**Hospital Management Database Schema Documentation**

**Overview:** designed to efficiently manage and organize data related to the hospital operations.

This relational database schema ensures data integrity, supports efficient querying, and is adaptable to future needs. Below is an overview of the key components and their roles within the system:

#### **1. Hospitals Table**

**Table Name:** hospitals

**Purpose:**  
The hospitals table serves as the central directory for all hospital entities within the system. It stores essential details about each hospital, providing foundational data required for linking with other key tables, including patients, employees, rooms, and more. This structure ensures that all hospital-related operations, from patient admissions to employee management, are centralized and easily accessible.

**Role:**  
As the core table in the database, hospitals functions as a primary reference point for multiple relationships within the database. It connects directly to other critical tables, ensuring that any patient, employee, or facility data can be traced back to the corresponding hospital. This centralization is crucial for maintaining data integrity across the system, ensuring that all operations are correctly attributed to the right hospital entity.

**Columns:**

* **hospital\_id (BIGINT):** Unique identifier for each hospital. Acts as the primary key, ensuring that each hospital entry is distinct.
* **name (VARCHAR(255)):** The official name of the hospital. This column ensures that each hospital's identity is recorded clearly.
* **location (VARCHAR(255)):** The geographical location or address of the hospital. This is critical for operational purposes, such as patient and supply logistics.

**Operations:**

* **Insert:** New hospitals can be added to the database using the INSERT statement. This operation is crucial when expanding the healthcare network or establishing new facilities.
* **Update:** Hospital information, such as changes in name or location, can be modified using the UPDATE statement. This ensures that the hospital data remains current.
* **Delete:** Hospitals can be removed using the DELETE statement. However, this operation should be performed with caution due to the potential cascading effects on related data (e.g., patient records). Ensure proper data migration or archiving before deletion.

**Note:** Regular audits and validations should be conducted on this table to maintain the accuracy of the data.

#### **2. Patients Table**

**Table Name:** patients

**Purpose:**  
The patients table is designed to store comprehensive information about all patients who receive care at the hospitals. This table forms the basis for patient management, enabling the healthcare system to track individual patient details, including personal information, contact details, and the hospital they are associated with.

**Role:**  
This table plays a critical role in linking patient records to their respective hospital and other relevant data like medical history, prescriptions, and admissions. By associating patients with their respective hospital through the hospital\_id, this table ensures that all patient-related activities are accurately documented and attributed to the correct hospital entity.

**Columns:**

* **patient\_id (BIGINT):** Serves as the primary key, uniquely identifying each patient.
* **first\_name (VARCHAR(255)):** Records the patient's first name, ensuring clarity in identity.
* **last\_name (VARCHAR(255)):** Captures the patient's last name for accurate identification.
* **gender (CHAR(1)):** Indicates the patient’s gender, essential for medical documentation.
* **birth\_date (DATE):** Stores the patient's date of birth, crucial for age-related medical decisions.
* **address (VARCHAR(255)):** Contains the residential address of the patient.
* **place\_of\_birth (VARCHAR(255)):** Notes the patient's birthplace, which may be relevant for certain medical conditions.
* **phone\_number (VARCHAR(255)):** Records the contact number, used for communication.
* **email (VARCHAR(255)):** Stores the patient’s email address for electronic communication.
* **hospital\_id (BIGINT):** A foreign key that links the patient to their hospital, ensuring all patient data is tied back to the correct hospital entity.

**Operations:**

* **Insert:** New patients are added using the INSERT statement when they first register at a hospital.
* **Update:** Patient information can be updated using the UPDATE statement, for instance, if a patient changes their address or contact details.
* **Delete:** Typically, patient records are not deleted due to medical record-keeping requirements. However, if necessary, a DELETE operation can be performed, ideally after archiving or transferring the data to a secure storage system to maintain a record of care provided.

**Note:** For compliance with data privacy laws, patient data must be handled with the utmost care, ensuring that only authorized personnel can access or modify records.

### ****3. Employees Table****

**Table Name:** employees

**Purpose:**  
The employees table records the details of all hospital staff, including doctors, nurses, administrative personnel, and other support staff. This table is essential for managing employee records, from their roles and departments to their contact information and employment history.

**Role:**  
This table connects the staff to their respective hospital and department, ensuring that all employee-related data is centralized and accessible for management, payroll, and operational purposes. It is integral to the administration of hospital operations, linking staff with their roles and responsibilities within the hospital.

**Columns:**

* **employee\_id (BIGINT):** Unique identifier for each employee. Serves as the primary key.
* **first\_name (VARCHAR(255)):** Records the employee's first name.
* **last\_name (VARCHAR(255)):** Captures the employee's last name.
* **gender (CHAR(1)):** Indicates the employee's gender.
* **role (VARCHAR(255)):** Describes the employee's role within the hospital (e.g., doctor, nurse, administrator).
* **department (VARCHAR(255)):** Indicates the department where the employee works.
* **specialisation (VARCHAR(255)):** Notes any specializations, particularly relevant for medical professionals.
* **phone\_number (VARCHAR(255)):** Contact number for the employee.
* **start\_date (DATE):** The date the employee started working at the hospital.
* **end\_date (DATE):** The date the employee left the hospital, if applicable.
* **salary\_usd (FLOAT(53)):** Records the employee's salary.
* **hospital\_id (BIGINT):** Foreign key linking the employee to their hospital.

**Operations:**

* **Insert:** Add new employees when they join the hospital using the INSERT statement.
* **Update:** Update employee records, such as changes in department, role, or salary, using the UPDATE statement.
* **Delete:** Employee records might be deleted if the data retention policy allows, typically after a set period post-employment. Deletions should be handled carefully, ensuring compliance with labor laws and record-keeping standards.

**Note:** Employee data should be maintained securely, with access restricted to authorized personnel only. Regular updates and audits are essential to keep records accurate and up to date.

### ****4. Rooms Table****

**Table Name:** rooms

**Purpose:**  
The rooms table catalogues the different types of rooms available in each hospital, including details such as room type, cost, and capacity. This table is essential for hospital management, especially in allocating rooms to patients and managing hospital resources.

**Role:**  
This table plays a vital role in the logistical management of hospital facilities. It helps in tracking room availability, planning patient admissions, and ensuring that the hospital operates efficiently by optimizing room utilization.

**Columns:**

* **room\_id (BIGINT):** Unique identifier for each room. Part of the composite primary key.
* **hospital\_id (BIGINT):** Links the room to a specific hospital. Part of the composite primary key.
* **room\_type (VARCHAR(255)):** Describes the type of room (e.g., ICU, general ward, private).
* **room\_cost (FLOAT(53)):** Indicates the cost associated with the room.
* **room\_capacity (INTEGER):** Specifies the number of patients the room can accommodate.

**Operations:**

* **Insert:** New rooms are added to the table using the INSERT statement when a hospital expands or modifies its facilities.
* **Update:** Room details, such as changes in cost or capacity, can be updated using the UPDATE statement.
* **Delete:** Rooms might be deleted from the table if they are decommissioned or repurposed, following a thorough review to ensure no active patient records are associated with the room.

**Note:** The room data should be kept accurate to facilitate efficient hospital operations, with changes carefully managed to prevent discrepancies in patient allocation.

### ****5. Prescription Table****

**Table Name:** prescription

**Purpose:**  
The prescription table records details of medical prescriptions issued to patients, including information about the prescribing doctor, the prescribed medicine, dosage, and quantity.

**Role:**  
This table is central to managing patient treatment plans, linking each prescription to the patient and the prescribing doctor. It ensures that all prescribed medications are tracked, aiding in patient care and supporting pharmacy operations.

**Columns:**

* **prescription\_id (BIGINT):** Unique identifier for each prescription. Primary key.
* **patient\_id (BIGINT):** Foreign key linking the prescription to the patient.
* **doctor\_id (BIGINT):** Foreign key linking the prescription to the doctor who prescribed the medication.
* **dose\_mg (VARCHAR(255)):** Specifies the dosage of the medication.
* **medicine (VARCHAR(255)):** Name of the prescribed medicine.
* **quantity (INTEGER):** Quantity of the medicine prescribed.

**Operations:**

* **Insert:** Prescriptions are added to the table using the INSERT statement when a doctor prescribes medication to a patient.
* **Update:** Typically, prescriptions are not updated once issued. If an update is necessary (e.g., change in dosage), it should be done carefully, ensuring proper documentation.
* **Delete:** Deletion of prescriptions is generally not recommended due to legal and medical record-keeping requirements. If necessary, it should be done following strict protocols, with records archived appropriately.

**Note:** Proper handling and documentation of prescription data are essential for legal compliance and accurate medical records. Always adhere to relevant regulations and protocols when managing prescription data.

### ****6. Insurance Table****

**Table Name:** insurance

**Purpose:**  
The insurance table captures information about patients' insurance policies, including policy numbers, providers, and coverage rates. This table is crucial for managing insurance claims and ensuring that patients' treatments are covered as per their insurance agreements.

**Role:**  
The insurance table connects patients with their insurance providers, facilitating the processing of insurance claims and managing coverage details. It ensures that billing and financial operations are streamlined and that insurance-related data is accurately recorded.

**Columns:**

* **patient\_id (BIGINT):** Foreign key linking the insurance record to the patient.
* **policy\_number (VARCHAR(255)):** Unique insurance policy number. Acts as a unique constraint.
* **insurance\_provider (VARCHAR(255)):** Name of the insurance provider.
* **coverage\_rate (INTEGER):** Percentage of coverage provided by the insurance.

**Operations:**

* **Insert:** Add new insurance records when patients obtain insurance policies using the INSERT statement.
* **Update:** Update insurance details such as coverage rate or insurance provider, if applicable, using the UPDATE statement.
* **Delete:** Deleting insurance records should be done cautiously. Ensure that the patient’s record and related insurance information are handled according to data retention policies.

**Note:** The insurance data should be managed with high security to protect sensitive financial and personal information.

### ****7. Lab Table****

**Table Name:** lab

**Purpose:**  
The lab table stores information about lab tests performed for patients, including test types, results, and dates. It is essential for tracking lab work and results, which are integral to patient diagnosis and treatment.

**Role:**  
This table links lab tests to patients and their respective hospitals, facilitating the tracking and management of diagnostic tests. It supports medical staff in reviewing test results and making informed decisions about patient care.

**Columns:**

* **lab\_id (BIGINT):** Unique identifier for each lab test. Primary key.
* **hospital\_id (BIGINT):** Foreign key linking the lab test to the hospital where the test was conducted.
* **patient\_id (BIGINT):** Foreign key linking the lab test to the patient.
* **lab\_doctor\_id (BIGINT):** Foreign key linking the lab test to the doctor who ordered or reviewed the test.
* **test\_type (VARCHAR(255)):** Type of lab test conducted.
* **test\_date (DATE):** Date when the test was performed.
* **test\_result (VARCHAR(255)):** Result of the lab test.
* **result\_date (DATE):** Date when the test result was reported.

**Operations:**

* **Insert:** Add new lab test records using the INSERT statement.
* **Update:** Update lab test details if necessary (e.g., correcting results or test types), using the UPDATE statement.
* **Delete:** Deleting lab test records is generally not recommended due to their importance in patient medical history. If deletion is necessary, it should be done with proper archival.

**Note:** Lab data should be securely stored and handled to maintain patient confidentiality and comply with regulatory requirements.

### ****8. Bill Table****

**Table Name:** bill

**Purpose:**  
The bill table maintains detailed records of all charges associated with a patient’s hospital stay, including costs for lab tests, room charges, consultations, medications, and other expenses. This table is crucial for accurate billing and financial management.

**Role:**  
It plays a central role in hospital financial operations, linking billing records to patients, hospitals, and doctors. This ensures that all charges are tracked and processed correctly for billing purposes.

**Columns:**

* **bill\_id (BIGINT):** Unique identifier for each bill. Primary key.
* **patient\_id (BIGINT):** Foreign key linking the bill to the patient.
* **hospital\_id (BIGINT):** Foreign key linking the bill to the hospital where the services were provided.
* **doctor\_id (BIGINT):** Foreign key linking the bill to the doctor who provided the services.
* **lab\_cost (FLOAT(53)):** Cost associated with lab tests.
* **room (FLOAT(53)):** Cost associated with the hospital room.
* **consult (FLOAT(53)):** Cost of consultations with doctors.
* **med\_cost (FLOAT(53)):** Cost of medications.
* **other (FLOAT(53)):** Any additional costs not covered by the above categories.
* **bill\_date (DATE):** Date when the bill was issued.

**Operations:**

* **Insert:** Add new billing records using the INSERT statement when services are provided.
* **Update:** Update billing details if errors are found or adjustments are required, using the UPDATE statement.
* **Delete:** Typically, bills are not deleted to maintain financial records. However, adjustments or corrections should be handled according to financial protocols.

**Note:** Billing records should be carefully managed to ensure accuracy in financial reporting and patient charges.

### ****9. Pharmacy Table****

**Table Name:** pharmacy

**Purpose:**  
The pharmacy table tracks the dispensing of medications to patients, including details about the medicine, its price, and the associated prescription. It is essential for managing pharmacy operations and ensuring accurate medication records.

**Role:**  
This table connects pharmacy operations with prescriptions and patient records, ensuring that medication dispensed is tracked and matched to the corresponding insurance policies. It supports inventory management and patient billing.

**Columns:**

* **pharmacy\_id (BIGINT):** Unique identifier for each pharmacy record. Primary key.
* **patient\_id (BIGINT):** Foreign key linking the pharmacy record to the patient.
* **hospital\_id (BIGINT):** Foreign key linking the pharmacy record to the hospital where the medication was dispensed.
* **prescription\_id (BIGINT):** Foreign key linking the pharmacy record to the prescription.
* **medicine (VARCHAR(255)):** Name of the medicine dispensed.
* **medicine\_price (FLOAT(53)):** Price of the medicine.
* **quantity (INTEGER):** Quantity of the medicine dispensed.
* **policy\_number (VARCHAR(255)):** Foreign key linking the pharmacy record to the insurance policy.
* **p\_date (DATE):** Date when the medicine was dispensed.

**Operations:**

* **Insert:** Add new records of medications dispensed using the INSERT statement.
* **Update:** Update pharmacy records if necessary (e.g., price changes), using the UPDATE statement.
* **Delete:** Deletion of pharmacy records is typically avoided to maintain accurate dispensing records. Any necessary changes should be managed with appropriate documentation.

**Note:** Ensure that pharmacy records are maintained accurately to support both billing and inventory management.

### ****10. Admissions Table****

**Table Name:** admissions

**Purpose:**  
The admissions table records details about patient admissions to hospitals, including admission and discharge dates, diagnosis, and the assigned room. This table is essential for tracking patient stays and hospital bed utilization.

**Role:**  
This table connects patient admissions with hospital rooms and staff, supporting operational management and patient care tracking. It helps in monitoring room usage and ensuring that patient records are accurately maintained.

**Columns:**

* **admission\_id (BIGINT):** Unique identifier for each admission. Primary key.
* **patient\_id (BIGINT):** Foreign key linking the admission to the patient.
* **diagnosis (VARCHAR(255)):** Diagnosis provided upon admission.
* **admission\_date (DATE):** Date of patient admission.
* **discharge\_date (DATE):** Date of patient discharge.
* **doctor\_id (BIGINT):** Foreign key linking the admission to the doctor responsible.
* **room\_id (BIGINT):** Foreign key linking the admission to the room where the patient stayed.
* **hospital\_id (BIGINT):** Foreign key linking the admission to the hospital.

**Operations:**

* **Insert:** Record new admissions using the INSERT statement.
* **Update:** Update admission details as needed, such as discharge dates or diagnoses, using the UPDATE statement.
* **Delete:** Deleting admissions records is generally not recommended due to their importance in medical history. Any necessary deletions should be handled with proper archiving.

**Note:** Maintain accurate admission records for effective hospital management and patient care.

### ****11. Suppliers Table****

**Table Name:** suppliers

**Purpose:**  
The suppliers table maintains information about suppliers providing goods and services to hospitals, including their contact details and countries of operation. This table is vital for managing the supply chain and ensuring that hospitals have access to necessary resources.

**Role:**  
This table links suppliers to inventory and procurement processes, facilitating the acquisition of medical supplies and equipment. It supports operational efficiency and ensures that hospitals can maintain adequate stock levels.

**Columns:**

* **supplier\_id (BIGINT):** Unique identifier for each supplier. Primary key.
* **name (VARCHAR(255)):** Name of the supplier.
* **contact (VARCHAR(255)):** Contact details for the supplier.
* **country (VARCHAR(255)):** Country where the supplier is based.

**Operations:**

* **Insert:** Add new supplier records using the INSERT statement.
* **Update:** Update supplier details if necessary (e.g., contact information), using the UPDATE statement.
* **Delete:** Suppliers can be removed using the DELETE statement if they are no longer active or relevant, ensuring that associated inventory records are appropriately managed.

**Note:** Supplier data should be updated regularly to maintain accurate contact and operational details.

### ****12. Inventory Table****

**Table Name:** inventory

**Purpose:**  
The inventory table tracks medical and other supplies available in hospitals, including details about the item, its price, quantity, and supplier. This table is crucial for inventory management and procurement processes.

**Role:**  
This table connects hospital inventory with suppliers, enabling effective tracking of stock levels and cost management. It supports hospital operations by ensuring that necessary supplies are always available.

**Columns:**

* **inventory\_id (BIGINT):** Unique identifier for each inventory item. Primary key.
* **item\_name (VARCHAR(255)):** Name of the inventory item.
* **price (FLOAT(53)):** Price of the item.
* **quantity (INTEGER):** Quantity of the item in stock.
* **supplier\_id (BIGINT):** Foreign key linking the inventory item to its supplier.
* **hospital\_id (BIGINT):** Foreign key linking the inventory item to the hospital.

**Operations:**

* **Insert:** Add new inventory items using the INSERT statement.
* **Update:** Update inventory details (e.g., price or quantity), using the UPDATE statement.
* **Delete:** Deleting inventory records should be done carefully to avoid discrepancies in stock levels. Ensure proper handling of associated data.

**Note:** Regular inventory audits are essential to ensure accurate stock levels and pricing.

### ****13. Medical History Table****

**Table Name:** medical\_history

**Purpose:**  
The medical\_history table records comprehensive details about patients' past medical conditions, treatments, and allergies. This table is vital for maintaining a complete medical history for each patient, which aids in ongoing and future medical care.

**Role:**  
It provides a historical context for patient care, supporting doctors in making informed decisions based on past medical data. This table helps in understanding patient history and improving treatment planning.

**Columns:**

* **hospital\_id (BIGINT):** Foreign key linking to the hospital.
* **patient\_id (BIGINT):** Foreign key linking to the patient.
* **blood\_type (VARCHAR(255)):** Patient’s blood type.
* **allergy (VARCHAR(255)):** Recorded allergies.
* **diagnosis (VARCHAR(255)):** Past diagnoses.
* **condition\_name (VARCHAR(255)):** Names of past medical conditions.

**Operations:**

* **Insert:** Add new medical history records using the INSERT statement.
* **Update:** Update medical history details as needed (e.g., new allergies or conditions), using the UPDATE statement.
* **Delete:** Typically, medical history records are not deleted to maintain complete patient records. If deletion is necessary, ensure proper archival.

**Note:** Medical history records should be meticulously maintained to support accurate patient care and historical reference.

### ****14. Appointments Table****

**Table Name:** appointments

**Purpose:**  
The appointments table records details about patient appointments with doctors. It includes the appointment date, status, and purpose. This table helps manage and organize patient visits, ensuring effective scheduling and tracking of appointments.

**Role:**  
This table is central to scheduling and managing patient appointments. It links patients, doctors, and hospitals, providing a complete record of appointments to support efficient healthcare delivery and administrative coordination.

**Columns:**

* **appointment\_id (BIGINT):** Unique identifier for each appointment. Primary key.
* **patient\_id (BIGINT):** Foreign key linking the appointment to the patient.
* **doctor\_id (BIGINT):** Foreign key linking the appointment to the doctor.
* **hospital\_id (BIGINT):** Foreign key linking the appointment to the hospital.
* **app\_date (DATE):** Date of the appointment.
* **app\_status (VARCHAR):** Status of the appointment (e.g., scheduled, completed, cancelled).
* **purpose (VARCHAR):** Purpose of the appointment (e.g., consultation, follow-up).
* **contact (VARCHAR):** Contact information related to the appointment.

**Operations:**

* **Insert:** Add new appointments when scheduling a visit.
* **Update:** Modify appointment details such as date or status if changes are needed.
* **Delete:** Remove appointments if cancelled or no longer required, adhering to record-keeping protocols.

### ****15. Manipulation Table****

**Table Name:** manipulation

**Purpose:**  
The manipulation table tracks medical manipulations or procedures performed on patients, including the name, category, cost, and date of the procedure. This table supports the management of medical interventions and financial tracking related to procedures.

**Role:**  
This table is key to documenting medical procedures performed at hospitals. It links patients and hospitals, providing a detailed record of manipulations essential for billing, insurance processing, and maintaining patient records.

**Columns:**

* **manipulation\_id (BIGINT):** Unique identifier for each manipulation. Primary key.
* **patient\_id (BIGINT):** Foreign key linking the manipulation to the patient.
* **hospital\_id (BIGINT):** Foreign key linking the manipulation to the hospital.
* **man\_name (VARCHAR):** Name of the medical manipulation or procedure.
* **category (VARCHAR):** Category of the manipulation (e.g., surgical, diagnostic).
* **cost (FLOAT):** Cost associated with the manipulation.
* **man\_date (DATE):** Date when the manipulation was performed.
* **isinsurance (BOOLEAN):** Indicates if the manipulation cost is covered by insurance.

**Operations:**

* **Insert:** Add new manipulation records when a procedure is performed.
* **Update:** Modify manipulation details such as cost or category as needed.
* **Delete:** Remove records if necessary, following proper archival and documentation practices.