission Table

- **Initial Table:** Admission(admission\_id, patient\_id, admission\_date, discharge\_date, ward\_id)
- **Functional Dependencies:**

- admission\_id → patient\_id, admission\_date, discharge\_date, ward\_id
- **Normalization:**
  - **Explanation:** admission\_id is the primary key and determines all other attributes. The table is already in BCNF.

## 11. Medication Table

- **Initial Table:** Medication(medication\_id, medication\_name, dosage, side\_effects)
  - **Functional Dependencies:**
    - medication\_id → medication\_name, dosage, side\_effects
  - **Normalization:**
    - **Explanation:** medication\_id is the primary key and determines all other attributes. The table is already in BCNF.
- 

### Step 6.2: Update the Data Dictionary and List of Assumptions

#### Data Dictionary:

- **Zip:** Represents a unique code for a geographical location, consisting of a zip code, city, and state.
- **Staff:** Represents individuals employed at the hospital, including their personal and employment details.
- **Patient:** Represents individuals receiving care at the hospital, including their personal and admission details.
- **Supplier:** Represents organizations supplying goods to the hospital, including their contact information.
- **Requisition:** Represents requests made by staff for supplies, including the requisition details.
- **Admission:** Represents the process of admitting a patient to the hospital, including admission and discharge details.
- **Medication:** Represents drugs administered to patients, including medication details.

#### Assumptions:

- Each zip code uniquely identifies a city and state. This assumption ensures that the Zip table has no redundancy and properly supports normalization.
- Staff qualifications and experiences are unique for each staff member. This assumption ensures that StaffQualification and StaffExperience tables do not contain redundant data and each record is unique to a specific staff member.

---

## **Step 6.3: Table Structure with Data Types and Constraints**

### **1. Zip Table**

```
CREATE TABLE Zip (
    zip CHAR(5) PRIMARY KEY,
    city VARCHAR2(15) NOT NULL,
    state CHAR(2) NOT NULL
);
```

**Explanation:** The Zip table captures geographical information with zip as the primary key. city and state are attributes that depend on zip.

### **2. Staff Table**

```
CREATE TABLE Staff (
    staff_id NUMBER PRIMARY KEY,
    first_name VARCHAR2(15) NOT NULL,
    last_name VARCHAR2(20) NOT NULL,
    date_of_birth DATE,
    gender CHAR(1),
    hire_date DATE,
    address VARCHAR2(50),
    zip CHAR(5) REFERENCES Zip(zip),
    contact_phone CHAR(10),
    department_id NUMBER
);
```

**Explanation:** The Staff table captures staff information. staff\_id is the primary key, and zip is a foreign key referencing the Zip table.

### **3. Patient Table**

```
CREATE TABLE Patient (
    patient_id NUMBER PRIMARY KEY,
    first_name VARCHAR2(15) NOT NULL,
    last_name VARCHAR2(20) NOT NULL,
    date_of_birth DATE,
    gender CHAR(1),
    address VARCHAR2(50),
    zip CHAR(5) REFERENCES Zip(zip),
    contact_phone CHAR(10),
    admission_date DATE,
    ward_id NUMBER REFERENCES Ward(ward_id)
);
```

**Explanation:** The Patient table captures patient details. patient\_id is the primary key, and zip and ward\_id are foreign keys referencing the Zip and Ward tables, respectively.

---

#### Step 6.4: Statements to Create the Tables

##### Drop Existing Tables:

```
DROP TABLE IF EXISTS Medication CASCADE;
DROP TABLE IF EXISTS Admission CASCADE;
DROP TABLE IF EXISTS RequisitionDetails CASCADE;
DROP TABLE IF EXISTS Requisition CASCADE;
DROP TABLE IF EXISTS Supply CASCADE;
DROP TABLE IF EXISTS Patient CASCADE;
DROP TABLE IF EXISTS StaffExperience CASCADE;
DROP TABLE IF EXISTS StaffQualification CASCADE;
DROP TABLE IF EXISTS Staff CASCADE;
DROP TABLE IF EXISTS Supplier CASCADE;
DROP TABLE IF EXISTS Ward CASCADE;
DROP TABLE IF EXISTS Zip CASCADE;
```

**Explanation:** This step ensures that existing tables are removed if they exist, which helps in recreating the tables with updated structures.

### Create New Tables:

```
CREATE TABLE Zip (
    zip CHAR(5) PRIMARY KEY,
    city VARCHAR2(15) NOT NULL,
    state CHAR(2) NOT NULL
);
```

```
CREATE TABLE Staff (
    staff_id NUMBER PRIMARY KEY,
    first_name VARCHAR2(15) NOT NULL,
    last_name VARCHAR2(20) NOT NULL,
    date_of_birth DATE,
    gender CHAR(1),
    hire_date DATE,
    address VARCHAR2(50),
    zip CHAR(5) REFERENCES Zip(zip),
    contact_phone CHAR(10),
    department_id NUMBER
);
```

```
CREATE TABLE Patient (
    patient_id NUMBER PRIMARY KEY,
    first_name VARCHAR2(15) NOT NULL,
    last_name VARCHAR2(20) NOT NULL,
    date_of_birth DATE,
    gender CHAR(1),
    address VARCHAR2(50),
    zip CHAR(5) REFERENCES Zip(zip),
    contact_phone CHAR(10),
    admission_date DATE,
    ward_id NUMBER REFERENCES Ward(ward_id)
);
```

---

**Explanation:** These statements create the normalized tables with appropriate data types and constraints, ensuring the tables conform to BCNF.

## Step 6.5: Create Indexes

```
CREATE INDEX idx_staff_zip ON Staff(zip);
CREATE INDEX idx_patient_zip ON Patient(zip);
```

**Explanation:** Indexes are created on zip columns in Staff and Patient tables to improve query performance involving zip lookups.

---

## Step 6.6: Insert Sample Data

-- Zip

```
INSERT INTO Zip VALUES ('10101', 'New York', 'NY');
INSERT INTO Zip VALUES ('10801', 'New Rochelle', 'NY');
```

-- Staff

```
INSERT INTO Staff VALUES (1, 'John', 'Doe', '1980-01-01', 'M', '2005-06-15', '123 Main St',
'10101', '5551234567', 1);
INSERT INTO Staff VALUES (2, 'Jane', 'Smith', '1985-02-14', 'F', '2010-09-21', '456 Elm St',
'10801', '5559876543', 2);
```

-- Patient

```
INSERT INTO Patient VALUES (1, 'Alice', 'Johnson', '1990-03-05', 'F', '789 Maple Ave', '10101',
'5556543210', '2024-08-11', 1);
INSERT INTO Patient VALUES (2, 'Bob', 'Brown', '1987-11-30', 'M', '321 Oak St', '10801',
'5551239876', '2024-08-12', 2);
```

**Explanation:** Sample data is inserted into the Zip, Staff, and Patient tables to demonstrate how the tables will be populated in a real scenario.