

# Project 1 Group 2

## **LINK TO GITHUB:**

Julia Gild—@JuliaGild

John Neely— @NeelyJohn

Valerie Tran— @ValerieTran

Jack Glawatz— @jackglawatz

Oluwabukola Ogundare- @RachaelOgundare

## **Table of Contents**

Overview.....	3
Prompt.....	3
Introduction to My Business.....	3
Overview of Our Operations.....	3
List of Entities:.....	6
Data Model:.....	9
Data Model Description:.....	9
Problem Description:.....	9
Data Dictionary:.....	10
Billing.....	10
Insurance Claim.....	10
Insurance Provider.....	11
Medicine.....	12
Patient.....	12
Payment.....	13
Prescription.....	13
Receipt of Storage.....	13
Schedule.....	14
Staff.....	14
Storage.....	15
Visit.....	15
SQL Queries:.....	22
Complex.....	23
C1.....	23
C2.....	24
C3 –.....	25
C4 –.....	25
C5 –.....	27

C6 – Good.....	28
C7 -.....	30
C8 - Good.....	30
<b>Simple:.....</b>	<b>31</b>
S1—.....	31
S2—.....	31
S3—.....	32

## Overview

### Prompt:

Pretend you are the owner/operator of an emergency healthcare clinic needing to build a relational database. You hired some students from the MIST 4610 class at the University of Georgia to create the database for you. They need to know more about your organization to identify which entities, attributes, and relationships are important for you. Start by describing your business as a real client

### Introduction to My Business

Hello, everyone! I run an **emergency healthcare clinic** that provides walk-in urgent care services to the local community. Our clinic operates 7 days a week and handles a wide range of non-life-threatening medical needs. Since patient care is our top priority, we aim to streamline operations with a new relational database that ensures efficiency and accuracy in managing our day-to-day processes.

### Overview of Our Operations

Here are some details about how we operate that may help you identify the key entities, attributes, and relationships for our database:

#### **1. Billing :**

- We handle billing for services provided during visits.
- Payment can be made via insurance, out-of-pocket, or a combination.
- Bills include a breakdown of services provided (e.g., X-rays, blood tests) and their costs.

#### **2. Insurance Claim**

- If a patient has insurance coverage, an insurance claim is filed to cover part or all of the bill.
- Claims are associated with a provider and a specific bill for reimbursement purposes.
- The claim includes the amount requested, the date it was filed, and the services covered under the claim.
- Once processed, the claim updates the Billing entity based on the insurance reimbursement.
- 

#### **3. Insurance Providers:**

- We work with multiple insurance companies.
- For each insurance provider, we store their name, contact details, policy types, and a list of covered services.

#### **4. Medicine**

- If a patient requires medication, a prescription is issued during the visit.
- Each medicine is linked to a specific prescription, detailing the dosage, manufacturer, and instructions.
- The medicine is linked to the patient who needs it and the prescription specifying the dosage and frequency.

#### **5. Patients:**

- We see hundreds of patients weekly, both recurring and new.
- We collect and store information like patient names, contact details, date of birth, insurance details, medical history, and their visit records.
- Each patient is assigned a unique patient ID.

#### **6. Payment**

- Payments are made by the patient or their insurance to cover the bill.
- Each payment links to a specific bill and records the amount paid, payment method (e.g., insurance, cash, credit card), and payment status.
- Payments may cover all or part of a bill, and partial payments can be tracked

#### **7. Prescriptions:**

- If a patient requires medication, the doctor prescribes it during the visit.
- Each prescription is linked to the patient ID, visit ID, prescribed medication, dosage, and instructions.

#### **8. Receipt of Storage**

- If medicine is received for storage, a receipt of storage is generated to track the intake.
- Each receipt links the medicine to a specific storage location, recording the amount of medication received.

#### **9. Schedule**

- Schedule tracks the assignments of doctors and nurses to specific rooms for patient care.
- Each schedule includes the room ID, the staff involved, their on-call status, and any shift notes regarding patient care.

#### **10. Staff:**

- Our team includes doctors, nurses, lab technicians, and administrative staff.
- Staff members have unique IDs, and we track their roles, schedules, credentials, and contact information.
- Each patient is seen by at least one doctor or nurse during their visit.

#### **11. Visits:**

- While most of our patients are walk-ins, some schedule appointments.

- Each visit or appointment generates a unique record that includes the patient ID, staff IDs involved, date and time of the visit, reason for visit, and the outcome (e.g., treatment provided, referral).

## **12. Storage:**

- We offer various medical services, such as diagnostic tests, minor procedures, and prescription refills.
- Each service has a code, name, description, and cost.
- Our clinic also maintains an inventory of medical supplies (e.g., syringes, medications) that must be tracked to avoid shortages.

## **Database Goals**

The database should:

- **Track patient records** efficiently, ensuring data accuracy and compliance with privacy regulations.
- **Manage staff schedules** and link their interactions to specific patient visits.
- **Integrate billing and payment** data to provide clear, detailed invoices.
- **Support reporting** for inventory usage, service performance, and patient trends.
- Enable easy **search and retrieval of information** for audits, quality assurance, and decision-making.

**List of Entities:**

**1. Billing**

- billID, billDescription, billAmt, patientID(FK), providerID(FK), statusID

**2. Insurance Claim**

- claimID, claimPatient, claimDescription, claimDate, claimAmount, providerID(FK), billingID(FK)

**3. Insurance Provider**

- providerID(FK), providerName

**4. Medicine**

- medicineID, medicineName, recommendedMedicineDosage, medicineManufacturer, prescriptionID(FK), patientID(FK)

**5. Patient**

- patientID, patientName, patientHistory, patientBday, patientGender, patientPhone, providerID(FK)s

**6. Payment**

- paymentID(PK), paymentAmount, paymentMethod, billID(FK)

**7. Prescription**

- prescriptionID(PK), patientID(FK), dosageAmt, frequency, roomID, staffID

**8. ReceiptofStorage**

- receiptID, medecineID,storageID,dosageRecieved

**9. Schedule**

- roomID, doctorID, nurseID, onCallStatus, shiftNotes

**10. Staff**

- staffID, staffName, staffRole, staffNumber, staffDOB, staffGender, billing, roomID

**11. Storage**

- storageID,storageName,storageCapacity

**12. Visit**

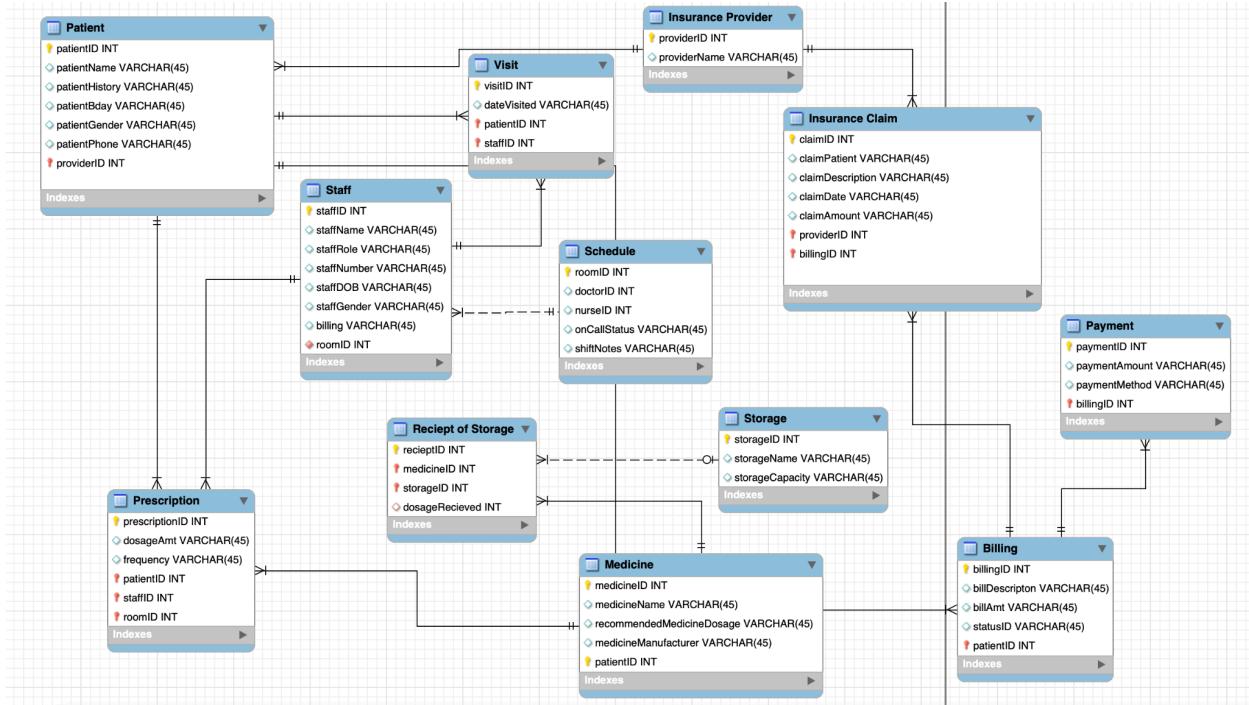
- visitID(PK), patientID, dateVisit, staffID(FK)

**Relationships:**

1. Patient - Insurance Provider (M:1)
  - Many Patients are associated with one Insurance Provider.
  - One Insurance Provider can cover multiple Patients.
2. Patient - Visit (1:M)
  - One Patient can have multiple Visits, but each Visit belongs to only one Patient.
3. Visit - Staff (M:1)
  - Many Visits are attended by one Staff member.
  - A single Staff member can be responsible for multiple Visits.
4. Patient - Prescription (1:M)
  - One Patient can receive multiple Prescriptions, but each Prescription is assigned to a single Patient.
5. Prescription - Staff (M:1)

- Many Prescriptions are written by one Staff member.
  - A single Staff member can prescribe multiple Prescriptions.
6. Prescription - Room (M:1)
- Many Prescriptions are linked to a single Room.
  - A Room may have multiple prescriptions associated with it.
7. Patient - Billing (1:M)
- One Patient can have multiple Billing records, but each Billing entry is associated with only one Patient.
8. Billing - Insurance Claim (1:M)
- One Billing record can be linked to multiple Insurance Claims, but each Insurance Claim corresponds to only one Billing entry.
9. Billing - Payment (1:M)
- One Billing entry can have multiple Payments, but each Payment is linked to a single Billing record.
10. Insurance Claim - Insurance Provider (M:1)
- Many Insurance Claims are processed by one Insurance Provider.
  - A single Insurance Provider handles multiple claims.
11. Storage - Medicine (1:M)
- One Storage unit can hold multiple Medicines, but each Medicine belongs to only one Storage location.
12. Storage - Receipt of Storage (1:M)
- One Storage unit can be linked to multiple Receipts of Storage, but each Receipt of Storage is tied to only one Storage location.
13. Medicine - Receipt of Storage (M:1)
- Many Receipts of Storage are linked to a single Medicine.
  - A Medicine can be restocked multiple times, but each receipt belongs to a single medicine type.
14. Schedule - Staff (M:N)
- One Schedule entry can include multiple Staff members (doctors/nurses), and each Staff member can appear in multiple Schedules

**Data Model:**



## Data Model Description:

This data model represents a hospital management system, detailing the relationships between patients, staff, medical records, billing, insurance, prescriptions, and storage. Patients visit the hospital, where they are attended to by staff members such as doctors and nurses. Their medical history, prescriptions, and visits are recorded in the system. Billing and insurance claims are tracked, ensuring proper financial management. The hospital also maintains an inventory of medicines and storage facilities, with a system in place for tracking the receipt and distribution of medical supplies. This model helps streamline hospital operations by organizing patient care, financial transactions, and resource management efficiently.

## Problem Description:

A multi-sector medical facility is currently in need of a new healthcare management system. Currently, the facility is running into record organization issues that ultimately result in billing delays, overlapping patient visits, and lack of storage capacity. The facility is reaching out to the students from the MIST 4610 class at UGA to create a database system that streamlines operations, improves efficiency, and provides accurate data tracking. This system should include the management of patients, medical staff members, appointments and schedules, billing requests, insurance claims, prescriptions and medications, and storage. This database will, in turn, optimize the organization's daily operations, improve patient care, and increase workflow efficiency.

## **Data Dictionary:**

### **Billing**

Column Name	Description	Data Type	Size	Format	Key
billingID	Unique ID for each billing record	INT	45	-	Primary
billDescription	Description of the bill	VARCHAR	45	-	-
billAmt	Amount charged in the bill	VARCHAR	45	-	-
statusID	Status of the billing process	VARCHAR	45	-	-
patientID	Refers to the patient associated with the bill	INT	45	-	Foreign

### **Insurance Claim**

Column Name	Description	Data Type	Size	Format	Key
claimID	Unique identifier for insurance claim	INT	45	-	Primary
claimPatient	Patient associated with the claim	VARCHAR	45	-	-

claimDescription	Description of the claim	VARCHAR	45	-	-
claimDate	Date the claim was filed	VARCHAR	45	MMDDYY YY	-
claimAmount	Amount of the insurance claim	VARCHAR	45	-	-
providerID	Foreign key linking to Insurance Provider	INT	45	-	Foreign
billingID	Foreign key linking to Billing table	INT	45	-	Foreign

## Insurance Provider

Column Name	Description	Data Type	Size	Format	Key
providerID	Unique identifier for each insurance provider	INT	45	-	Primary
providerName	Name of the Insurance Company	VARCHAR	45	-	-

## Medicine

Column Name	Description	Data Type	Size	Format	Key
medicineID	Unique ID for each medicine	INT	45	-	Primary
medicineName	Name of the medicine	VARCHAR	45	-	-
recommendedMedicineDosage	Recommended dosage for medicine	VARCHAR	45	-	-
medicineManufacturer	Manufacturer of the medicine	VARCHAR	45	-	-
patientID	Refers to the patient associated with the medicine	INT	45	-	Foreign

## Patient

Column Name	Description	Data Type	Size	Format	Key
patientID	Unique ID for each patient	INT	45	-	Primary
patientName	Name of the patient	VARCHAR	45	-	-
patientHistory	Medical history of the patient	VARCHAR	45	-	-
patientBday	Date of birth of the patient	VARCHAR	45	MM/DD/YYYY	-
patientGender	Gender of the patient	VARCHAR	45	-	-
patientPhone	Contact number of the patient	VARCHAR	45	(999)999-9999	-
providerID	Refers to the insurance provider	INT	45	-	Foreign

## Payment

Column Name	Description	Data Type	Size	Format	Key
paymentID	Unique ID for each payment	INT	45	-	Primary
paymentAmount	Amount paid	VARCHAR	45	-	-
paymentMethod	Method of payment (Credit, Cash, etc.)	VARCHAR	45	-	-
billingID	Refers to the related billing record	INT	45	-	Foreign

## Prescription

Column Name	Description	Data Type	Size	Format	Key
prescriptionID	Unique ID for each prescription	INT	45	-	Primary
dosageAmt	Medicine amount per dose	VARCHAR	45	-	-
frequency	How often medicine is taken	VARCHAR	45	-	-
patientID	Patient receiving the prescription	INT	45	-	Foreign
staffID	Staff handling the prescription	INT	45	-	Foreign
roomID	Assigned hospital room	INT	45	-	Foreign
patientID	Patient receiving the prescription	INT	45		Foreign

## Receipt of Storage

Column Name	Description	Data Type	Size	Format	Key
receiptID	Unique identifier for each receipt, used to track medicine transactions	INT	45	-	Primary
medicineID	Foreign key linking to Medicine	INT	45	-	Foreign

storageID	Foreign key linking to storage	INT	45	-	Foreign
dosageReceived	Quantity of medicine received in a transaction	INT	45	-	-

## Schedule

Column Name	Description	Data Type	Size	Format	Key
roomID	Unique identifier for the schedule record	INT	45	-	Primary
doctorID	Identifier for the doctor's card	VARCHAR	45	-	-
nurseID	Identifier for the nurse's selection	VARCHAR	45	-	-
onCallStatus	Status of the on-call schedule	VARCHAR	45	-	-
shiftNotes	Notes for the shift	VARCHAR	45	-	-

## Staff

Column Name	Description	Data Type	Size	Format	Key
staffID	Unique identifier for staff	INT	45	-	Primary
staffName	Full name of the staff	VARCHAR	45	-	-
staffRole	Role/designation of the staff member	VARCHAR	45	-	-
staffNumber	Contact number of the staff	VARCHAR	45	(999)99 9-9999	-
staffDOB	Date of birth of the staff	VARCHAR	45	MM/DD /YYYY	-
staffGender	Gender of the staff	VARCHAR	45	-	-
billing	Billing details of the staff	VARCHAR	45	-	-

roomID	Foreign key linking to Schedule table	INT	45	-	Foreign
--------	---------------------------------------	-----	----	---	---------

## Storage

Column Name	Description	Data Type	Size	Format	Key
storageID	Unique identifier for storage	INT	45	-	Primary
storageName	Name of the storage location	VARCHAR	45	-	-
storageCapacity	Capacity of the storage	VARCHAR	45	-	-

## Visit

Column Name	Description	Data Type	Size	Format	Key
visitID	Unique identifier for each visit	INT	45	-	Primary
dateVisited	Date of the patient visit	VARCHAR	45	-	-
patientID	Foreign key to the patient table	INT	45	-	Foreign
staffID	Foreign key linking to the Staff table	INT	45	-	Foreign



```
1 • USE al_Group_47114_G2;
2
3 • CREATE TABLE Patient (
4     patientID INT,
5     patientName VARCHAR(45),
6     patientHistory VARCHAR(255),
7     patientBday VARCHAR(45),
8     patientGender VARCHAR(45),
9     patientPhone VARCHAR(45),
10    providerID INT,
11    PRIMARY KEY(patientID),
12    FOREIGN KEY(providerID) REFERENCES InsuranceProvider(providerID)
13 );
14
15 • CREATE TABLE Visit (
16     visitID INT,
17     dateVisited VARCHAR(45),
18     patientID INT,
19     staffID INT,
20     PRIMARY KEY(visitID),
21     FOREIGN KEY(patientID) REFERENCES Patient(patientID),
22     FOREIGN KEY(staffID) REFERENCES Staff(staffID)
23 );
24
25 • CREATE TABLE InsuranceProvider (
26     providerID INT,
27     providerName VARCHAR(45),
28     PRIMARY KEY(providerID)
29 );
30
```

```
31 • Ⓜ CREATE TABLE InsuranceClaim (
32     claimID INT,
33     claimPatient INT,
34     claimDescription VARCHAR(255),
35     claimDate VARCHAR(45),
36     claimAmount VARCHAR(45),
37     providerID INT,
38     billingID INT,
39     PRIMARY KEY(claimID),
40     FOREIGN KEY(claimPatient) REFERENCES Patient(patientID),
41     FOREIGN KEY(providerID) REFERENCES InsuranceProvider(providerID),
42     FOREIGN KEY(billingID) REFERENCES Billing(billingID)
43 );
44
45 • Ⓜ CREATE TABLE Staff (
46     staffID INT,
47     staffName VARCHAR(45),
48     staffRole VARCHAR(45),
49     staffNumber VARCHAR(45),
50     staffDOB VARCHAR(45),
51     staffGender VARCHAR(45),
52     billing VARCHAR(45),
53     roomID INT,
54     PRIMARY KEY(staffID),
55     FOREIGN KEY(roomID) REFERENCES Schedule(roomID)
56 );
57
```

```
58 • Ⓛ CREATE TABLE Schedule (
59     roomID INT,
60     doctorID INT,
61     nurseID INT,
62     onCallStatus VARCHAR(45),
63     shiftNotes VARCHAR(255),
64     PRIMARY KEY(roomID),
65     FOREIGN KEY(doctorID) REFERENCES Staff(staffID),
66     FOREIGN KEY(nurseID) REFERENCES Staff(staffID)
67 );
68
69 • Ⓛ CREATE TABLE Prescription (
70     prescriptionID INT,
71     dosageAmt VARCHAR(45),
72     frequency VARCHAR(45),
73     patientID INT,
74     staffID INT,
75     roomID INT,
76     PRIMARY KEY(prescriptionID),
77     FOREIGN KEY(patientID) REFERENCES Patient(patientID),
78     FOREIGN KEY(staffID) REFERENCES Staff(staffID),
79     FOREIGN KEY(roomID) REFERENCES Schedule(roomID)
80 );
```

```
82 • Ⓛ CREATE TABLE ReceiptOfStorage (
83     receiptID INT,
84     medicineID INT,
85     storageID INT,
86     dosageReceived INT,
87     PRIMARY KEY(receiptID),
88     FOREIGN KEY(medicineID) REFERENCES Medicine(medicineID),
89     FOREIGN KEY(storageID) REFERENCES Storage(storageID)
90 );
91
92 • Ⓛ CREATE TABLE Storage (
93     storageID INT,
94     storageName VARCHAR(45),
95     storageCapacity VARCHAR(45),
96     PRIMARY KEY(storageID)
97 );
98
99 • Ⓛ CREATE TABLE Payment (
100    paymentID INT,
101    paymentAmount VARCHAR(45),
102    paymentMethod VARCHAR(45),
103    billingID INT,
104    PRIMARY KEY(paymentID),
105    FOREIGN KEY(billingID) REFERENCES Billing(billingID)
106 );
```

```
108 • Ⓛ CREATE TABLE Medicine (
109     medicineID INT,
110     medicineName VARCHAR(45),
111     recommendedMedicineDosage VARCHAR(45),
112     medicineManufacturer VARCHAR(45),
113     prescriptionID INT,
114     patientID INT,
115     PRIMARY KEY(medicineID),
116     FOREIGN KEY(prescriptionID) REFERENCES Prescription(prescriptionID),
117     FOREIGN KEY(patientID) REFERENCES Patient(patientID)
118 );
119
120 • Ⓛ CREATE TABLE Billing (
121     billingID INT,
122     billDescription VARCHAR(255),
123     billAmt VARCHAR(45),
124     statusID INT,
125     patientID INT,
126     PRIMARY KEY(billingID),
127     FOREIGN KEY(patientID) REFERENCES Patient(patientID)
128 );
```

## SQL Queries:

	Query 1	Query 2	Query 3	Query 4	Query 5	Query 6	Query 7	Query 8	Query 9	Query 10
Multiple Table Join	X	X	X	X	X	X			X	
Subquery						X		X	X	
Correlated Subquery						X		X	X	
GROUP BY	X	X	X	X	X					X
GROUP BY with HAVING	X		X	X	X					
WHERE				X		X	X	X	X	
Calculated Functions(i.e, AVG, SUM)		X	X	X	X	X			X	
REGEXP							X			
NOT EXISTS								X		

## Complex

C1—

- What is the total billing amount for each patient, and how many payments have they made?

```
3    ## What is the total billing amount for each patient, and how many payments have they made?
4
5 • SELECT patientName, SUM(billAmt), SUM(paymentAmount),
6      SUM(billAmt)-SUM(paymentAmount) AS 'Remaining Balance', COUNT(paymentID)
7      FROM Patient
8      JOIN Billing on Billing.patientID = Patient.PatientID
9      JOIN Payment on Payment.billingID = Billing.billingID
10     GROUP BY patientName
11     HAVING SUM(billAmt)-SUM(paymentAmount) > 0
12     ORDER BY SUM(billAmt) DESC;
13
```

Result Grid				
patientName	SUM(billAmt)	SUM(paymentAmount)	Remaining Balance	COUNT(paymentID)
Dennie Napthine	40710	37860.00	2850	12
Roxana Galero	34232	16788.00	17444	8
Meriel Inchbald	31418	27438.00	3980	9
Jedidiah Pantlin	28525	17535.00	10990	7
Basilius Marsham	23801	18097.00	5704	7
Vi Shearman	23035	21445.00	1590	7
Javier Caswall	22488	21371.00	1117	10

- This query provides valuable insights into patient billing and payment behavior by calculating the total billing amount for each patient and tracking the number of payments they have made. The total billing amount reveals how much a patient has been charged overall, giving the hospital a clear view of their financial responsibility. Meanwhile, the number of payments indicates how actively a patient has been addressing their bills. This information is particularly useful for identifying patients who may be at financial risk, especially if they have a high total billing amount but few payments recorded. Additionally, tracking payment patterns can help the hospital predict cash flow, manage collections more effectively, and prioritize follow-up actions for patients who may require financial guidance or support.

C2

2. Display which storage facilities are nearing capacity.

```
3      ## 4. Which storage facilities are nearing capacity
4
5 •  SELECT storageName, storageCapacity, COUNT(receiptID)
6    FROM Storage
7    JOIN ReceiptOfStorage ON ReceiptOfStorage.storageID = Storage.storageID
8    GROUP BY storageName, storageCapacity
9    ORDER BY storageCapacity ASC;
10
11
```

Result Grid		
storageName	storageCapacity	COUNT(receiptID)
ICU Medical Storage	150	3
Oncology Drug Storage	180	4
Emergency Medicine Cabinet	200	4
Pediatrics Medicine Room	250	3
Surgical Supplies Room	300	4
Orthopedic Supplies Room	350	4
Cardiology Equipment Room	400	3
Radiology Storage	450	3
Pharmacy Storage A	500	3
General Storage B	600	4

- This query identifies which storage facilities are approaching their maximum capacity, providing crucial information for inventory management. By highlighting facilities that are nearly full, the hospital can take proactive measures such as redistributing supplies, adjusting delivery schedules, or planning for additional storage space. Monitoring storage levels is essential to ensure that medical supplies, medications, and equipment are always available without risking overstocking or space constraints. Identifying these trends early helps maintain operational efficiency and prevents potential disruptions in patient care due to supply shortages

C3

3. List Patients and Payment Details if they have made more than one payment

```
15
16 •   SELECT
17     p.patientID,
18     p.patientName,
19     COUNT(py.paymentID) AS totalPayments
20   FROM
21     Patient p
22   JOIN
23     Billing b ON p.patientID = b.patientID
24   JOIN
25     Payment py ON b.billingID = py.billingID
26   GROUP BY
27     p.patientID, p.patientName
28   HAVING
29     totalPayments > 1;
30
31
```

The screenshot shows a database query results grid. At the top, there are navigation controls: a zoom icon (100%), a dropdown menu, the time (1:16), and a scroll bar. Below the controls is a toolbar with buttons for 'Result Grid' (selected), 'Filter Rows', 'Search' (with a magnifying glass icon), and 'Export' (with a document icon). The main area displays a table with three columns: 'patientID', 'patientName', and 'totalPayments'. The data rows are:

patientID	patientName	totalPayments
53	Bryn Chaudret	3
131	Redford Irnys	9
94	Les Spours	2
188	Carree Copner	2

- This query identifies patients who have made more than one payment. The goal is to help the healthcare system track patients with multiple transactions, which may indicate ongoing treatments, installment plans, or potential billing issues. This information supports better financial management, follow-up strategies, and ensures patients receive appropriate care without financial oversight.

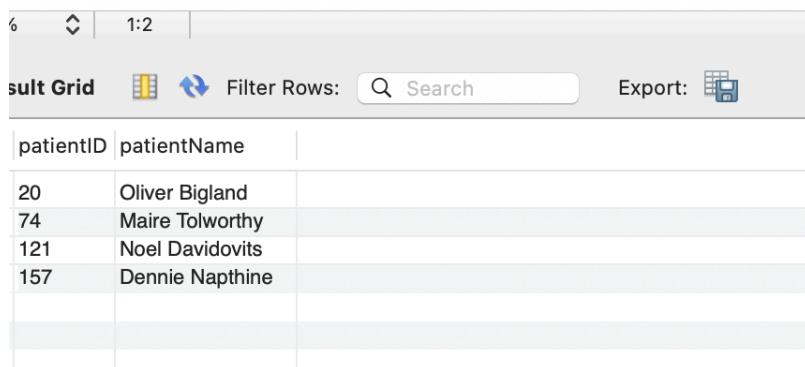
C4

4. List the patient's name and ID number who are identified as High-Risk Patients (Unpaid Bills + Frequent Visits). To be considered High-Risk, patients have to have both 3+ unpaid bills and 4+ visits.

```

1 • USE al_Group_47114_G2;
2 • SELECT
3     p.patientID,
4     p.patientName
5 FROM
6     Patient p
7 JOIN
8     Billing b ON p.patientID = b.patientID
9 JOIN
10    Visit v ON p.patientID = v.patientID
11 WHERE
12     b.statusID = 'Pending'
13 GROUP BY
14     p.patientID, p.patientName
15 HAVING
16     COUNT(DISTINCT b.billingID) >= 3
17     AND COUNT(DISTINCT v.visitID) >= 4;

```



The screenshot shows a database query results grid. The grid has two columns: 'patientID' and 'patientName'. The data rows are:

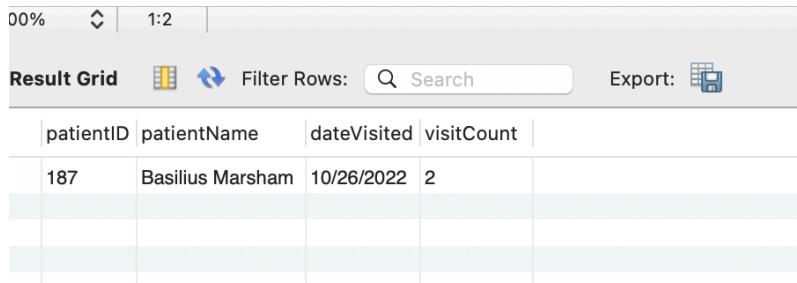
patientID	patientName
20	Oliver Bigland
74	Maire Tolworthy
121	Noel Davidovits
157	Dennie Napthine

- This SQL query helps the hospital identify high-risk patients by pinpointing those with 3 or more unpaid bills and 4 or more visits. Filtering for unpaid bills ensures that only patients with outstanding payments are considered, while grouping the data by patient and counting their records highlights those frequently visiting the hospital. This information is crucial for the hospital's billing department to prioritize overdue accounts and follow up with patients who may require financial support. Additionally, identifying frequent visitors enables care teams to provide better case management, chronic care planning, or social work services to improve patient outcomes. Overall, this query enhances the hospital's ability to manage resources efficiently while addressing both financial risks and patient care needs.

## C5

5. List Patients name, ID, visit date, and visit ID if they have overlapping two or more visits on the same day.

```
1 •  USE al_Group_47114_G2;
2 •  SELECT
3     p.patientID,
4     p.patientName,
5     v.dateVisited,
6     COUNT(v.visitID) AS visitCount
7   FROM
8     Patient p
9   JOIN
10    Visit v ON p.patientID = v.patientID
11   GROUP BY
12     p.patientID, p.patientName, v.dateVisited
13   HAVING
14     visitCount > 1;
```



The screenshot shows a database query results grid. The top bar includes percentage completion (00%), zoom controls (0.5%, 1:2), and search/filter options. The main area displays a single row of data in a table format:

patientID	patientName	dateVisited	visitCount
187	Basilus Marsham	10/26/2022	2

- This query helps find patients who might need extra attention or care. For example, if someone visits multiple times in a day, it could mean they have a serious health problem that needs quick treatment, like in emergencies. Also, if patients come back for check-ups after a procedure, it might be because of complications or more tests needed. By spotting these patterns, the hospital can improve how it schedules appointments, manages resources, and follows up with patients, leading to better care and outcomes for those with urgent or complex needs.

C6

6. Retrieve the latest prescription for each patient and list the medicine name, dosage amount and frequency of the medicine.

59 • SELECT

```
60   Patient.patientID,  
61   Patient.patientName,  
62   Prescription.prescriptionID,  
63   Medicine.medicineName,  
64   Prescription.dosageAmt,  
65   Prescription.frequency  
66   FROM Patient  
67   JOIN Prescription ON Patient.patientID = Prescription.patientID  
68   JOIN Medicine ON Prescription.medicineID = Medicine.medicineID  
69   WHERE Prescription.prescriptionID =  
70       (SELECT MAX(Prescription.prescriptionID)  
71       FROM Prescription  
72       WHERE Prescription.patientID = Patient.patientID  
73 );
```

0% 1:59

result Grid



Filter Rows:

Search

Export:



patientID	patientName	prescription...	medicineName	dosageAmt	frequency
97	Chelsey Romaines	703	Medicine_101	1000mg	Every 12 hours
97	Chelsey Romaines	703	Atorvastatin	1000mg	Every 12 hours
198	Derrek Waterfall	706	Medicine_132	500mg	Every 12 hours
198	Derrek Waterfall	706	Lisinopril	500mg	Every 12 hours

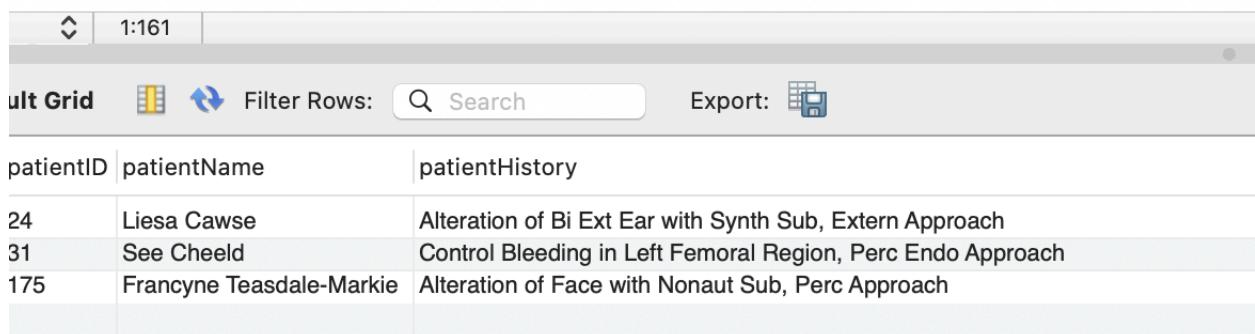
- This query is designed to retrieve the most recent prescription for each patient by utilizing a correlated subquery. The outer query selects patient details along with prescription information, ensuring the data is relevant to the correct individual. The inner subquery plays a key role by identifying the maximum prescriptionID for each patient, which represents the most recent prescription, assuming prescription IDs increase over time. The WHERE condition ensures that only the latest prescription is included in the final result. This approach is especially useful in healthcare systems for tracking current medication records, ensuring medical staff have access to the most up-to-date prescription information, which helps improve patient care and reduces the risk of administering outdated treatments.

## C7

List patients whose medical history includes specific procedures related to "Control Bleeding" or any kind of "Alteration"

- **SELECT**

```
patientID,  
patientName,  
patientHistory  
FROM Patient  
WHERE patientHistory REGEXP '\b(Control Bleeding|Alteration)\b';
```



A screenshot of a database grid interface. The top bar shows the time as 1:161. Below the bar are buttons for 'Multi Grid' (highlighted), 'Filter Rows', 'Search' (with a magnifying glass icon), and 'Export' (with a document icon). The main area is a table with three columns: 'patientID', 'patientName', and 'patientHistory'. The data rows are:

patientID	patientName	patientHistory
24	Liesa Cawse	Alteration of Bi Ext Ear with Synth Sub, Extern Approach
31	See Cheeld	Control Bleeding in Left Femoral Region, Perc Endo Approach
175	Francyne Teasdale-Markie	Alteration of Face with Nonaut Sub, Perc Approach

- The query retrieves patients whose medical history includes procedures related to "Control Bleeding" or "Alteration" by using a regular expression (REGEXP) filter. The result set displays patient IDs, names, and relevant patient history entries, ensuring that only records matching the specified keywords appear. This approach is useful for quickly identifying patients who have undergone specific procedures, helping medical staff analyze trends, provide targeted treatments, and streamline decision-making.

## C8 - Good

### 8. Identify patients who have never filed an insurance claim

```
75 •   SELECT p.patientID, p.patientName  
76     FROM Patient p  
77 WHERE NOT EXISTS (  
78     SELECT *  
79       FROM InsuranceClaim ic  
80      WHERE ic.claimPatient = p.patientID  
81 );  
82  
83  
84
```

The screenshot shows a database query results grid. The top bar includes a progress indicator (0%), a dropdown menu, a timestamp (2:75), and a search bar. Below the header, there are buttons for 'Result Grid' (selected), 'Filter Rows', 'Search', and 'Export'. The data grid has two columns: 'patientID' and 'patientName'. The rows show the following data:

patientID	patientName
2	Amii Brambell
3	Terence Lancaster
14	Maje Walbrun
15	Brendis Tolliday

- The hospital uses this query to identify patients who have never filed an insurance claim, helping staff understand who may be paying out-of-pocket or not utilizing their insurance benefits. This allows the hospital to provide financial guidance, assist with claim filing, and ensure patients are aware of available coverage. By analyzing this data, the hospital can improve billing efficiency, enhance patient support programs, and reduce the risk of missed reimbursements.

## Simple:

S1

1. Find patients with a claim amount greater than the average claim amount of their insurance provider.

```
51 •   SELECT p.patientName, ic.claimAmount
52     FROM Patient p
53     JOIN InsuranceClaim ic ON p.patientID = ic.claimPatient
54     WHERE ic.claimAmount >
55         (SELECT AVG(claimAmount)
56          FROM InsuranceClaim
57         WHERE providerID = ic.providerID);
58
```

100% 1:51

Result Grid Filter Rows: Search Export:

patientName	claimAmount
Fidela Lamblin	4069.00
Aldwin Treslove	4845.00
Efren Gouldsmith	3189.00
Stoddard Tinline	3878.00

- This query helps the hospital flag patients whose claims are significantly higher than the average for their insurance provider. If a patient has a higher-than-average claim, it could mean they have a more complex medical condition requiring additional care or specialized treatment. For example, if a patient consistently has high claims, they might have ongoing health issues that need closer monitoring. It could also help identify potential billing discrepancies, ensuring accurate claims and preventing overcharges. By spotting these patterns, the hospital can improve

financial planning, optimize resource allocation, and enhance patient care for those with complex medical needs.

S2

2. Count the number of insurance claims filed by each patient:

```
45 •   SELECT claimPatient, COUNT(claimID) AS totalClaims  
46     FROM InsuranceClaim  
47     GROUP BY claimPatient  
48     ORDER BY totalClaims DESC;  
49
```

The screenshot shows a database query results grid. At the top, there is a toolbar with a zoom icon (100%), a dropdown menu, the time (1:45), and other standard icons. Below the toolbar is a header row with two columns: 'claimPatient' and 'totalClaims'. The main body of the grid contains four rows of data, each representing a patient and their total number of claims. The data is as follows:

claimPatient	totalClaims
46	4
55	4
59	4
84	4

- This query helps the hospital understand patient interaction with insurance claims. Seeing how many claims each patient files can reveal patterns in hospital visits and treatment needs. For example, patients with frequent claims may rely heavily on medical services, which could indicate a need for better care coordination or alternative treatment plans. It also helps in managing administrative workloads by identifying high-volume claim processing areas. With this information, the hospital can streamline operations, improve patient experience, and ensure smoother insurance processing.

