1. a) Database is a collection of related data. Data refers to main facts that can be recorded and have implicit meaning.

Database management system is a software system to facilitate the creation and maintenance of a computerized database.

A DBMS can improve the operations of a real estate business by providing an efficient way to manage data through the following improvements.

It provides a backup and recovery services.

It provides multiple interfaces to different users.

It presents complex relationships among data e.g properties and clients.

b) Through authentication where only users can access the database.

By using credentials such as usernames, passwords and other advanced methods like biometric authentication.

Through using encryption which protects stored data, and ensures that unauthorized users do not understand it without description keys.

By using security updates.

Through audit trials by recording who accessed which data in order to detect unauthorized access.

c) **Types of database users and their roles**.

Database administrators, they have full access to all the system features and are responsible for coordinating and monitoring database use.

Managers, these are Real Estate managers. They can insert, update data related to properties, agents and clients.

Clients: these users may view property listings and submit preferences but have no access to the underlying data in the system.

Agents, these can view properties and transactions related to their performance.

d) DDL (Data Definition Languages)

It is used by database designers to specify the conceptual schema of a database. It is also used to define internal and external schemas.

DML (Data Manipulation Languages)

This is used to specify database retrievals and updates. It’s used in inserting updating and retrieving data.

DCL (Data Control Language)

This manages user permissions and access control.

f**) normalize the structure to 3NF.**

A table is in 1NF if:

All columns contain atomic values.

Each column has a unique name.

Each row is unique and there are no duplicate rows.

In 2NF:

It is in 1NF.

All non-key attributes are fully functionally dependent on primary key.

In 3NF:

It is in 2NF.

It has no non- key attributes depending on the primary key.

g)

1. CREATE TABLE Properties (Property ID INT PRIMARY KEY, Address VARCHAR (255) NOT NULL, City VARCHAR (100) NOT NULL, Type VARCHAR (50) CHECK (Type IN ('Residential', 'Commercial', 'Industrial')), Size INT NOT NULL, Price DECIMAL (10, 2) NOT NULL, Status VARCHAR (20) CHECK (Status IN ('Available', 'Sold', 'Rented'))
2. INSERT INTO Properties (Property ID, Address, City, Type, Size, Price, Status) VALUES (1, '123 Main St', 'Kampala', 'Residential', 2000, 250000.00, 'Available'), (2, '456 Elm St', 'Kampala', 'Commercial', 5000, 500000.00, 'Available'), ... (8 more records) ... ;
3. SELECT \* FROM Properties WHERE Status = 'Available' AND City = 'Kampala';
4. UPDATE Properties SET Status = 'Sold' WHERE Property ID = 1;

h)

If a transaction is recorded with a Property ID that does not exist in the Properties table, it violates referential integrity. Constraints like foreign keys prevent such issues by either disallowing the transaction or cascading actions (like deleting related transactions) based on the DBMS configuration.

1. Property status constraint.

CHECK (Status IN ('Available', 'Sold', 'Rented'))

Agent commission rate constraint.

CHECK (Commission Rate BETWEEN 1.0 AND 15.0)

Transaction amount constraint.

1. 1) Primary entities and their relationship to one another.

Patient data, visit record, laboratory test, medical supplies, resources, user role, supply chain.

Patient data **to** visit record (one patient can have many visit records)

Visit record **to** laboratory test (each visit record can have multiple laboratory test)

Visit records **to** medical supplies

Medical supplies **to** supply chain (a medical supply item can be linked to multiple supply chain entities)

User role **to** patient data.

2) Missing relationship between patient data and laboratory test. There is no direct relation yet it should be there to show which tests were ordered for which patient.

User access control. The user might need further expansion with specific permission for different roles such as viewing or editing patient data.

Supply chain and resource. The relationship between the two might need clarification. Are resources like beds or staff being tracked alongside medical supplies?

3) The relationship between patient data and visit-records is that a patient can have many visit records but each visit records can only correspond to a single patient.

This is important because it helps in tracking patient history and ensuring proper follow-up since the visit-records includes each patient encounter with the system.

4) The user-role table defines different access levels for various users in the system, such as:

Admin: full system access.

Doctor: access to patients’ data and visit records.

Nurse: limited access to patients ‘records but may have access to treatment records.

Pharmacist: access to resource and medical supply data.

5) The supply chain and resource tables can be used together to track the status of medical supplies.

**Supply chain table:** this contains information on the inventory, order status and supplies.it can be used to identify if orders for medical supplies have been delayed or if quantities are running low.

**Resource table:** this tracks the available resources in the hospital like staff, beds. By integrating this with the supply chain, the hospital can monitor and optimize resource utilization ensuring that the hospital does not run out of critical supplies.

7) If the hospital decides to expand the system for non- malaria patients, the following parts would need extension.

Visit record. The treatment and diagnosis information in the visit record may need to expand to include general treatments beyond malaria.

Treatment record. There should be treatment table that logs. all treatments given for diseases.

A new entity like disease may need to be added with attributes like disease ID, disease name.

8) [CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) `patient data`.`patient data` (`patient ID` INT [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , `Name` VARCHAR(100) NULL , `Age` INT NULL , `Gender` VARCHAR(10) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , `Address` VARCHAR(225) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , PRIMARY KEY (`patient ID`)) ENGINE = InnoDB

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) `patient data`.`visit record` (`visit ID` INT [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , `Patient ID` INT [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , `visit date` DATE NULL , `diagnosis` VARCHAR(225) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , `Treatment` VARCHAR(225) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , PRIMARY KEY (`visit ID`)) ENGINE = InnoDB;

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) `hospital data`.`labaratory test` (`Test ID` INT [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , `VISIT ID` INT [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , `test type` VARCHAR(100) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , `test result` VARCHAR(225) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , PRIMARY KEY (`Test ID`)) ENGINE = InnoDB;

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) `medical supplies`.`medical supplies` (`supply ID` INT [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , `NAME` VARCHAR(100) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , `quantity` INT [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , `expiry date` DATE [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , PRIMARY KEY (`supply ID`)) ENGINE = InnoDB;