

Database Systems

Database Document



Alex Kavanagh

Contents

[Database Description 2](#_Toc164625113)

[Enhanced ER Diagram 3](#_Toc164625114)

[Enhanced Logistical Data Model 3](#_Toc164625115)

[Storage Analysis 5](#_Toc164625116)

[Table Design 7](#_Toc164625117)

[Security 10](#_Toc164625118)

[Views 11](#_Toc164625119)

[Queries 12](#_Toc164625120)

## Database Description

Harmony Regional Medical, situated in the Waterford area, is a thriving hospital and recovery centre with dedicated staff in various departments, primarily composed of doctors and nurses. Each staff member is assigned a unique identification number and has essential information stored, including their address, contact number, date of birth, salary, date of hire, and next of kin.

In separate subsets of the staff additional details such as specialty and level are assigned to the doctor, assisting and RND for the Nurse are recorded. The doctors are responsible for diagnosing patients during appointments, while nurses attend to patients in the wards.

Appointments at the hospital are uniquely identified by a code, and pertinent details such as date, time, purpose, and status are stored. While every patient has an appointment, not all patients require admission to a ward.

The hospital comprises of different departments, each with its own set of resources. Departmental data includes the name, department head, and location. These departments cater to various procedures, and resources allocated to them are recorded with information like name, type, quantity, and availability status.

Many patients are housed in various wards at Harmony Regional Medical. In-Patient and Out-Patient are the two subsets of patient information, like staff. Information such as the date of admission, the date of release, and the room number are saved for in-patients, while follow-up data is stored for out-patients. Name, date of birth, phone number, gender, address, and details about their next of kin are all maintained about the patient. Every patient also keeps up-to-date medical records including information about their diagnosis, past medical treatments, test results, and pertinent record dates.

Harmony Regional Medical's database update introduces distinct staff identifiers and subsets for nurses and doctors, streamlining information oversight. Names, heads, and locations are now included in departmental data, and patient appointments are uniquely classified for effective scheduling. Both in-patient and out-patient subsets are covered by the improved patient information management, which records admission and follow-up information. The purpose of these upgrades is to make the information system more efficient and well-organized, which will benefit Harmony Regional Medical's system thus allowing them to provide high-quality healthcare services.

## Enhanced ER Diagram

A diagram of a computer flowchart

Description automatically generated

## Enhanced Logistical Data Model

**All in one table:**

Patient (patientId, name, dateOfBirth, gender, phoneNumber, address, nextOfKin, admissionDate, dischargeDate, roomNumber, followUp)

Primary Key: patientId

**Two Tables:**

In\_PatientPatients (patientId, name, dateOfBirth, gender, phoneNumber, address, nextOfKin, admissionDate, dischargeDate, roomNumber)

Primary Key: patientId

Out\_PatientPatients (patientId, name, dateOfBirth, gender, phoneNumber, address, nextOfKin, followUp)

Primary Key: patientId

**Three Tables:**

Patient (patientId, name, dateOfBirth, gender, phoneNumber, address, nextOfKin)

Primary Key: patientId

In\_PatientPatients (patientId, admissionDate, dischargeDate, roomNumber)

Primary Key: patientId

Foreign Key: patientId references Patient(patientId)

Out\_PatientPatients (patientId, followUp)

Primary Key: patientId

Foreign Key: patientId references Patient(patientId)

Records (medicalRecordId, diagnosis, treatmentHistory, testResults, recordDate, patientId)

Primary Key: patientId, medicalRecordId

Foreign Key: patientId references Patient(patientId)

Department (departmentId, departmentName, departmentHead, location)

Primary Key: departmentId

Resource (resourceId, resourceName, resourceType, resourceQuantity, availabilityStatus, departmentId)

Primary Key: resourceId

Foreign Key: departmentId references Department(departmentId)

Ward (wardId, name, type, capacity, currentOccupancy, departmentId)

Primary Key: wardId

Foreign Key: departmentId references Department(departmentId)

In\_PatientWards (patientId, wardId)

Foreign Key: patientId references Patient(patientId)

Foreign Key: wardId references Ward(wardId)

Appointment (appointmentId, appointmentDate, appointmentTime, purpose, status, patientId, staffId)

Primary Key: appointmentId

Foreign Key: patientId references Patient(patientId)

Foreign Key: staffId references Staff(staffId)

**All in one table:**

Staff (staffId, name, address, phoneNumber, dateOfBirth, salary, dateOfHire, nextOfKin, specialty, level, assisting, RND)

Primary Key: staffId

**Two Tables:**

DoctorStaff (staffId, name, address, phoneNumber, dateOfBirth, salary, dateOfHire, nextOfKin, specialty, level)

Primary Key: staffId

NurseStaff (staffId, name, address, phoneNumber, dateOfBirth, salary, dateOfHire, nextOfKin, assisting, RND)

Primary Key: staffId

**Three Tables:**

Staff (staffId, name, address, phoneNumber, dateOfBirth, salary, dateOfHire, nextOfKin)

Primary Key: staffId

DoctorStaff (staffId, specialty, level)

Primary Key: staffId

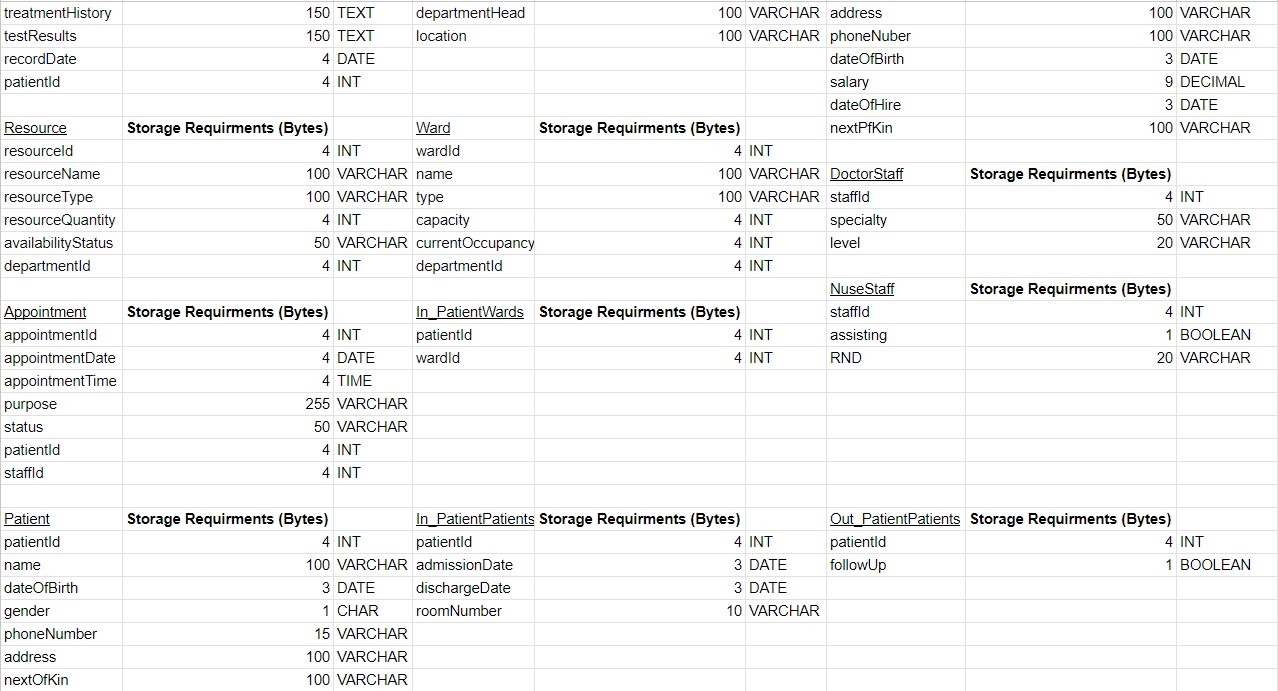
Foreign Key: staffId references Staff(staffId)

NurseStaff (staffId, assisting, RND)

Primary Key: staffId

Foreign Key: staffId references Staff(staffId)

## Storage Analysis



The image above shows the minimum storage requirement for each table’s attributes, with this in mind I will make up records and calculate the total storage requirements.

Records Table: 1000 records

Department Table: 50 records

Staff Table: 200 records

Resource Table: 150 records

Ward Table: 50 records

Appointment Table: 500 records

Patient Table: 300 records

In\_PatientPatients Table: 100 records

Out\_PatientPatients Table: 200 records

DoctorStaff Table: 30 records

NurseStaff Table: 40 records

In\_PatientWards Table: 80 records

Records Table:

Total bytes per record: 4 + 150 + 150 + 150 + 4 + 4 = 462 bytes x 1000 = 462,000 bytes

Department Table:

Total bytes per record: 4 + 100 + 100 + 100 = 304 bytes x 50 = 15,200 bytes

Staff Table:

Total bytes per record: 4 + 100 + 100 + 100 + 3 + 9 + 3 + 100 = 419 bytes x 200 = 83,800 bytes

Resource Table:

Total bytes per record: 4 + 100 + 100 + 4 + 50 + 4 = 262 bytes x 150 = 39,300 bytes

Ward Table:

Total bytes per record: 4 + 100 + 100 + 4 + 4 + 4 =216 bytes x 50 = 10,800 bytes

Appointment Table:

Total bytes per record: 4 + 4 + 4 + 255 + 50 + 4 + 4 = 325 bytes x 500 = 162,500 bytes

Patient Table:

Total bytes per record: 4 + 100 + 3 + 1 + 15 + 100 + 100 = 323 bytes x 300 = 96,900 bytes

In\_PatientPatients Table:

Total bytes per record: 4 + 3 + 3 + 10 = 20 bytes x 100 = 2000 bytes

Out\_PatientPatients Table:

Total bytes per record: 4 + 1 = 5 bytes x 200 = 1000 bytes

DoctorStaff Table:

Total bytes per record: 4 + 50 + 20 = 74 bytes x 30 = 2200 bytes

NurseStaff Table:

Total bytes per record: 4 + 1 + 20 = 25 bytes x 40 = 1000 bytes

In\_PatientWards Table:

Total bytes per record: 4 + 4 = 8 bytes x 80 = 640 bytes

Total Storage Requirements = 462,000 + 15,200 + 83,800 + 39,300 + 10,800 + 162,500 + 96,900 + 2,000

= 874,360 bytes

## Table Design

All in One Table:

Patient

- patientId (Primary Key)

- name

- dateOfBirth

- gender

- phoneNumber

- address

- nextOfKin

- admissionDate

- dischargeDate

- roomNumber

- followUp

Two Tables:

In\_PatientPatients

- patientId (Primary Key)

- name

- dateOfBirth

- gender

- phoneNumber

- address

- nextOfKin

- admissionDate

- dischargeDate

- roomNumber

Out\_PatientPatients

- patientId (Primary Key)

- name

- dateOfBirth

- gender

- phoneNumber

- address

- nextOfKin

- followUp

Three Tables:

Patient

- patientId (Primary Key)

- name

- dateOfBirth

- gender

- phoneNumber

- address

- nextOfKin

In\_PatientPatients

- patientId (Primary Key, Foreign Key referencing Patient(patientId))

- admissionDate

- dischargeDate

- roomNumber

Out\_PatientPatients

- patientId (Primary Key, Foreign Key referencing Patient(patientId))

- followUp

Records

- medicalRecordId

- diagnosis

- treatmentHistory

- testResults

- recordDate

- patientId (Primary Key, Foreign Key referencing Patient(patientId))

Department

- departmentId (Primary Key)

- departmentName

- departmentHead

- location

Resource

- resourceId (Primary Key)

- resourceName

- resourceType

- resourceQuantity

- availabilityStatus

- departmentId (Foreign Key referencing Department(departmentId))

Ward

- wardId (Primary Key)

- name

- type

- capacity

- currentOccupancy

- departmentId (Foreign Key referencing Department(departmentId))

In\_PatientWards

- patientId (Foreign Key referencing Patient(patientId))

- wardId (Foreign Key referencing Ward(wardId))

Appointment

- appointmentId (Primary Key)

- appointmentDate

- appointmentTime

- purpose

- status

- patientId (Foreign Key referencing Patient(patientId))

- staffId (Foreign Key referencing Staff(staffId))

All in One Table:

Staff

- staffId (Primary Key)

- name

- address

- phoneNumber

- dateOfBirth

- salary

- dateOfHire

- nextOfKin

- specialty

- level

- assisting

- RND

Two Tables:

DoctorStaff

- staffId (Primary Key, Foreign Key referencing Staff(staffId))

- specialty

- level

NurseStaff

- staffId (Primary Key, Foreign Key referencing Staff(staffId))

- assisting

- RND

Three Tables:

Staff

- staffId (Primary Key)

- name

- address

- phoneNumber

- dateOfBirth

- salary

- dateOfHire

- nextOfKin

DoctorStaff

- staffId (Primary Key, Foreign Key referencing Staff(staffId))

- specialty

- level

NurseStaff

- staffId (Primary Key, Foreign Key referencing Staff(staffId))

- assisting

- RND

## Security

Department Head:

Read-only: to All

Read/update: to All

Create/read/update: to All

CREATE USER Department Head IDENTIFIED BY department\_head;

GRANT ALL PRIVILEGES ON HospitalManagementSystem TO Department Head;

*SELECT: The user can retrieve data from the table. (read)*

*INSERT: The user can add new rows to the table. (create)*

*UPDATE: The user can modify existing rows in the table. (update)*

The Department Head user account in the Hospital Management System has full access, allowing them to read, update, create, and modify data across all tables. This comprehensive access facilitates effective oversight and management of hospital operations, empowering the Department Head to make informed decisions and maintain efficient processes.

Doctor:

Read-only: to Ward, Resource, staffInfo view, staffQualifications view

Read/update: to Records

Create/read/update: to Patient, Appointment

CREATE USER Doctor IDENTIFIED BY doctor;

GRANT SELECT ON ward TO Doctor;

GRANT SELECT ON resource TO Doctor;

GRANT SELECT ON staffInfo TO Doctor;

GRANT SELECT ON staffQualifications TO Doctor;

GRANT SELECT, UPDATE ON records TO Doctor;

GRANT CREATE, SELECT, UPDATE ON patient TO Doctor;

GRANT CREATE, SELECT, UPDATE ON appointment TO Doctor;

*SELECT: The user can retrieve data from the table. (read)*

*INSERT: The user can add new rows to the table. (create)*

*UPDATE: The user can modify existing rows in the table. (update)*

The Doctor user profile grants read-only access to Ward and Resource data, allowing them to view relevant information. They can read and update medical Records, while also creating, reading, and updating Patient and Appointment records. These permissions enable efficient patient care management within the healthcare system.

Nurse:

Read-only: to Patient, Resource, staffInfo view, staffQualifications view

Read/update: to Ward

Create/read/update: to Records

CREATE USER Nurse IDENTIFIED BY nurse;

GRANT SELECT ON patient TO Nurse;

GRANT SELECT ON resource TO Nurse;

GRANT SELECT ON staffInfo TO Nurse;

GRANT SELECT ON staffQualifications TO Nurse;

GRANT SELECT, UPDATE ON ward TO Nurse;

GRANT CREATE, SELECT, UPDATE ON records TO Nurse;

*SELECT: The user can retrieve data from the table. (read)*

*INSERT: The user can add new rows to the table. (create)*

*UPDATE: The user can modify existing rows in the table. (update)*

The Nurse user profile provides specific access privileges tailored to their role. Nurses have read-only access to Patient and Resource data, allowing them to view relevant information. They can read and update Ward details, while also creating, reading, and updating medical Records. These permissions facilitate efficient patient care management within the healthcare system.

## Views

All staff members are given the ability to view restricted data on the Staff table, including staffId, name, and phone number. The entire contents of the Staff table, including certain personal information like date of birth, salary, address, employment date, and next of kin, will only be fully accessible to the Department Head. The basic staff information listed below will be accessible to all staff members, allowing them to keep track of tasks and get in touch with coworkers who aren't in the office.

CREATE VIEW staffInfo AS

SELECT staffId, name, phoneNumber

FROM staff;

This view would provide information about staff members and their qualifications, showing whether they have a specialty as a doctor or if they assist as a nurse, allowing all staff members to understand the roles and qualifications of their coworkers.

CREATE VIEW staffQualifications AS

SELECT s.staffId, s.name, ds.specialty AS doctorSpecialty, ns.assisting AS nurseAssisting

FROM Staff AS s

LEFT JOIN DoctorStaff AS ds ON s.staffId = ds.staffId

LEFT JOIN NurseStaff AS ns ON s.staffId = ns.staffId;

## Queries

1. Show the names, admission dates, discharge dates, and room numbers of patients who were admitted to the hospital.
2. Display the names and next of kin information for all patients who visited the hospital as out-patients.
3. List the appointment dates, times, and purposes for all appointments scheduled for a specific patient.
4. Show the medical records, diagnoses, and treatment histories for a particular patient.
5. List the names, types, and quantities of all available resources in a specific department.
6. Display the names, current occupancies, and capacities of all hospital wards.
7. List the names, specialties, and levels of all doctors employed at the hospital.
8. Show the names and phone numbers of all nurses working at the hospital.